

LaTeX

[Wikibooks.org](https://en.wikibooks.org/wiki/LaTeX)

June 29, 2012

On the 28th of April 2012 the contents of the English as well as German Wikibooks and Wikipedia projects were licensed under Creative Commons Attribution-ShareAlike 3.0 Unported license. An URI to this license is given in the list of figures on page 1271. If this document is a derived work from the contents of one of these projects and the content was still licensed by the project under this license at the time of derivation this document has to be licensed under the same, a similar or a compatible license, as stated in section 4b of the license. The list of contributors is included in chapter Contributors on page 1211. The licenses GPL, LGPL and GFDL are included in chapter Licenses on page 1295, since this book and/or parts of it may or may not be licensed under one or more of these licenses, and thus require inclusion of these licenses. The licenses of the figures are given in the list of figures on page 1271.

Contents

I.	Getting Started	3
1.	Introduction	5
1.1.	What is TeX?	5
1.2.	What is LaTeX?	8
1.3.	Other terms regarding TeX	14
1.4.	Prerequisites	15
1.5.	What next?	32
2.	Basics	35
2.1.	The LaTeX source	36
2.2.	Input File Structure	44
2.3.	Our first document	56
2.4.	And what now?	71

3.	Document Structure	77
3.1.	The document environment	78
4.	Errors and Warnings	101
4.1.	Error messages	102
4.2.	Warnings	104
4.3.	Examples	105
5.	Bibliography Management	117
5.1.	Embedded system	118
5.2.	Citations	121
5.3.	BibTeX	133
5.4.	Bibliography in the table of contents	180
5.5.	Multiple Bibliographies	187
5.6.	References	189
II.	Formatting	191
6.	List Structures	193
6.1.	List Structures	194
6.2.	Easylist package	216
7.	Tables	219
7.1.	Floating with table	221
7.2.	The tabular environment	224

7.3.	Spanning	248
7.4.	Resize tables	255
7.5.	Sideways tables	258
7.6.	Alternate Row Colors in Tables	258
7.7.	Colors of individual Cells	258
7.8.	Partial Vertical Lines	258
7.9.	Space between rows	258
7.10.	The <code>table</code> environment - captioning etc	258
7.11.	The <code>tabular*</code> environment - controlling table width	258
7.12.	The <code>tabularx</code> package - simple column stretching	258
7.13.	Vertically centered images	258
7.14.	Professional tables	258
7.15.	Adding rule spacing above or below <code>\hline</code> and <code>\cline</code> commands	258
7.16.	Tables with different font size	258
7.17.	Table with legend	258
7.18.	Need more complicated features?	258
7.19.	Summary	258
7.20.	References	258

8.	Title Creation	258
8.1.	Standard Title Pages	259
8.2.	Custom Title Pages	262
9.	Page Layout	273
9.1.	Page dimensions	274
9.2.	Page orientation	284
9.3.	Page styles	288
9.4.	Multi-column pages	307
9.5.	Manual page formatting	310
9.6.	Widows and orphans	313
9.7.	Summary	315
10.	Paragraph Formatting	317
10.1.	Paragraph Alignment	317
10.2.	Paragraph Indents	320
10.3.	Line Spacing	324
10.4.	Special Paragraphs	327
10.5.	Notes and References	341
11.	Importing Graphics	343
11.1.	The <code>graphicx</code> package	344
11.2.	Creating Vector Graphics	378

11.3. References	385
12. Creating Graphics	387
12.1. Overview	389
12.2. Alternatives	392
13. Floats, Figures and Captions	399
13.1. Floats	400
13.2. Captions	407
13.3. Labels in the figures	434
13.4. References	436
14. Hyperlinks	437
14.1. Hyperref	437
14.2. Usage	438
14.3. Customization	445
14.4. Problems with Links and Equations	457
14.5. Problems with Links and Pages	458
14.6. Problems with Bookmarks	462
14.7. Problems with tables and figures	464
14.8. Problems with long caption and \listoffigures or long title	466
14.9. Problems with already existing .toc, .lof and similar files	467
14.10. Problems with footnotes and special characters	468

14.11.	Problems with Beamer	468
14.12.	References	469
15.	Colors	471
15.1.	Adding the color package	472
15.2.	Entering colored text	473
15.3.	Entering colored background for the text	475
15.4.	Predefined colors	476
15.5.	The 68 standard colors known to dvips	477
15.6.	Defining new colors	483
15.7.	Advanced color settings	489
15.8.	Sources	490
16.	Fonts	491
16.1.	XeTeX	494
16.2.	Useful websites	496
III.	Technical Texts	499
17.	Mathematics	501
17.1.	Mathematics environments	503
17.2.	Symbols	506
17.3.	Greek letters	508
17.4.	Operators	510

17.5.	Powers and indices	512
17.6.	Fractions and Binomials	514
17.7.	Roots	522
17.8.	Sums and integrals	527
17.9.	Brackets, braces and delimiters	533
17.10.	Matrices and arrays	538
17.11.	Adding text to equations	549
17.12.	Formatting mathematics symbols	554
17.13.	Plus and minus signs	561
17.14.	Controlling horizontal spacing	562
17.15.	Advanced Mathematics: AMS Math package	572
17.16.	List of Mathematical Symbols	579
17.17.	Summary	583
17.18.	Notes	584
17.19.	Further reading	584
17.20.	External links	585
18.	Advanced Mathematics	587
18.1.	Equation numbering	588
18.2.	Vertically aligning displayed mathematics	595
18.3.	Indented Equations	620

18.4.	Page breaks in math environments	621
18.5.	Boxed Equations	625
18.6.	Custom operators	628
18.7.	Advanced formatting	630
18.8.	Text in aligned math display	638
18.9.	Changing font size	640
18.10.	Forcing <code>\displaystyle</code> for all math in a document	643
18.11.	Adjusting vertical whitespace around displayed math	644
18.12.	Notes	645
19.	Labels and Cross-referencing	647
19.1.	Introduction	647
19.2.	Examples	652
19.3.	The <code>varioref</code> package	660
19.4.	The <code>hyperref</code> package and <code>\autoref{}</code>	661
19.5.	The <code>hyperref</code> package and <code>\nameref{}</code>	663
19.6.	The <code>hyperref</code> package and <code>\phantomsection</code>	664
19.7.	References	665
20.	Theorems	667
20.1.	Basic theorems	668
20.2.	Theorem counters	669

20.3.	Proofs	671
20.4.	Theorem styles	674
20.5.	Conflicts	677
20.6.	Notes	678
20.7.	External links	678
21.	Algorithms and Pseudocode	679
22.	Typesetting Algorithms	681
22.1.	Typesetting using the <code>algorithmic</code> package	681
22.2.	Typesetting using the <code>algorithmicx</code> package	683
22.3.	The <code>algorithm</code> environment	692
22.4.	Typesetting using the <code>program</code> package	697
22.5.	Source Code Formatting using the <code>Listings</code> package	704
23.	References	707
24.	Linguistics	709
24.1.	Enumerated examples	711
24.2.	Syntactic trees	720
24.3.	Glosses	737
24.4.	IPA characters	740
25.	References	747
26.	External links	749

IV. Special Documents and Pages	751
27. Indexing	753
27.1. Using makeidx	753
27.2. Using glossaries	758
27.3. Defining glossary entries	759
27.4. Defining terms	760
27.5. The letter class	762
27.6. Envelopes	766
27.7. Windowed envelopes	770
27.8. Reference: letter.cls commands	774
27.9. Sources	777
28. Presentations	779
28.1. The Beamer package	780
28.2. The powerdot package	807
28.3. References	808
28.4. Links	808
29. Teacher's Corner	811
29.1. Intro	811
29.2. The exam class	812
29.3. References	816

V. Advanced Topics	817
30. General Guidelines	819
30.1. Project structure	820
30.2. Getting LaTeX to process multiple files	821
30.3. The file <code>mystyle.sty</code>	835
30.4. The main document <code>document.tex</code>	837
30.5. Writing your document	841
30.6. External Links	842
31. Customizing LaTeX	843
31.1. New commands	844
31.2. New Environments	849
31.3. Command-line LaTeX	853
31.4. Creating your own package	854
31.5. Creating your own style	856
32. Packages	859
32.1. Using an existing package	861
32.2. Package documentation	863
32.3. Packages list	865
32.4. Creating packages	885
32.5. External resources	885

33.	Advanced Topics	887
33.1.	Adding your own counters	888
33.2.	Boxes	892
33.3.	Rules and Struts	897
34.	Collaborative Writing of LaTeX Documents	899
34.1.	Abstract	900
34.2.	Introduction	904
34.3.	Interchanging Documents	905
34.4.	The Version Control System <i>Subversion</i>	907
34.5.	Hosting LaTeX files in <i>Subversion</i>	910
34.6.	<i>Subversion</i> really makes the difference	914
34.7.	Managing collaborative bibliographies	924
34.8.	Conclusion	931
34.9.	Acknowledgements	934
34.10.	References	934
35.	Export To Other Formats	937
35.1.	Convert to PDF	939
35.2.	Convert to PostScript	943
35.3.	Convert to RTF	944
35.4.	Convert to HTML	946

35.5.	Convert to image formats	947
35.6.	Convert to plain text	951
36.	Internationalization	953
36.1.	Hyphenating	954
36.2.	Text encoding	957
36.3.	Output encoding	960
36.4.	Hyphenating languages	963
37.	Accents	997
37.1.	Text mode	998
37.2.	Math mode	1003
37.3.	Tabbing environment	1005
37.4.	Degree symbol for temperature and math	1006
37.5.	Related issues	1006
37.6.	External links	1008
38.	Tips and Tricks	1009
38.1.	Always writing LaTeX in roman	1009
38.2.	<i>id est</i> and <i>exempli gratia</i> (i.e. and e.g.)	1011
38.3.	Grouping Figure/Equation Numbering by Section	1012
38.4.	Generic header	1013
38.5.	Graphics and Graph editors	1015

38.6.	Spell-checking and Word Counting	1026
38.7.	New even page	1028
38.8.	Sidebar with information	1030
38.9.	Hide auxiliary files	1031
VI.	Appendix	1033
39.	Authors	1035
39.1.	Included books	1035
39.2.	Wiki users	1038
40.	Links	1039
41.	Installation	1047
41.1.	TeX and LaTeX	1048
41.2.	Editors	1051
41.3.	Bibliography management	1053
41.4.	Graphics tools	1055
41.5.	See also	1057
41.6.	References	1057
42.	Useful Measurement Macros	1059
42.1.	Units	1059
42.2.	Length 'macros'	1061
42.3.	Length manipulation macros	1064

42.4.	Samples	1066
42.5.	References	1068
42.6.	See also	1068
43.	Text Formatting	1069
43.1.	Spacing	1070
43.2.	Hyphenation	1076
43.3.	Quote-marks	1080
43.4.	Diacritics and accents	1083
43.5.	Space between Words and Sentences	1084
43.6.	Ligatures	1088
43.7.	Slash marks	1090
43.8.	Emphasizing Text	1092
43.9.	Fonts	1093
43.10.	Symbols and special characters	1115
43.11.	Notes and References	1125
44.	Sample LaTeX documents	1127
44.1.	General examples	1127
44.2.	Semantics of Programming Languages	1129
45.	Command Glossary	1131
45.1.	#	1131

45.2. A	1135
45.3. B	1137
45.4. C	1139
45.5. D	1142
45.6. E	1142
45.7. F	1143
45.8. G	1146
45.9. H	1146
45.10. I	1148
45.11. K	1150
45.12. L	1150
45.13. M	1153
45.14. N	1155
45.15. O	1158
45.16. P	1159
45.17. R	1161
45.18. S	1163
45.19. T	1167
45.20. U	1173
45.21. V	1174

46. More Bibliographies	1177
46.1. The example data	1178
46.2. The limits of BibTeX styles	1178
46.3. Natbib	1179
46.4. Citation	1184
47. Index	1191
47.1. A	1191
47.2. B	1192
47.3. C	1194
47.4. D	1195
47.5. E	1196
47.6. F	1197
47.7. G	1198
47.8. H	1199
47.9. I	1200
47.10. L	1201
47.11. M	1202
47.12. P	1203
47.13. Q	1204
47.14. R	1205

47.15.	S	1205
47.16.	T	1206
47.17.	U	1208
47.18.	V	1208
47.19.	W	1209
47.20.	X	1209
48.	Contributors	1211
	List of Figures	1271
49.	Licenses	1295
49.1.	GNU GENERAL PUBLIC LICENSE	1295
49.2.	GNU Free Documentation License	1304
49.3.	GNU Lesser General Public License	1311

Part I.

Getting Started

1. Introduction

1.1. What is TeX?

TeX¹ is a low-level markup and programming language created by Donald Knuth² to typeset documents attractively and consistently. The X, representing the Greek letter "chi", is pronounced as in the ending of the Scottish *loch* or the name *Bach*. Its name originates from the Greek word "τεχνολογία" (*technologia*), which translates as "technology" in English; its first syllable is

1 <http://en.wikibooks.org/wiki/TeX>

2 <http://en.wikipedia.org/wiki/Donald%20Knuth>

" $\tau\epsilon\chi$ ", similar to *TeX* in the Latin alphabet. According to a different approach the name originates from the Greek word " $\tau\acute{\epsilon}\chi\eta\eta$ " (*techni*), which translates as *art* in English. In the second case the first syllable is " $\tau\epsilon\chi$ " again. Within the typesetting system, its name is formatted as

UNKNOWN TEMPLATE TeX

Knuth started writing the TeX typesetting engine in 1977 to explore the potential of the digital printing equipment that was beginning to infiltrate the publishing industry at that time, especially in the hope that he could reverse the trend of deteriorating typographical quality that he saw affecting his own books and articles.

TeX is a programming language, in the sense that it supports the if-else construct, you can make calculations with it (that are performed while compiling the document), etc., but you would find it very hard to do anything else but typesetting with it. The fine control TeX offers makes it very powerful, but also difficult and time-consuming to use. TeX is renowned for being extremely sta-

ble, for running on many different kinds of computers, and for being virtually bug free.

As powerful as TeX might be, effective use requires a fairly high level of learning, and investment of time to build custom macros for text formatting. For those searching for a more straightforward solution, document preparation systems based on TeX, consisting of collections of pre-built macros, exist. **LaTeX** is one of the more popular of these. It is appreciated for being time saving, automating certain tasks and reducing user introduced errors. This is what makes it so popular. However, this comes at the cost of a sacrifice of flexibility. TeX remains the language of choice for those wishing a dedicated formatting language and almost unlimited functionality.

1.2. What is LaTeX?

LaTeX (pronounced either "Lah-tech" or "Lay-tech") is a macro package based on TeX created by Leslie Lamport³. Its purpose is to simplify TeX typesetting, especially for documents containing mathematical formulae. Within the typesetting system, its name is formatted as

UNKNOWN TEMPLATE LaTeX

. Many later authors have contributed extensions, called *packages* or *styles*, to LaTeX. Some of these are bundled with most TeX/LaTeX software distributions; more can be found in the Comprehensive TeX Archive Network (CTAN⁴).

3 <http://en.wikipedia.org/wiki/Leslie%20Lamport>

4 <http://www.ctan.org>

Since LaTeX comprises a group of TeX⁵ commands, LaTeX document processing is essentially programming. You create a text file in LaTeX markup, which LaTeX reads to produce the final document.

This approach has some disadvantages in comparison with a WYSIWYG⁶ (What You See Is What You Get) program such as Openoffice.org⁷ Writer or Microsoft Word⁸.

In LaTeX:

- You don't (usually) see the final version of the document when editing it.
- You generally need to know the necessary commands for LaTeX markup.
- It can sometimes be difficult to obtain a certain look for the document.

On the other hand, there are certain advantages to the LaTeX approach:

5 <http://en.wikibooks.org/wiki/TeX>

6 <http://en.wikipedia.org/wiki/WYSIWYG>

7 <http://en.wikipedia.org/wiki/Openoffice.org>

8 <http://en.wikipedia.org/wiki/Microsoft%20Word>

- Document sources can be read with any text editor and understood, unlike the complex binary and XML⁹ formats used with WYSIWYG programs.
- You can concentrate purely on the structure and contents of the document, not get caught up with superficial layout issues.
- You don't need to manually adjust fonts, text sizes, line heights, or text flow for readability, as LaTeX takes care of them automatically.
- In LaTeX the document structure is visible to the user, and can be easily copied to another document. In WYSIWYG applications it is often not obvious how a certain formatting was produced, and it might be impossible to copy it directly for use in another document.
- The layout, fonts, tables and so on are consistent throughout the document.
- Mathematical formulae can be easily typeset.
- Indexes, footnotes, citations and references are generated easily.
- Since the document source is plain text, tables, figures, equations, etc. can be generated programmatically with any language.
- You are forced to structure your documents correctly.

9 <http://en.wikipedia.org/wiki/XML>

The LaTeX-like approach can be called WYSIWYM¹⁰, i.e. What You See Is What You Mean: you can't see what the final version will look like while typing. Instead you see the logical structure of the document. LaTeX takes care of the formatting for you.

The LaTeX document is a plain text file containing the content of the document, with additional markup. When the source file is processed by the macro package, it can produce documents in several formats. LaTeX natively supports DVI¹¹ and PDF, but by using other software you can easily create PostScript, PNG, JPEG, etc.

¹⁰ <http://en.wikipedia.org/wiki/WYSIWYM>

¹¹ <http://en.wikipedia.org/wiki/DVI%20file%20format>

1.2.1. Philosophy of use

Flexibility and modularity

One of the most frustrating things beginners and even advanced users might encounter using LaTeX is the lack of flexibility regarding the document design and layout. If you want to design your document in a very specific way, you may have trouble accomplishing this. Keep in mind that LaTeX *does* the formatting for you, and mostly the right way. If it is not exactly what you desired, then the LaTeX way is at least not worse, if not better. One way to look at it is that LaTeX is a bundle of macros for TeX that aims to carry out everything regarding document formatting, so that the writer only needs to care about content. If you really want flexibility, use plain TeX instead.

One solution to this dilemma is to make use of the modular possibilities of LaTeX. You can build your own macros, or use macros developed by others. You are likely not the first person to face some particular formatting problem,

and someone who encountered a similar problem before may have published their solution as a package.

CTAN¹² is a good place to find many resources regarding TeX and derivative packages. It is the first place where you should begin searching.

Questions and documentation

Besides internet resources being plentiful, the best documentation source remains the official manual for every specific package, and the reference documentation, i.e. the *TeXbook* by D. Knuth and *LaTeX: A document preparation system* by L. Lamport.

12 <http://www.ctan.org/>

1.3. Other terms regarding TeX

When searching for information on LaTeX, you might also stumble upon XeTeX¹³, ConTeXt¹⁴, LuaTeX¹⁵ or other names with a -TeX suffix. XeTeX is a TeX engine which uses Unicode and supports widely popular `.ttf` and `.otf` fonts. LuaTeX is another engine with Lua scripting engine embedded aiming at making TeX internals more flexible.

ConTeXt is a TeX-based document preparation system (as LaTeX is) with a very consistent and easy syntax and support for pdfTeX, XeTeX and LuaTeX engines. It does not have the same objective as LaTeX however.

Therefore code written for TeX, LaTeX or ConTeXt are (mostly) not compatible. Additionally, engine-specific code (like font for XeTeX) may not be compiled by every engine.

13 <http://en.wikipedia.org/wiki/XeTeX>

14 <http://en.wikipedia.org/wiki/ConTeXt>

15 <http://en.wikipedia.org/wiki/LuaTeX>

1.4. Prerequisites

At a minimum, you'll need a TeX distribution, a good text editor and a DVI or PDF viewer. But, if this is the first time you are trying out LaTeX, you don't even need to install anything. Just create an user account at:

- ScribTeX.com¹⁶

and continue this tutorial in the next chapter. ScribTeX is a web based online editor for LaTeX documents with collaboration capabilities allowing you to experiment with LaTeX syntax without having to bother with installing and configuring a distribution and an editor. When you later feel that you would benefit from having a standalone LaTeX installation, you can return to this chapter and follow the instructions below.

16 <http://scribtex.com/>

1.4.1. Installing a distribution

If you want to use LaTeX locally on your computer, you generally need to install a TeX distribution. TeX distributions are packaged collections of packages and programs that enable you to typeset without having to manually fetch files and configure things. The recommended distributions for each of the major operating systems are:

- TeX Live¹⁷ is a major TeX distribution for Unix/Linux, Mac OS and Windows.
- MiKTeX¹⁸ is a Windows-specific distribution.
- MacTeX¹⁹ is a Mac OS-specific distribution based on TeX Live.

17 <http://www.tug.org/texlive/>

18 <http://www.miktex.org/>

19 <http://www.tug.org/mactex/>

Windows

TeX live and MikTeX have easy installers that take care of setting up the environment and downloading packages.

- TeX Live can be downloaded here²⁰.
- MiKTeX can be downloaded here²¹.

Linux

- Ubuntu has a 2009 version of TeX Live in the repositories, so you can use:
`sudo apt-get install texlive`. Here²² is a script to automate the installation of TeX Live 2010 on Ubuntu.

²⁰ <http://www.tug.org/texlive/acquire.html>

²¹ <http://miktex.org/>

²² <http://alexkrispin.wordpress.com/2010/10/12/a-script-to-automate-the-installation-of-texlive-2010/>

- Fedora has only 2007 version of TeX Live, but luckily there is a good repository here²³ that you can use to directly install the latest version: `yum install texlive` (plus `yum install texlive-latex` and any of the `texlive-scheme-*` packages).
- Gentoo provides an ebuild to install texlive using the command `emerge texlive`. http://www.gentoo.org/proj/en/tex/texlive-migration-guide.xml#doc_chap24 This is a meta-ebuild that pulls in many other packages depending on its USE flags.

If your distribution does not have the TeX Live packages, you should report a wish to the bug tracking system. In the worst case you will need to download TeX Live²⁵ yourself and run the installer by hand. It is not hard, but requires that you know how to use a console and make multiple choices during the installation.

²³ <http://fedoraproject.org/wiki/Features/TeXLive>

²⁴ http://www.gentoo.org/proj/en/tex/texlive-migration-guide.xml#doc_chap24

²⁵ <http://www.tug.org/texlive/acquire.html>

Mac OS

- Download MacTeX.mpkg.zip on the MacTeX page²⁶, unzip it and follow the instructions.

1.4.2. Getting a text editor

You will also need a text editor to write LaTeX code. You should use a text editor (e.g. Notepad), not a word processor (Word, Openoffice). Dedicated LaTeX editors are more useful than plain text editors, because they usually have autocompletion of commands, spell and error checking and handy macros.

26 <http://www.tug.org/mactex/>

TeXworks

TeXworks²⁷ is a dedicated TeX editor that is included in MiKTeX and TeX Live. It was developed with the idea that a simple interface is better than a cluttered one, and thus to make it easier for people in their early days with LaTeX to get to what they want to do: write their documents. TeXworks originally came about precisely because a math professor wanted his students to have a better initial experience with LaTeX.

You can install TeXworks with the package manager of your Linux distribution or choose it as an install option in the Windows or Mac installer.

27 <http://www.tug.org/texworks/>

LyX

LyX²⁸ is a popular LaTeX editor for Windows, Linux and Mac OS. It contains formula and table editors and shows visual clues of the final document on the screen enabling users to write LaTeX documents without worrying about the actual syntax.

Kile

Kile²⁹ is a LaTeX editor for KDE³⁰ (cross platform), providing a powerful GUI for editing multiple documents and compiling them with many different TeX compilers. Kile is based on Kate editor, has a quick access toolbar for symbols, document structure viewer, a console and customizable build options. Kile can be run in all operating systems that can run KDE.

28 <http://www.lyx.org/>

29 <http://kile.sourceforge.net/>

30 http://en.wikipedia.org/wiki/KDE_Software_Compilation_4

TeXmaker

TeXmaker³¹ is a cross-platform editor very similar to Kile in features and user interface. In addition it has its own PDF viewer.

TeXstudio

TeXstudio³² is a cross-platform open source LaTeX editor with interface similar to Texmaker.

31 <http://www.xmlmath.net/texmaker/>

32 <http://texstudio.sourceforge.net/>

TeXnicCenter

TeXnicCenter³³ is a popular free and open source LaTeX editor for Windows. It also has a similar user interface to TeXmaker and Kile.

WinEdt

WinEdt³⁴ is a powerful and versatile text editor with strong predisposition towards creation of LaTeX/TeX documents for Windows. It has been designed and configured to integrate with TeX Systems such as MiTeX or TeX Live. Its in-built macro helps in compiling the LaTeX source to the WYSIWYG-like DVI or PDF or PS and also in exporting the document to other mark-up languages as html or XML.

33 <http://www.texniccenter.org/>

34 <http://www.winedt.com/>

BaKoMa TeX

BaKoMa TeX³⁵ is an editor for Windows and Mac OS with WYSIWYG-like features. It takes care of compiling the LaTeX source and updating it constantly to view changes to document almost in real time.

TeXShop

TeXShop³⁶ is a TeXworks-like editor for Mac OS.

35 <http://bakoma-tex.com/menu/about.php>

36 <http://en.wikipedia.org/wiki/TeXShop>

gedit-latex-plugin

Gedit with `gedit-latex-plugin`³⁷ is also worth trying out for users of GNOME. GEdit is a cross-platform application for Windows, Mac, and Linux

Gummi

Gummi³⁸ is a LaTeX editor for Linux, which compiles the output of `pdflatex` in realtime and shows it on the right half of the screen.

37 <http://www.michaels-website.de/gedit-latex-plugin/>

38 <http://gummi.midnightcoding.org/>

Emacs

Emacs³⁹ is a general purpose text processing system. When used in combination with AucTex and RefTex (extensions that may be installed into the Emacs program), Emacs provides a complete LaTeX editing environment complete with table of contents view, document preview and many other features. Emacs is a very mature editing system with a unique set of keyboard commands.

Vim

Vim⁴⁰ is another general purpose text editor for a wide variety of platforms including UNIX, Mac OS X and Windows. A variety of extensions exist including LaTeX Box⁴¹ and Vim-LaTeX.

39 <http://www.gnu.org/software/emacs>

40 <http://en.wikipedia.org/wiki/Vim>

41 http://www.vim.org/scripts/script.php?script_id=3109

LaTeXila

LaTeXila⁴² is another text editor for Linux (Gnome).

1.4.3. Viewers

Finally, you will need a viewer for the files LaTeX outputs. Normally LaTeX saves the final document as a `.dvi` (Device independent file format), but you will rarely want it to. DVI files do not contain embedded fonts and many document viewers are unable to open them.

Usually you will use a LaTeX compiler like `pdflatex` to produce a PDF file directly, or a tool like `dvi2pdf` to convert the DVI file to PDF format. Then you can view the result with your preferred PDF viewer (Adobe Reader, Okular, Evince, Sumatra, Foxit).

42 <http://latexila.sourceforge.net/>

Practically all LaTeX distributions have a DVI viewer for viewing the default output of `latex`, and also tools such as `dvi2pdf` for converting the result automatically to PDF and PS formats.

1.4.4. Applications within a distribution

Here are the main programs you expect to find in any LaTeX distribution:

- `latex` compiler reads a LaTeX `.tex` file and creates a `.dvi`
- `pdflatex` compiler reads a LaTeX `.tex` file and creates a `.pdf`
- `dvi2ps` converts the `.dvi` file to `.ps` (postscript).
- `dvi2pdf` converts the `.dvi` file to `.pdf` (`dvi2pdfm` is an improved version).

Also `tex` and `pdftex` compilers are included, but you should be careful with using them, because they support only plain TeX. Note that since LaTeX is just a collection of macros for TeX if you compile a plain TeX document with a LaTeX compiler (such as `pdflatex`) it will work while the opposite is not

true: if you try to compile a LaTeX source with a TeX compiler you will get many errors.

When LaTeX was created, the only format it could create was DVI; later PDF support was added by `pdflatex`. PDF files can be created with both `pdflatex` and `dvilpdfm`. The output of `pdflatex` takes direct advantage of modern features of PDF such as hyperlinks and embedded fonts, which are not part of DVI. Passing through DVI imposes limitations of its older format. On the other hand, some packages, such as `PSTricks`, exploit the process of conversion to DVI, and therefore will not work with `pdflatex`. Some of those packages embed information in the DVI that doesn't appear when the DVI is viewed, but reemerges when the DVI is converted to another, newer format.

You would write your document slightly differently depending on the compiler you are using (`latex` or `pdflatex`). But as we will see later it is possible to add a sort of abstraction layer to hide the details of which compiler you're using, while the compiler can handle the translation itself.

The boxed red text represents the file formats, the blue text on the arrows represents the commands you have to use, the small dark green text under the boxes represents the image formats that are supported. Any time you pass through an arrow you lose some information, which might decrease the quality of your document. Therefore, in order to achieve the highest quality in your output file, you should choose the shortest route to reach your target format. This is probably the most convenient way to obtain an output in your desired format anyway. Starting from a LaTeX source, the best way is to use only *latex* for a DVI output or *pdflatex* for a PDF output, converting to PostScript only when it is necessary to print the document.

Most of the programs should be already within your LaTeX distribution; the others come with w:Ghostscript⁴³, which is a free and multi-platform software as well.

43 <http://en.wikipedia.org/wiki/Ghostscript>

Chapter ../Export To Other Formats/⁴⁴ discusses more about exporting LaTeX source to other file formats.

1.5. What next?

Now you should check that your editor and LaTeX distribution are functioning properly by trying to compile the first example source in the beginning of the next chapter⁴⁵.

Throughout this book you should also utilise other means for learning about LaTeX. Good sources are:

- the `#latex`⁴⁶ IRC channel on Freenode,

44 Chapter 35 on page 937

45 <http://en.wikibooks.org/wiki/...%2FAbsolute%20Beginners>

46 <http://webchat.freenode.net?channels=latex>

- the TeX Stack Exchange⁴⁷ Q&A,
- the TeX⁴⁸ FAQ,
- and the TeXample.net⁴⁹ Community.

47 <http://tex.stackexchange.com/>

48 <http://www.tex.ac.uk/cgi-bin/texfaq2html>

49 <http://www.texample.net/>

2. Basics

This tutorial is aimed at getting familiar with the bare bones of LaTeX¹.

Before starting, ensure you have LaTeX installed on your computer (see Installation² for instructions of what you will need). We begin by creating the source LaTeX file, and then take you through how to feed this file through the LaTeX system to produce quality output, such as postscript or PDF.

1 Chapter 1 on page 5

2 Chapter 41 on page 1047

2.1. The LaTeX source

LaTeX uses a markup language in order to describe document structure and presentation. LaTeX converts your source text, combined with the markup, into a high quality document. For the purpose of analogy, web pages work in a similar way: the HTML is used to describe the document, but it is your browser that presents it in its full glory - with different colours, fonts, sizes, etc.

The input for LaTeX is a plain ASCII text file. You can create it with any text editor. It contains the text of the document, as well as the commands that tell LaTeX how to typeset the text.

For the truly impatient, a minimal example looks something like the following (the commands will be explained later):

```
\documentclass{article}  
\begin{document}
```

```
Hello world!  
\end{document}
```

2.1.1. Spaces

"Whitespace" characters, such as blank or tab, are treated uniformly as "space" by LaTeX. Several consecutive whitespace characters are treated as one "space". Whitespace at the start of a line is generally ignored, and a single line break is treated as "whitespace." An empty line between two lines of text defines the end of a paragraph. Several empty lines are treated the same as one empty line. The text below is an example. On the left hand side is the text from the input file, and on the right hand side is the formatted output.

It does not matter whether you enter one or several spaces after a word.

An empty line starts a new paragraph.

It does not matter whether you enter one or several spaces after a word.
An empty line starts a new paragraph.

2.1.2. Special Characters

The following symbols are reserved characters that either have a special meaning under LaTeX or are unavailable in all the fonts. If you enter them directly in your text, they will normally not print, but rather make LaTeX do

things you did not intend.

```
# $ % ^ & _ { } ~ \
```

As you will see, these characters can be used in your documents all the same by adding a prefix backslash:

```
\# \$ \% \textasciicircum{} \& \_ \{ \} \~{} \textbackslashash{}
```

The other symbols and many more can be printed with special commands in mathematical formulae or as accents.

The backslash character `\` can *not* be entered by adding another backslash in front of it (`\\`); this sequence is used for line breaking. For introducing a backslash in math mode, you can use `\backslashash` instead.

The command `\~` produces a tilde which is placed over the next letter. For example `\~n` gives ñ. To produce just the character `~`, use `\~{ }` which places a `~` over an empty box.

Similarly, the command `\hat` produces a hat over the next character, for example `\hat{o}` produces \hat{o} . If you need in text to display the `\textasciicircum` symbol you have to use `\textasciicircum`.

If you want to insert text that might contain several particular symbols (such as URIs), you can consider using the `\verb` command, which will be discussed later in the section on formatting³.

2.1.3. LaTeX commands

LaTeX commands are case sensitive, and take one of the following two formats:

- They start with a backslash `\` and then have a name consisting of letters only. Command names are terminated by a space, a number or any other "non-letter".
- They consist of a backslash `\` and exactly one non-letter.

³ <http://en.wikibooks.org/wiki/LaTeX%2FFormatting>

Some commands need an argument, which has to be given between curly braces `{ }` after the command name. Some commands support optional parameters, which are added after the command name in square brackets `[]`. The general syntax is:

```
\commandname[option1,option2,...]{argument1}{argument2}...
```

2.1.4. LaTeX groups

A group is basically defined by a pair of braces. The range of commands put between braces is limited to them. The `\begingroup` and `\endgroup` commands are equivalent to opening brace and closing brace.

Example:

```
{
\bf This is bold.
}
This is no more bold.
```

For some commands it is important to restrict their range of action, and that's where groups come to be very useful.

2.1.5. LaTeX environments

Environments in LaTeX have a role that is quite similar to commands, but they usually have effect on a wider part of the document. Their syntax is:

```
\begin{environmentname}  
text to be influenced  
\end{environmentname}
```

Between the `\begin` and the `\end` you can put other commands and nested environments. The internal mechanism of environments defines a group, which makes its usage safe (no influence on the other parts of the document). In general, environments can accept arguments as well, but this feature is not commonly used and so it will be discussed in more advanced parts of the document.

Anything in LaTeX can be expressed in terms of commands and environments.

2.1.6. Comments

When LaTeX encounters a % character while processing an input file, it ignores the rest of the current line, the line break, and all whitespace at the beginning of the next line.

This can be used to write notes into the input file, which will not show up in the printed version.

```
This is an % stupid
% Better: instructive <----
example: Supercal%
         ifragilist%
icexpialidocious
```

This is an example: Supercalifragilisticexpialidocious

Note that the % character can be used to split long input lines that do not allow whitespace or line breaks, as with Supercalifragilisticexpialidocious above.

The core LaTeX language does not have a predefined syntax for commenting out regions spanning multiple lines. Refer to multi-line comments⁴ for simple workarounds.

2.2. Input File Structure

When LaTeX processes an input file, it expects it to follow a certain structure. Thus every input file must start with the command

⁴ http://en.wikibooks.org/wiki/LaTeX%2FFormatting%23Multi-line_comments

```
\documentclass{...}
```

This specifies what sort of document you intend to write. After that, you can include commands that influence the style of the whole document, or you can load packages that add new features to the LaTeX system. To load such a package you use the command

```
\usepackage{...}
```

When all the setup work is done, you start the body of the text with the command

```
\begin{document}
```

Now you enter the text mixed with some useful LaTeX commands. At the end of the document you add the

```
\end{document}
```


command, which tells LaTeX to call it a day. Anything that follows this command will be ignored by LaTeX. The area between `\documentclass` and `\begin{document}` is called the *preamble*.

2.2.1. Document Classes

The first information LaTeX needs to know when processing an input file is the type of document the author wants to create. This is specified with the `\documentclass` command.

```
\documentclass[options]{class}
```

Here `class` specifies the type of document to be created. The LaTeX distribution provides additional classes for other documents, including letters and slides. The options parameter customizes the behavior of the document class. The options have to be separated by commas.

Example: an input file for a LaTeX document could start with the line

```
\documentclass[11pt,twoside,a4paper]{article}
```

which instructs LaTeX to typeset the document as an article with a base font size of 11 points, and to produce a layout suitable for double sided printing on A4 paper.

Here are some document classes that can be used with LaTeX:

Document Classes	
article	for articles in scientific journals, presentations, short reports, program documentation, invitations, ...
IEEEtran	for articles with the IEEE Transactions format.
proc	a class for proceedings based on the article class.

Document Classes

minimal	is as small as it can get. It only sets a page size and a base font. It is mainly used for debugging purposes.
report	for longer reports containing several chapters, small books, thesis, ...
book	for real books
slides	for slides. The class uses big sans serif letters.

Document Classes

<code>memoir</code>	for changing sensibly the output of the document. It is based on the <code>book</code> class, but you can create any kind of document with it http://www.ctan.org/tex-archive/macros/latex/contrib/memoir/m
<code>letter</code>	for writing letters.
<code>beamer</code>	for writing presentations (see LaTeX/Presentations ⁶).

The most common options for the standard document classes are listed in the following table:

⁵ <http://www.ctan.org/tex-archive/macros/latex/contrib/memoir/memman.pdf>

⁶ Chapter 28 on page 779

Document Class Options

`10pt`, `11pt`, `12pt`

Sets the size of the main font in the document. If no option is specified, `10pt` is assumed.

`a4paper`, `letterpaper`, ...

Defines the paper size. The default size is `letterpaper`; However, many European distributions of TeX now come pre-set for A4, not Letter, and this is also true of all distributions of pdfLaTeX. Besides that, `a5paper`, `b5paper`, `executivepaper`, and `legalpaper` can be specified.

`fleqn`

Typesets displayed formulas left-aligned instead of centered.

Document Class Options

<code>leqno</code>	Places the numbering of formulas on the left hand side instead of the right.
<code>titlepage, notitlepage</code>	Specifies whether a new page should be started after the document title or not. The article class does not start a new page by default, while report and book do.
<code>onecolumn, twocolumn</code>	Instructs LaTeX to typeset the document in one column or two columns.

Document Class Options

`twoside`, `oneside`

Specifies whether double or single sided output should be generated. The classes `article` and `report` are single sided and the `book` class is double sided by default. Note that this option concerns the style of the document only. The option `twoside` does not tell the printer you use that it should actually make a two-sided printout.

`landscape`

Changes the layout of the document to print in landscape mode.

Document Class Options

`openright`, `openany`

Makes chapters begin either only on right hand pages or on the next page available. This does not work with the `article` class, as it does not know about chapters. The `report` class by default starts chapters on the next page available and the `book` class starts them on right hand pages.

Document Class Options

`draft`

makes LaTeX indicate hyphenation and justification problems with a small square in the right-hand margin of the problem line so they can be located quickly by a human. It also suppresses the inclusion of images and shows only a frame where they would normally occur.

For example, if you want a report to be in 12pt type on A4, but printed one-sided in draft mode, you would use:

```
\documentclass[12pt, a4paper, oneside, draft]{report}
```

2.2.2. Packages

While writing your document, you will probably find that there are some areas where basic LaTeX cannot solve your problem. If you want to include graphics, colored text or source code from a file into your document, you need to enhance the capabilities of LaTeX. Such enhancements are called packages. Packages are activated with the

```
\usepackage[options]{package}
```

command, where `package` is the name of the package and options is a list of keywords that trigger special features in the package. Some packages come with the LaTeX base distribution. Others are provided separately.

Modern TeX distributions come with a large number of packages pre-installed. If you are working on a Unix system, use the command `texdoc` for accessing package documentation. For more information, see the Packages⁷ section.

2.3. Our first document

Now we can create our first document. We will produce the absolute bare minimum that is needed in order to get some output; the well known **Hello World!** approach will be suitable here.

- Open your favorite text-editor. vim⁸, emacs⁹, Notepad++, and other text editors will have syntax highlighting that will help to write your files.
- Reproduce the following text in your editor. This is the LaTeX source.

```
% hello.tex - Our first LaTeX example!  
\documentclass{article}  
\begin{document}  
Hello World!  
\end{document}
```

8 <http://en.wikibooks.org/wiki/Learning%20the%20vi%20Editor%20FVim>

9 <http://en.wikibooks.org/wiki/emacs>

- Save your file as `hello.tex`.

When picking a name for your file, make sure it bears a `.tex` extension.

2.3.1. What does it all mean?

<pre>% hello.tex - Our first LaTeX example</pre>	<p>The first line is a <i>comment</i>. This is because it begins with the percent symbol (<code>%</code>); when LaTeX sees this, it simply ignores the rest of the line. Comments are useful for people to annotate parts of the source file. For example, you could put information about the author and the date, or whatever you wish.</p>
--	---

<pre>\documentclass{article}</pre>	<p>This line is a command and tells LaTeX to use the <code>article</code> document class. A document class file defines the formatting, which in this case is a generic article format. The handy thing is that if you want to change the appearance of your document, substitute article for another class file that exists.</p>
<pre>\begin{document}</pre>	<p>This line is the beginning of the environment called <code>document</code>; it alerts LaTeX that content of the document is about to commence. Anything above this command is known generally to belong in the <i>preamble</i>.</p>

Hello World!	This was the only actual line containing real content - the text that we wanted displayed on the page.
<pre>\end{document}</pre>	The document environment ends here. It tells LaTeX that the document source is complete, anything after this line will be ignored.

As we have said before, each of the LaTeX commands begins with a backslash (`\`). This is LaTeX's way of knowing that whenever it sees a backslash, to expect some commands. Comments are not classed as a command, since all they tell LaTeX is to ignore the line. Comments never affect the output of the document.

2.3.2. Generating the document

It is clearly not going to be the most exciting document you have ever seen, but we want to see it nonetheless. I am assuming that you are at a command prompt, already in the directory where `hello.tex` is stored. LaTeX itself does not have a GUI (graphical user interface), since it is just a program that crunches away at your input files, and produces either a DVI or PDF file. Some LaTeX installations feature a graphical front-end where you can click LaTeX into compiling your input file. On other systems there might be some typing involved, so here is how to coax LaTeX into compiling your input file on a text based system. Please note: this description assumes that you already have a working LaTeX installation on your computer.

1. Type the command: `latex hello` (the `.tex` extension is not required, although you can include it if you wish)
2. Various bits of info about LaTeX and its progress will be displayed. If all went well, the last two lines displayed in the console will be:

Output written on hello.dvi (1 page, 232 bytes) .
Transcript written on hello.log.

This means that your source file has been processed and the resulting document is called *hello.dvi*, which takes up 1 page and 232 bytes of space. Now you may view the DVI file. On Unix with X11 you can type `xdvi foo.dvi`, on Windows you can use a program called *yap* (yet another previewer). (Now *evince* and *okular*, the standard document viewers for many Linux distributions are able to view DVI files.)

This way you created the DVI file, but with the same source file you can create a PDF document. The steps are exactly the same as before, but you have to replace the command `latex` with `pdflatex`:

1. Type the command: `pdflatex hello` (as before, the `.tex` extension is not required)
2. Various bits of info about LaTeX and its progress will be displayed. If all went well, the last two lines displayed in the console will be:

Output written on hello.pdf (1 page, 5548 bytes).
Transcript written on hello.log.

you can notice that the PDF document is bigger than the DVI, even if it contains exactly the same information. The main differences between the DVI and PDF formats are:

- **DVI** needs less disk space and it is faster to create. It does not include the fonts within the document, so if you want the document to be viewed properly on another computer, there must be all the necessary fonts installed. It does not support any interactivity such as hyperlinks or animated images. DVI viewers are not very common, so you can consider using it for previewing your document while typesetting.
- **PDF** needs more disk space and it is slower to create, but it includes all the necessary fonts within the document, so you will not have any problem of portability. It supports internal and external hyperlinks. It also supports

advanced typographic features: hanging punctuation¹⁰, font expansion and margin kerning resulting in more flexibility available to the TeX engine and better looking output. Nowadays it is the *de facto* standard for sharing and publishing documents, so you can consider using it for the final version of your document.

About now, you saw you can create both DVI and PDF document from the same source. This is true, but it gets a bit more complicated if you want to introduce images or links. This will be explained in detail in the next chapters, but for now assume you can compile in both DVI and PDF without any problem.

Note, in this instance, due to the simplicity of the file, you only need to run the LaTeX command once. However, if you begin to create complex documents, including bibliographies and cross-references, etc, LaTeX needs to be executed multiple times to resolve the references. But this will be discussed in the future when it comes up.

¹⁰ <http://en.wikipedia.org/wiki/Hanging%20punctuation>

2.3.3. Compressed PDF

For a PDF output, you may have noticed that the output PDF file is not always the same size depending on the engine you used to compile the file. So `latex` → `dvips` → `ps2pdf` will usually be much smaller than `pdflatex`. So if you want `pdflatex` features along with a small output file size, you can use the Ghostscript command:

```
$ gs -dBATCH -dNOPAUSE -q -sDEVICE=pdfwrite
-sOutputFile="Compressed.pdf" "Original.pdf"
```

2.3.4. Picking suitable filenames

Never, ever use directories (folders) or file names that contain spaces. Although your operating system probably supports them, some don't, and they will only cause grief and tears with TeX. Make filenames as short or as long as you wish,

but strictly avoid spaces. Stick to lower-case letters without accents (a-z), the digits 0-9, the hyphen (—), and only one full point or period (.) to separate the file extension (somewhat similar to the conventions for a good Web URL): it will let you refer to TeX files over the Web more easily and make your files more portable. Some operating systems do not distinguish between upper-case and lower-case letters, others do. Therefore it's best not to mix them.

2.3.5. Files you might encounter

When you work with LaTeX you will soon find yourself in a maze of files with various extensions and probably no clue. The following list explains the most common file types you might encounter when working with TeX:

Common file extensions in LaTeX
--

Common file extensions in LaTeX

<code>.aux</code>	<p>A file that transports information from one compiler run to the next. Among other things, the <code>.aux</code> file is used to store information associated with cross-references.</p>
<code>.bbl</code>	<p>Bibliography file output by BiBTeX and used by LaTeX</p>
<code>.bib</code>	<p>Bibliography database file</p>
<code>.blg</code>	<p>BiBTeX log file.</p>
<code>.bst</code>	<p>BiBTeX style file.</p>
<code>.cls</code>	<p>Class files define what your document looks like. They are selected with the <code>\documentclass</code> command.</p>

Common file extensions in LaTeX

`.dtx`

Documented TeX. This is the main distribution format for LaTeX style files. If you process a `.dtx` file you get documented macro code of the LaTeX package contained in the `.dtx` file.

`.ins`

The installer for the files contained in the matching `.dtx` file. If you download a LaTeX package from the net, you will normally get a `.dtx` and a `.ins` file. Run LaTeX on the `.ins` file to unpack the `.dtx` file.

`.fd`

Font description file telling LaTeX about new fonts.

Common file extensions in LaTeX

<code>.dvi</code>	Device Independent File. This is the main result of a LaTeX compile run with <i>latex</i> . You can look at its content with a DVI previewer program or you can send it to a printer with <i>dvips</i> or a similar application.
<code>.pdf</code>	Portable Document Format. This is the main result of a LaTeX compile run with <i>pdflatex</i> . You can look at its content or print it with any PDF viewer.
<code>.log</code>	Gives a detailed account of what happened during the last compiler run.

Common file extensions in LaTeX

<code>.toc</code>	Stores all your section headers. It gets read in for the next compiler run and is used to produce the table of contents.
<code>.lof</code>	This is like <code>.toc</code> but for the list of figures.
<code>.lot</code>	And again the same for the list of tables.
<code>.idx</code>	If your document contains an index. LaTeX stores all the words that go into the index in this file. Process this file with <code>makeindex</code> .
<code>.ind</code>	The processed <code>.idx</code> file, ready for inclusion into your document on the next compile cycle.

Common file extensions in LaTeX

<code>.ilg</code>	Logfile telling what makeindex did.
<code>.sty</code>	LaTeX Macro package. This is a file you can load into your LaTeX document using the <code>\usepackage</code> command.
<code>.tex</code>	LaTeX or TeX input file. It can be compiled with latex.
<code>.out</code>	hyperref package file, just one for the master file.

2.4. And what now?

2.4.1. Formatting text

See *Document Structure*¹¹ and the *Formatting* part.

2.4.2. Big Projects

See General Guidelines¹² for further explanations.

When working on big documents, you might want to split the input file into several parts. LaTeX has three commands to insert a file into another when building the document.

The simplest is the `\input` command:

11 Chapter 3 on page 77

12 Chapter 30 on page 819

```
\input{filename}
```

`\input` inserts the contents of another file, named *filename.tex*; note that the *.tex* extension is omitted. For all practical purposes, `\input` is no more than a simple, automated copy-and-paste of the source code in *filename.tex*.

The other main inclusion command is `\include`:

```
\include{filename}
```

This command is roughly equivalent to `\clearpage \input{filename} \clearpage`. The `\include` command is different from `\input` in that it's the output that is added instead of the commands from the other files. Therefore a new page will be created at every `\include` command, which makes it appropriate to use it for large entities such as book chapters. The file extension of the *.tex* file which is to be included must also be omitted from the filename, unlike with `\input` for

which it is optional. This is because `\include` also creates an appropriate .aux file and needs the name without extension to achieve this.

Very large documents (that usually include many files) take a very long time to compile, and most users find it convenient to test their last changes by including only the files they have been working on. One option is to hunt down all `\include` commands in the inclusion hierarchy and to comment them out:

```
%\include{filename1}  
\include{filename2}  
\include{filename3}  
%\include{filename4}
```

In this case, the user wants to include only *filename2.tex* and *filename3.tex*. If the inclusion hierarchy is intricate, commenting can become error-prone: page numbering will change, and any cross references won't work. A better alternative is to retain the include calls and use the `\includeonly` command *in the preamble*:

```
\includeonly{filename2,filename3}
```

This way, only `\include` commands for the specified files will be executed, and inclusion will be handled in only one place. Note that there must be no spaces between the filenames and the commas.

Remember that the input file should omit all the commands referring to the main document structure, which should be kept in the original document file. This includes lines containing `usepackages`, `document class`, and everything but the code strictly referring to the section that is to be included. In this way, you'll avoid finding characters of your code in the output document, or worse, not finding anything after the included file, in case you forget to erase the

```
\end{document}
```

line of your included file.

2.4.3. Working in a team

See chapter ../Collaborative Writing of LaTeX Documents/¹³.

3. Document Structure

The main point of writing a text is to convey ideas, information, or knowledge to the reader. The reader will understand the text better if these ideas are well-structured, and will see and feel this structure much better if the typographical form reflects the logical and semantic structure of the content.

LaTeX is different from other typesetting systems in that you just have to tell it the logical and semantical structure of a text. It then derives the typographical form of the text according to the “rules” given in the document class file and in various style files. LaTeX allows users to structure their documents with a variety of hierarchal constructs, including chapters, sections, subsections and paragraphs.

3.1. The document environment

After the document class¹ declaration, the text of your document is enclosed between two commands which identify the beginning and end of the actual document:

```
\documentclass[11pt,a4paper,oneside]{report}

\begin{document}
...
\end{document}
```

You would put your text where the dots are. The reason for marking off the beginning of your text is that LaTeX allows you to insert extra setup specifications before it (where the blank line is in the example above: we'll be using this soon). The reason for marking off the end of your text is to provide

a place for LaTeX to be programmed to do extra stuff automatically at the end of the document, like making an index.

A useful side-effect of marking the end of the document text is that you can store comments or temporary text underneath the `\end{document}` in the knowledge that LaTeX will never try to typeset them:

```
...  
\end{document}
```

3.1.1. Preamble

The *preamble* is everything from the start of the LaTeX source file until the `\begin{document}` command. It normally contains commands that affect the entire document.

% simple.tex - A simple article to illustrate document structure.

```
\documentclass{article}  
\usepackage{mathpmtmx}  
  
\begin{document}
```

The first line is a comment (as denoted by the `%` sign). The `\documentclass` command takes an argument, which in this case is `article`, because that's the type of document we want to produce. It is also possible to create your own, as is often done by journal publishers, who simply provide you with their own class file, which tells LaTeX how to format your content. But we'll be happy with the standard `article` class for now. `\usepackage` is an important command that tells LaTeX to utilize some external macros. In this instance, we specified `mathpmtmx` which means LaTeX will use the Postscript Times type 1 font instead of the default ComputerModern font. And finally, the `\begin{document}`. This strictly isn't part of the preamble, but I'll put it here anyway, as it implies the end of the preamble by nature of stating that the document is now starting.

3.1.2. Top Matter

At the beginning of most documents there will be information about the document itself, such as the title and date, and also information about the authors, such as name, address, email etc. All of this type of information within LaTeX is collectively referred to as *top matter*. Although never explicitly specified (there is no `\topmatter` command) you are likely to encounter the term within LaTeX documentation.

A simple example:

```
\documentclass[11pt,a4paper,oneside]{report}

\begin{document}
\title{How to Structure a LaTeX Document}
\author{Andrew Roberts}
\date{December 2004}
\maketitle
\end{document}
```

The `\title`, `\author`, and `\date` commands are self-explanatory. You put the title, author name, and date in curly braces after the relevant command. The title and author are usually compulsory (at least if you want LaTeX to write the title automatically); if you omit the `\date` command, LaTeX uses today's date by default. You always finish the top matter with the `\maketitle` command, which tells LaTeX that it's complete and it can typeset the title according to the information you have provided and the class (style) you are using. If you omit `\maketitle`, the titling will never be typeset (unless you write your own).

Here is a more complicated example:

```
\title{How to Structure a \LaTeX{} Document}
\author{Andrew Roberts}
School of Computing,
University of Leeds,
Leeds,
United Kingdom,
LS2 1HE
\texttt{andy@comp.leeds.ac.uk} <!---->
```

```
\date{\today}  
\maketitle
```

as you can see, you can use commands as arguments of `\title` and the others. The double backslash (`\`) is the LaTeX command for forced linebreak. LaTeX normally decides by itself where to break lines, and it's usually right, but sometimes you need to cut a line short, like here, and start a new one.

If there are two authors separate them with the `\and` command:

```
\title{Our Fun Document}  
\author{Jane Doe \and John Doe}  
\date{\today}  
\maketitle
```

If you are provided with a class file from a publisher, or if you use the AMS article class (`amsart`), then you can use several different commands to enter author information. The email address is at the end, and the `\texttt` commands formats the email address using a mono-spaced font. The built-in command called `\today` will be replaced with the current date when pro-

cessed by LaTeX. But you are free to put whatever you want as a date, in no set order. If braces are left empty, then the date is omitted.

Using this approach, you can create only basic output whose layout is very hard to change. If you want to create your title freely, see the Title Creation² section.

3.1.3. Abstract

As most research papers have an abstract, there are predefined commands for telling LaTeX which part of the content makes up the abstract. This should appear in its logical order, therefore, after the top matter, but before the main sections of the body. This command is available for the document classes *article* and *report*, but not *book*.

`\documentclass{article}`

```
\begin{document }  
  
\begin{abstract }  
Your abstract goes here...  
...  
\end{abstract }  
...  
\end{document }
```

By default, LaTeX will use the word "Abstract" as a title for your abstract, if you want to change it into anything else, e.g. "Executive Summary", add the following line before your abstract:

```
\renewcommand{\abstractname}{Executive Summary}
```

3.1.4. Sectioning Commands

The commands for inserting sections are fairly intuitive. Of course, certain commands are appropriate to different document classes. For example, a book

has chapters but an article doesn't. Here is an edited version of some of the structure commands in use from *simple.tex*.

```
\section{Introduction}  
This section's content...
```

```
\section{Structure}  
This section's content...
```

```
\subsection{Top Matter}  
This subsection's content...
```

```
\subsection{Article Information}  
This subsection's content...
```

Notice that you do not need to specify section numbers; LaTeX will sort that out for you. Also, for sections, you do not need to markup which content belongs to a given block, using `\begin` and `\end` commands, for example. LaTeX provides 7 levels of depth for defining sections:

Command	Level	Comment
---------	-------	---------

Command	Level	Comment
<code>\part{part}</code>	-1	not in letters
<code>\chapter{chapter}</code>	0	only books and reports
<code>\section{section}</code>	1	not in letters
<code>\subsection{subsection}</code>	2	not in letters
<code>\subsubsection{subsubsection}</code>	3	not in letters
<code>\paragraph{paragraph}</code>	4	not in letters
<code>\subparagraph{subparagraph}</code>	5	not in letters

All the titles of the sections are added automatically to the table of contents (if you decide to insert one). But if you make manual styling changes to your heading, for example a very long title, or some special line-breaks or unusual font-play, this would appear in the Table of Contents as well, which you almost certainly don't want. LaTeX allows you to give an optional extra version of the heading text which only gets used in the Table of Contents and any running heads, if they are in effect. This optional alternative heading goes in [square brackets] before the curly braces:

```
\section[Effect on staff turnover]{An analysis of the
effect of the revised recruitment policies on staff
turnover at divisional headquarters}
```

Section numbering

Numbering of the sections is performed automatically by LaTeX, so don't bother adding them explicitly, just insert the heading you want between the curly braces. Parts get roman numerals (Part I, Part II, etc.); chapters and sections get decimal numbering like this document, and appendices (which are just a special case of chapters, and share the same structure) are lettered (A, B, C, etc.). You can change the depth to which section numbering occurs, so you can turn it off selectively. By default it is set to 2. If you only want parts, chapters, and sections numbered, not subsections or subsections etc., you can change the value of the `secnumdepth` counter using the `\setcounter` command, giving the depth level from the previous table. For example, if you want to change it to "1":

```
\setcounter{secnumdepth}{1}
```

A related counter is `tocdepth`, which specifies what depth to take the Table of Contents to. It can be reset in exactly the same way as `secnumdepth`. For example:

```
\setcounter{tocdepth}{3}
```

To get an unnumbered section heading which does not go into the Table of Contents, follow the command name with an asterisk before the opening curly brace:

```
\subsection*{Introduction}
```

All the divisional commands from `\part*` to `\subparagraph*` have this "starred" version which can be used on special occasions for an unnumbered heading when the setting of `secnumdepth` would normally mean it would be numbered.

If you want the unnumbered section to be in the table of contents anyway, use the `\addcontentsline` command like this:

```
\section*{Introduction}
\addcontentsline{toc}{section}{Introduction}
```

Note if you use pdf bookmarks you will need to add a phantom section so that bookmark will lead to the correct place in the document:

```
\phantomsection
\addcontentsline{toc}{section}{Introduction}
\section*{Introduction}
```

For chapters you will also need to clear the page (this will also correct page numbering in the ToC):

```
\cleardoublepage
\phantomsection
\addcontentsline{toc}{chapter}{Bibliography}
```

```
\bibliographystyle{unsrt}  
\bibliography{my_bib_file}
```

The value where the section numbering starts from can be set with the following command:

```
\setcounter{section}{4}
```

The next section after this command will now be numbered 5.

Any counter can be incremented/decremented with the following command:

```
\addtocounter{counter}{integer}
```

The `\phantomsection` command is defined in the `hyperref` package.

3.1.5. Appendices

The separate numbering of appendices is also supported by LaTeX. The `\appendix` macro can be used to indicate that following sections or chapters are to be numbered as appendices.

In the report or book classes this gives:

```
\appendix
\chapter{First Appendix}
```

For the article class use:

```
\appendix
\section{First Appendix}
```

Only use the `\appendix` macro once for all appendices.

3.1.6. Ordinary paragraphs

Paragraphs of text come after section headings. Simply type the text and leave a blank line between paragraphs. The blank line means "start a new paragraph here": it does **not** mean you get a blank line in the typeset output. For formatting paragraph indents and spacing between paragraphs, refer to the Formatting³ section.

3.1.7. Table of contents

All auto-numbered headings get entered in the Table of Contents (ToC) automatically. You don't have to print a ToC, but if you want to, just add the command `\tableofcontents` at the point where you want it printed (usually after the Abstract or Summary).

3 http://en.wikibooks.org/wiki/LaTeX%2FFormatting%23Paragraph_Indents

Entries for the ToC are recorded each time you process your document, and reproduced the next time you process it, so you need to re-run LaTeX one extra time to ensure that all ToC pagenumber references are correctly calculated. We've already seen how to use the optional argument to the sectioning commands to add text to the ToC which is slightly different from the one printed in the body of the document. It is also possible to add extra lines to the ToC, to force extra or unnumbered section headings to be included.

The commands `\listoffigures` and `\listoftables` work in exactly the same way as `\tableofcontents` to automatically list all your tables and figures. If you use them, they normally go after the `\tableofcontents` command. The `\tableofcontents` command normally shows only numbered section headings, and only down to the level defined by the `tocdepth` counter, but you can add extra entries with the `\addcontentsline` command. For example if you use an unnumbered section heading command to start a preliminary piece of text like a Foreword or Preface, you can write:

```
\subsection*{Preface}
\addcontentsline{toc}{subsection}{Preface}
```

This will format an unnumbered ToC entry for "Preface" in the "subsection" style. You can use the same mechanism to add lines to the List of Figures or List of Tables by substituting `lof` or `lot` for `toc`. If the `hyperref` package is used and the link does not point correct to the chapter, the command `\phantomsection` in combination with `\clearpage` or `\cleardoublepage` can be used (see also `Labels_and_Cross-referencing`⁴):

```
\cleardoublepage
\phantomsection
\addcontentsline{toc}{chapter}{List of Figures}
\listoffigures
```

To change the title of the TOC, you have to paste this command

```
\renewcommand{\contentsname}{<New table of contents title>}
```

in your document preamble. The List of Figures (LoF) and List of Tables (LoT) names can be changed by replacing the

```
\contentsname
```

with

```
\listfigurename
```

for LoF and

```
\listtablename
```

for LoT.

Depth

The default ToC will list headings of level 3 and above. To change how deep the table of contents displays automatically the following command can be used in the preamble:

```
\setcounter{tocdepth}{4}
```

This will make the table of contents include everything down to paragraphs. The levels are defined above on this page. Note that this solution does not permit changing the depth dynamically.

You can change the depth of specific section type, which could be useful for PDF bookmarks (if you are using the `hyperref` package) :

```
\makeatletter
\newcommand*{\toclevel@chapter}{-1} % Put chapter depth at the same
level as \part.
\chapter{EpiLog}
\newcommand*{\toclevel@chapter}{0} % Put chapter depth back to its
default value.
\makeatother
```

In order to further tune the display or the numbering of the table of contents, for instance if the appendix should be less detailed, you can make use of the `tocvsec2` package (`CTAN5`, `doc6`).

3.1.8. The Bibliography

Any good research paper will have a whole list of references. There are two ways to insert your references into LaTeX:

- you can embed them within the document itself. It's simpler, but it can be time-consuming if you are writing several papers about similar subjects so that you often have to cite the same books.

5 <http://www.ctan.org/pkg/tocvsec2>

6 <http://mirror.ctan.org/macros/latex/contrib/tocvsec2/tocvsec2.pdf>

- you can store them in an external BibTeX file ⁷ and then link them via a command to your current document and use a BibTeX style⁸ to define how they appear. This way you can create a small database of the references you might use and simply link them, letting LaTeX work for you.

In order to know how to add the bibliography to your document, see the Bibliography Management⁹ section.

7 <http://www.bibtex.org>

8 <http://www.cs.stir.ac.uk/~kjt/software/latex/showbst.html>

9 Chapter 5 on page 117

4. Errors and Warnings

LaTeX describes what it's typesetting while it does it, and if it encounters something it doesn't understand or can't do, it will display a message saying what's wrong. It may also display warnings for less serious conditions.

Don't panic if you see error messages: it's very common to mistype or misspell commands, forget curly braces, type a forward slash instead of a backslash, or use a special character by mistake. Errors are easily spotted and easily corrected in your editor, and you can then run LaTeX again to check you have fixed everything. Some of the most common errors are described in next sections.

4.1. Error messages

The format of an error message is always the same. Error messages begin with an exclamation mark at the start of the line, and give a description of the error, followed by another line starting with the number, which refers to the line-number in your document file which LaTeX was processing when the error was spotted. Here's an example, showing that the user mistyped the `\tableofcontents` command:

```
! Undefined control sequence.  
1.6 \tableofcotnets
```

When LaTeX finds an error like this, it displays the error message and pauses. You must type one of the following letters to continue:

Key	Meaning
x	Stop immediately and exit the program.

Key	Meaning
q	Carry on quietly as best you can and don't bother me with any more error messages.
e	Stop the program but re-position the text in my editor at the point where you found the error (This only works if you're using an editor which LaTeX can communicate with).
h	Try to give me more help .
i	(followed by a correction) means input the correction in place of the error and carry on (This is only a temporary fix to get the file processed. You still have to make that correction in the editor).

Key	Meaning
r	r un in non-stop mode. Plow through any errors, unless too many pile up and it fails (100 errors).

Some systems (Emacs is one example) run LaTeX with a "nonstop" switch turned on, so it will always process through to the end of the file, regardless of errors, or until a limit is reached.

4.2. Warnings

Warnings don't begin with an exclamation mark: they are just comments by LaTeX about things you might want to look into, such as overlong or underrun lines (often caused by unusual hyphenations, for example), pages running short or long, and other typographical niceties (most of which you can ignore

until later). Unlike other systems, which try to hide unevennesses in the text (usually unsuccessfully) by interfering with the letter spacing, LaTeX takes the view that the author or editor should be able to contribute. While it is certainly possible to set LaTeX's parameters so that the spacing is sufficiently sloppy that you will almost never get a warning about badly-fitting lines or pages, you will almost certainly just be delaying matters until you start to get complaints from your readers or publishers.

4.3. Examples

Only a few common error messages are given here: those most likely to be encountered by beginners. If you find another error message not shown here, and it's not clear what you should do, ask for help.

Most error messages are self-explanatory, but be aware that the place where LaTeX spots and reports an error may be later in the file than the place where it actually occurred. For example if you forget to close a curly brace which

encloses, say, italics, LaTeX won't report this until something else occurs which can't happen until the curly brace is encountered (e.g. the end of the document!) Some errors can only be righted by humans who can read and understand what the document is supposed to mean or look like.

Newcomers should remember to check the list of special characters: a very large number of errors when you are learning LaTeX are due to accidentally typing a special character when you didn't mean to. This disappears after a few days as you get used to them.

4.3.1. Too many }'s

```
! Too many }'s.  
1.6 \date December 2004}
```

The reason LaTeX thinks there are too many }'s here is that the opening curly brace is missing after the `\date` control sequence and before the word

December, so the closing curly brace is seen as one too many (which it is!). In fact, there are other things which can follow the `\date` command apart from a date in curly braces, so LaTeX cannot possibly guess that you've missed out the opening curly brace until it finds a closing one!

4.3.2. Undefined control sequence

```
! Undefined control sequence.  
1.6 \dtæ  
{December 2004}
```

In this example, LaTeX is complaining that it has no such command ("control sequence") as `\dtæ`. Obviously it's been mistyped, but only a human can detect that fact: all LaTeX knows is that `\dtæ` is not a command it knows about: it's undefined. Mistypings are the most common source of errors. Some editors allow common commands and environments to be inserted using drop-down menus or icons, which may be used to avoid these errors.

4.3.3. Not in Mathematics Mode

! Missing \$ inserted

A character that can only be used in the mathematics was inserted in normal text. Either switch to mathematic mode via `\begin{math}...\end{math}` or use the 'quick math mode': `\ensuremath{math stuff}`

This can also happen if you use the wrong character encoding, for example using utf8 without `"\usepackage[utf8]{inputenc}"` or using iso8859-1 without `"\usepackage[latin1]{inputenc}"`, there are several character encoding formats, make sure to pick the right one.

4.3.4. Runaway argument

Runaway argument?
{December 2004 \maketitle

```
! Paragraph ended before \date was complete.  
<to be read again>  
\par  
1.8
```

In this error, the closing curly brace has been omitted from the date. It's the opposite of the error of too many }'s, and it results in `\maketitle` trying to format the title page while LaTeX is still expecting more text for the date! As `\maketitle` creates new paragraphs on the title page, this is detected and LaTeX complains that the previous paragraph has ended but `\date` is not yet finished.

4.3.5. Underfull hbox

```
Underfull \hbox (badness 1394) in paragraph  
at lines 28--30  
[[[]\LY1/brm/b/n/10 Bull, RJ: \LY1/brm/m/n/10  
Ac-count-ing in Busi-  
[94]
```


This is a warning that LaTeX cannot stretch the line wide enough to fit, without making the spacing bigger than its currently permitted maximum. The badness (0-10,000) indicates how severe this is (here you can probably ignore a badness of 1394). It says what lines of your file it was typesetting when it found this, and the number in square brackets is the number of the page onto which the offending line was printed. The codes separated by slashes are the typeface and font style and size used in the line. Ignore them for the moment.

This comes up if you force a linebreak, e.g., `\`, and have a return before it. Normally TeX ignores linebreaks, providing full paragraphs to ragged text. In this case it is necessary to pull the linebreak up one line to the end of the previous sentence.

4.3.6. Overfull hbox

```
[101]
Overfull \hbox (9.11617pt too wide) in paragraph
at lines 860--861
```

[] \LY1/brm/m/n/10 Windows, \LY1/brm/m/it/10 see
\LY1/brm/m/n/10 X Win-

An overfull \hbox means that there is a hyphenation or justification problem: moving the last word on the line to the next line would make the spaces in the line wider than the current limit; keeping the word on the line would make the spaces smaller than the current limit, so the word is left on the line, but with the minimum allowed space between words, and which makes the line go over the edge.

The warning is given so that you can find the line in the code that originates the problem (in this case: 860-861) and fix it. The line on this example is too long by a shade over 9pt. The chosen hyphenation point which minimizes the error is shown at the end of the line (Win-). Line numbers and page numbers are given as before. In this case, 9pt is too much to ignore (over 3mm), and a manual correction needs making (such as a change to the hyphenation), or the flexibility settings need changing.

If the "overflow" word includes a forward slash, such as "input/output", this should be properly typeset as "input\slash output". The use of \slash has the same effect as using the "/" character, except that it can form the end of a line (with the following words appearing at the start of the next line). The "/" character is typically used in units, such as "mm/year" character, which should not be broken over multiple lines.

It can also issue this warning if the \end{document} tag was not included and/or was deleted.

4.3.7. Missing package

```
! LaTeX Error: File 'paralisy.sty' not found.
Type X to quit or <RETURN> to proceed,
or enter new name. (Default extension: sty)
Enter file name:
```

When you use the `\usepackage` command to request LaTeX to use a certain package, it will look for a file with the specified name and the filetype `.sty`. In this case the user has mistyped the name of the `paralist` package, so it's easy to fix. However, if you get the name right, but the package is not installed on your machine, you will need to download and install it before continuing. If you don't want to affect the global installation of the machine, you can simply download from Internet the necessary `.sty` file and put it in the same folder of the document you are compiling.

4.3.8. Package babel Warning: No hyphenation patterns were loaded for the language X

Although this is a warning from the `Babel` package and not from LaTeX, this error is very common and (can) give some strange hyphenation (word breaking) problems in your document. Wrong hyphenation rules can decrease the neatness of your document.

Package babel Warning: No hyphenation patterns were loaded for (babel) the language 'Latin'
(babel) I will use the patterns loaded for \language=0 instead.

This can happen after the usage of: (see LaTeX/Internationalization¹)

```
\usepackage[Latin]{babel}
```

The solution is not difficult, just install the used language in your LaTeX distribution².

1 Chapter 36 on page 953
2 Chapter 41.1 on page 1048

4.3.9. Package babel Error: You haven't loaded the option X yet.

If you previously set the X language, and then decided to switch to Y, you will get this error. This may seem awkward, as there is obviously no error in your code if you did not change anything. The answer lies in the `.aux` file, where babel defined your language. If you try the compilation a second time, it should work. If not, delete the `.aux` file, then everything will work as usual.

4.3.10. Software that can check your .tex Code

There are several programs capable of checking LaTeX source, with the aim of finding errors or highlighting bad practice, and providing more help to (particularly novice) users than the built-in error messages.

- `nag` (www.ctan.org/tex-archive/macros/latex/contrib/nag³) is a LaTeX package designed to indicate the use of obsolete commands.

3 <http://www.ctan.org/tex-archive/macros/latex/contrib/nag>

- `lacheck` (www.ctan.org/tex-archive/support/lacheck⁴) is a consistency checker intended to spot mistakes in code. It is available as source code or compiled for Windows and OS/2
- `chktex` (baruch.ev-en.org/proj/chktex/⁵) is a LaTeX semantic checker available as source code for Unix-like systems.

4 <http://www.ctan.org/tex-archive/support/lacheck>

5 <http://baruch.ev-en.org/proj/chktex/>

5. Bibliography Management

For any academic/research writing, incorporating references into a document is an important task. Fortunately, LaTeX has a variety of features that make dealing with references much simpler, including built-in support for citing references. However, a much more powerful and flexible solution is achieved thanks to an auxiliary tool called BibTeX¹ (which comes bundled as standard with LaTeX).

BibTeX provides for the storage of all references in an external, flat-file database. This database can be linked to any LaTeX document, and citations

¹ <http://www.bibtex.org>

made to any reference that is contained within the file. This is often more convenient than embedding them at the end of every document written. There is now a centralized bibliography source that can be linked to as many documents as desired (write once, read many!). Of course, bibliographies can be split over as many files as one wishes, so there can be a file containing references concerning *General Relativity* and another about *Quantum Mechanics*. When writing about *Quantum Gravity* (QG), which tries to bridge the gap between these two theories, both of these files can be linked into the document, in addition to references specific to QG.

5.1. Embedded system

If you are writing only one or two documents and aren't planning on writing more on the same subject for a long time, maybe you don't want to waste time creating a database of references you are never going to use. In this case

you should consider using the basic and simple bibliography support that is embedded within LaTeX.

LaTeX provides an environment called `thebibliography` that you have to use where you want the bibliography; that usually means at the very end of your document, just before the `\end{document}` command. Here is a practical example:

```
\begin{thebibliography}{9}

\bibitem{lampport94}
  Leslie Lampport,
  \emph{\LaTeX: A Document Preparation System}.
  Addison Wesley, Massachusetts,
  2nd Edition,
  1994.

\end{thebibliography}
```

OK, so what is going on here? The first thing to notice is the establishment of the environment. `thebibliography` is a keyword that LaTeX recognizes

as everything between the begin and end tags as being data for the bibliography. The mandatory argument, which I supplied after the begin statement, is telling LaTeX how wide the item label will be when printed. Note however, that the number itself is not the parameter, but the number of digits is. Therefore, I am effectively telling LaTeX that I will only need reference labels of one character in length, which ultimately means no more than nine references in total. If you want more than nine, then input any two-digit number, such as '56' which allows up to 99 references.

Next is the actual reference entry itself. This is prefixed with the `\bibitem{cite_key}` command. The *cite_key* should be a unique identifier for that particular reference, and is often some sort of mnemonic consisting of any sequence of letters, numbers and punctuation symbols (although not a comma). I often use the surname of the first author, followed by the last two digits of the year (hence *lamport94*). If that author has produced more than one reference for a given year, then I add letters after, 'a', 'b', etc. But, you should do whatever works for you. Everything after the key is the reference itself. You need to type it as you want it to be presented. I have put the different parts

of the reference, such as author, title, etc., on different lines for readability. These linebreaks are ignored by LaTeX. I wanted the title to be in italics, so I used the `\emph{ }` command to achieve this.

5.2. Citations

To actually cite a given document is very easy. Go to the point where you want the citation to appear, and use the following: `\cite{cite_key}`, where the *cite_key* is that of the bibitem you wish to cite. When LaTeX processes the document, the citation will be cross-referenced with the bibitems and replaced with the appropriate number citation. The advantage here, once again, is that LaTeX looks after the numbering for you. If it were totally manual, then adding or removing a reference would be a real chore, as you would have to re-number all the citations by hand.

Instead of WYSIWYG editors, typesetting systems like `\TeX{ }` or `\LaTeX{ }` `\cite{lampport94}` can be used.

5.2.1. Referring More Specific

Sometimes you want to refer to a certain page, figure or theorem in a text book. For that you can use the arguments to the `\cite` command:

```
\cite[p.~215]{citation01}
```

The argument, "p. 215", will show up inside the same brackets. Note the tilde in [p.~215], which replaces the end-of-sentence spacing with a non-breakable inter-word space. There are two reasons: end-of-sentence spacing is too wide, and "p." should not be separated from the page number.

5.2.2. Multiple Citations

When a sequence of multiple citations are needed, you should use a single `\cite{}` command. The citations are then separated by commas. Note that you must not use spaces between the citations. Here's an example:

```
\cite{citation01,citation02,citation03}
```

The result will then be shown as citations inside the same brackets.

5.2.3. No Cite

If you only want a reference to appear in the bibliography, but not where it is referenced in the main text, then the `\nocite{}` command can be used, for example:

```
Lampport showed in 1995 something... \nocite{lampport95}.
```

A special version of the command, `\nocite{*}`, includes all entries from the database, whether they are referenced in the document or not.

5.2.4. Natbib

Natbib commands	
Citation command	Output
<code>\citet{goossens93}</code>	Goossens et al. (1993)
<code>\citep{goossens93}</code>	(Goossens et al., 1993)
<code>\citet*{goossens93}</code>	Goossens, Mittlebach, and Samarin (1993)
<code>\citep*{goossens93}</code>	(Goossens, Mittlebach, and Samarin, 1993)
<code>\citeauthor{goossens93}</code>	Goossens et al.
<code>\citeauthor*{goossens93}</code>	Goossens, Mittlebach, and Samarin
<code>\citeyear{goossens93}</code>	1993
<code>\citeyearpar{goossens93}</code>	(1993)
<code>\citealt{goossens93}</code>	Goossens et al. 1993
<code>\citealp{goossens93}</code>	Goossens et al., 1993

Using the standard LaTeX bibliography support, you will see that each reference is numbered and each citation corresponds to the numbers. The numeric style of citation is quite common in scientific writing. In other disciplines, the author-year style, e.g., (Roberts, 2003), such as *Harvard* is preferred, and is in fact becoming increasingly common within scientific publications. A discussion about which is best will not occur here, but a possible way to get such an output is by the `natbib` package. In fact, it can supersede LaTeX's own citation commands, as `Natbib` allows the user to easily switch between `Harvard` or `numeric`.

The first job is to add the following to your preamble in order to get LaTeX to use the `Natbib` package:

```
\usepackage[options]{natbib}
```

Also, you need to change the bibliography style file to be used, so edit the appropriate line at the bottom of the file so that it reads: `\bibliographystyle{plainnat}`. Once done, it is basically a matter

of altering the existing `\cite` commands to display the type of citation you want.

Natbib-compatible styles		
Style	Source	Description
plainnat	Provided	natbib-compatible version of plain
abbrvnat	Provided	natbib-compatible version of abbrv
unsrtnat	Provided	natbib-compatible version of unsrt
apsrev	REVTeX 4 home page ²	natbib-compatible style for Physical Review journals

² <http://authors.aps.org/revtex4/>

Natbib-compatible styles		
Style	Source	Description
rmpaps	REVTeX 4 home page ³	natbib-compatible style for Review of Modern Physics journals
IEEEtranN	TeX Catalogue entry ⁴	natbib-compatible style for IEEE publications
achemso	TeX Catalogue entry ⁵	natbib-compatible style for American Chemical Society journals

³ <http://authors.aps.org/revtex4/>

⁴ <http://www.ctan.org/tex-archive/help/Catalogue/entries/ieeetran.html>

⁵ <http://www.ctan.org/tex-archive/help/Catalogue/entries/achemso.html>

Natbib-compatible styles

Style	Source	Description
rsc	TeX Catalogue entry ⁶	natbib-compatible style for Royal Society of Chemistry journals

Natbib options

Option	Meaning
round	Parentheses () (default)
square	Square brackets []
curly	Curly braces {}
angle	Angle brackets <>
colon	multiple citations are separated by semi-colons (default)

⁶ <http://www.ctan.org/tex-archive/help/Catalogue/entries/rsc.html>

Natbib options	
Option	Meaning
comma	multiple citations are separated by commas
authoryear	author year style citations (default)
numbers	numeric citations
super	superscripted numeric citations
sort	multiple citations are sorted into the order in which they appear in the references section
sort&compress	as sort, compressing multiple numeric citations where possible
longnamesfirst	the first citation of any reference will use the starred variant (full author list), subsequent citations will use the abbreviated <i>et al.</i> style

Natbib options	
Option	Meaning
sectionbib	for use with the chapterbib package. redefines \thebibliography to issue \section* instead of \chapter*
nonamebreak	keeps all the authors' names in a citation on one line to fix some hyperref problems - causes overflow boxes

The main commands simply add a *t* for 'textual' or *p* for 'parenthesized', to the basic `\cite` command. You will also notice how Natbib by default will compress references with three or more authors to the more concise *Ist surname et al* version. By adding an asterisk (*), you can override this default and list all authors associated with that citation. There are some other specialized commands that Natbib supports, listed in the table here. Keep in mind that for instance `abbrvnat` does not support `\citet*` and will automatically choose between all authors and *et al.*

The final area that I wish to cover about Natbib is customizing its citation style. There is a command called `\bibpunct` that can be used to override the defaults and change certain settings. For example, I have put the following in the preamble:

```
\bibpunct{(){}}{;}{a}{r}{r}{}
```

The command requires six mandatory parameters.

1. The symbol for the opening bracket.
2. The symbol for the closing bracket.
3. The symbol that appears between multiple citations.
4. This argument takes a letter:
 - *n* - numerical style.
 - *s* - numerical superscript style.
 - *any other letter* - author-year style.
5. The punctuation to appear between the author and the year (in parenthesis case only).

6. The punctuation used between years, in multiple citations when there is a common author. e.g., (Chomsky 1956, 1957). If you want an extra space, then you need `{, ~}`.

Some of the options controlled by `\bibpunct` are also accessible by passing options to the `natbib` package when it is loaded. These options also allow some other aspect of the bibliography to be controlled, and can be seen in the table (right).

So as you can see, this package is quite flexible, especially as you can easily switch between different citation styles by changing a single parameter. Do have a look at the `Natbib` manual⁷, it's a short document and you can learn even more about how to use it.

7 <http://www.ctex.org/documents/packages/bibref/natbib.pdf>

5.3. BibTeX

I have previously introduced the idea of embedding references at the end of the document, and then using the `\cite` command to cite them within the text. In this tutorial, I want to do a little better than this method, as it's not as flexible as it could be. Which is why I wish to concentrate on using BibTeX⁸.

A BibTeX database is stored as a *.bib* file. It is a plain text file, and so can be viewed and edited easily. The structure of the file is also quite simple. An example of a BibTeX entry:

```
@article{greenwade93,  
  author = "George D. Greenwade",  
  title   = "The {C}omprehensive {T}ex {A}rchive {N}etwork  
  ({CTAN})",  
  year   = "1993",  
  journal = "TUGBoat",  
  volume = "14",
```

8 <http://en.wikipedia.org/wiki/BibTeX>


```
number = "3",  
pages = "342--351"
```

```
}
```

Each entry begins with the declaration of the reference type, in the form of `@type`. BibTeX knows of practically all types you can think of, common ones are: *book*, *article*, and for papers presented at conferences, there is *proceedings*. In this example, I have referred to an article within a journal.

After the type, you must have a left curly brace '{' to signify the beginning of the reference attributes. The first one follows immediately after the brace, which is the *citation key*, or the *BibTeX key*. This key must be unique for all entries in your bibliography. It is this identifier that you will use within your document to cross-reference it to this entry. It is up to you as to how you wish to label each reference, but there is a loose standard in which you use the author's surname, followed by the year of publication. This is the scheme that I use in this tutorial.

Next, it should be clear that what follows are the relevant fields and data for that particular reference. The field names on the left are BibTeX keywords⁹. They are followed by an equals sign (=) where the value for that field is then placed. BibTeX expects you to explicitly label the beginning and end of each value. I personally use quotation marks ("), however, you also have the option of using curly braces ('{', '}'). But as you will soon see, curly braces have other roles, within attributes, so I prefer not to use them for this job as they can get more confusing. A notable exception is when you want to use characters with umlauts (ü, ö, etc), since their notation¹⁰ is in the format `\{ö\}`, and the quotation mark will close the one opening the field, causing an error in the parsing of the reference. Using `\usepackage[utf8]{inputenc}` in the preamble to the `.tex` source file can get round this, as the accented characters can just be stored in the `.bib` file without any need for special markup. This allows a consistent format to be kept throughout the `.bib` file, avoiding the need to use braces when there are umlauts to consider.

9 <http://en.wikipedia.org/wiki/BibTeX%23Bibliographic%20information%20file>

Remember that each attribute must be followed by a comma to delimit one from another. You do not need to add a comma to the last attribute, since the closing brace will tell BibTeX that there are no more attributes for this entry, although you won't get an error if you do.

It can take a while to learn what the reference types are, and what fields each type has available (and which ones are required or optional, etc). So, look at this entry type reference¹¹ and also this field reference¹² for descriptions of all the fields. It may be worth bookmarking or printing these pages so that they are easily at hand when you need them.

5.3.1. Authors

BibTeX can be quite clever with names of authors. It can accept names in *forename surname* or *surname, forename*. I personally use the former, but

11 <http://newton.ex.ac.uk/tex/pack/bibtex/btxdoc/node6.html>

12 <http://newton.ex.ac.uk/tex/pack/bibtex/btxdoc/node7.html>

remember that the order you input them (or any data within an entry for that matter) is customizable and so you can get BibTeX to manipulate the input and then output it however you like. If you use the *forename surname* method, then you must be careful with a few special names, where there are compound surnames, for example "John von Neumann". In this form, BibTeX assumes that the last word is the surname, and everything before is the forename, plus any middle names. You must therefore manually tell BibTeX to keep the 'von' and 'Neumann' together. This is achieved easily using curly braces. So the final result would be "John {von Neumann}". This is easily avoided with the *surname, forename*, since you have a comma to separate the surname from the forename.

Secondly, there is the issue of how to tell BibTeX when a reference has more than one author. This is very simply done by putting the keyword *and* in between every author. As we can see from another example:

```
@book{goossens93,  
  author = "Michel Goossens and Frank Mittelbach and Alexander  
  Samarin",
```

```
title    = "The LaTeX Companion",  
year    = "1993",  
publisher = "Addison-Wesley",  
address = "Reading, Massachusetts"  
}
```

This book has three authors, and each is separated as described. Of course, when BibTeX processes and outputs this, there will only be an 'and' between the penultimate and last authors, but within the .bib file, it needs the *ands so that it can keep track of the individual authors.*

5.3.2. Standard templates

Be careful if you copy the following templates, the % sign is not valid to comment out lines in bibtex files. If you want to comment out a line, you have to put it outside the entry.

@article

An article from a magazine or a journal.

- Required fields: author, title, journal, year.
- Optional fields: volume, number, pages, month, note.

```
@article{Xarticle,
  author      = "",
  title       = "",
  journal     = "",
  %volume    = "",
  %number   = "",
  %pages    = "",
  year       = "XXXXX",
  %month    = "",
  %note     = "",
}
```

@book

A published book

- Required fields: author/editor, title, publisher, year.
- Optional fields: volume/number, series, address, edition, month, note.

```

@book{Xbook,
  author      = "",
  title       = "",
  publisher   = "",
  %volume     = "",
  %number    = "",
  %series    = "",
  %address   = "",
  %edition   = "",
  year       = "XXXX",
  %month    = "",
  %note     = "",
}

```

@booklet

A bound work without a named publisher or sponsor.

- Required fields: title.
- Optional fields: author, howpublished, address, month, year, note.

```

@booklet{Xbooklet,

```

```
%author = "",
title = "",
%howpublished = "",
%address = "",
year = "XXXX",
%month = "",
%note = "",

}
```

@conference

Equal to inproceedings

- Required fields: author, title, booktitle, year.
- Optional fields: editor, volume/number, series, pages, address, month, organization, publisher, note.

```
@conference{Xconference,
  author = "",
  title = "",
  booktitle = "",
  %editor = "",
```



```
%volume = "",
%number = "",
%series = "",
%pages = "",
%address = "",
year = "XXXX",
%month = "",
%publisher= "",
%note = "",
}
```

@inbook

A section of a book *without* its own title.

- Required fields: author/editor, title, chapter and/or pages, publisher, year.
- Optional fields: volume/number, series, type, address, edition, month, note.

@incollection

A section of a book having its own title.

- Required fields: author, title, booktitle, publisher, year.

- Optional fields: editor, volume/number, series, type, chapter, pages, address, edition, month, note.

@inproceedings

An article in a conference proceedings.

- Required fields: author, title, booktitle, year.
- Optional fields: editor, volume/number, series, pages, address, month, organization, publisher, note.

@manual

Technical manual

- Required fields: title.
- Optional fields: author, organization, address, edition, month, year, note.

@masterthesis

Master's thesis

- Required fields: author, title, school, year.
- Optional fields: type (eg. "diploma thesis"), address, month, note.

```
@masterthesis{Xthesis,
  author      = "",
  title       = "",
  school      = "",
  %type       = "diploma thesis",
  %address    = "",
  year        = "XXXX",
  %month      = "",
  %note       = "",
}
```

@misc

Template useful for other kinds of publication

- Required fields: none
- Optional fields: author, title, howpublished, month, year, note.

```
@misc{Xmisc,
```

```
%author = "",
%title = "",
%howpublished = "",
%year = "XXXX",
%month = "",
%note = "",
}
```

@phdthesis

Ph.D. thesis

- Required fields: author, title, year, school.
- Optional fields: address, month, keywords, note.

@proceedings

The proceedings of a conference.

- Required fields: title, year.
- Optional fields: editor, volume/number, series, address, month, organization, publisher, note.

@techreport

Technical report from educational, commercial or standardization institution.

- Required fields: author, title, institution, year.
- Optional fields: type, number, address, month, note.

```
@techreport {Xtreport,  
  author = "",  
  title = "",  
  institution = "",  
  %type = "",  
  %number = "",  
  %address = "",  
  year = "XXXX",  
  %month = "",  
  %note = "",  
}
```

@unpublished

An unpublished article, book, thesis, etc.

- Required fields: author, title, note.
- Optional fields: month, year.

5.3.3. Not standard templates

@patent

(see Cite Patents with Bibtex¹³ for more details)

@collection

@electronic

13 <http://www.see-out.com/sandramau/bibpat.html>

5.3.4. Preserving capital letters

In the event that BibTeX has been set by the chosen style to not preserve all capitalization within titles, problems can occur, especially if you are referring to proper nouns, or acronyms. To tell BibTeX to keep them, use the good old curly braces around the letter in question, (or letters, if it's an acronym) and all will be well!

```
title = "The {LaTeX} Companion",
```

However, **avoid** putting the whole title in curly braces, as it will look odd if different capitalization format is used:

```
title = "{The LaTeX Companion}",
```

For convenience though, many people simply put double curly braces, which may help when writing scientific articles for different magazines, conferences with different BibTeX styles that do sometimes keep and sometimes not keep the capital letters:

```
title = {{The LaTeX Companion}},
```

As an alternative, try other BibTeX styles or modify the existing.

5.3.5. A few additional examples

Below you will find a few additional examples of bibliography entries. The first one covers the case of multiple authors in the Surname, Firstname format, and the second one deals with the incollection case.

```
@article{AbedonHymanThomas2003,  
  author = "Abedon, S. T. and Hyman, P. and Thomas, C.",  
  year = "2003",  
  title = "Experimental examination of bacteriophage latent-period  
evolution as a response to bacterial availability",  
  journal = "Applied and Environmental Microbiology",  
  volume = "69",  
  pages = "7499--7506"  
}
```



```

@incollection{Abedon1994,
  author = "Abedon, S. T.",
  title = "Lysis and the interaction between free phages and infected
  cells",
  pages = "397--405",
  booktitle = "Molecular biology of bacteriophage T4",
  editor = "Karam, Jim D. Karam and Drake, John W. and Kreuzer,
  Kenneth N. and Mosig, Gisela
  and Hall, Dwight and Eiserling, Frederick A. and Black,
  Lindsay W. and Kutter, Elizabeth
  and Carlson, Karin and Miller, Eric S. and Spicer,
  Eleanor",
  publisher = "ASM Press, Washington DC",
  year = "1994"
}

```

If you have to cite a website you can use @misc, for example:

```

@misc{website:fermentas-lambda,
  author = "Fermentas Inc.",
  title = "Phage Lambda: description \& restriction map",
  month = "November",
  year = 2008,
  url =

```

```
"http://www.fermentas.com/techinfo/nucleicacids/maplambda.htm"  
}
```

The note field comes in handy if you need to add unstructured information, for example that the corresponding issue of the journal has yet to appear:

```
@article{blackholes,  
  author="Rabbert Klein",  
  title="Black Holes and Their Relation to Hiding Eggs",  
  journal="Theoretical Easter Physics",  
  publisher="Eggs Ltd.",  
  year="2010",  
  note="(to appear)"  
}
```

5.3.6. Getting current LaTeX document to use your .bib file

At the end of your LaTeX file (that is, after the content, but before `\end{document}`), you need to place the following commands:

```
\bibliographystyle{plain}
\bibliography{sample1,sample2,...,samplen}
% Note the lack of whitespace between the commas and the next bib
file.
```

Bibliography styles are files recognized by BibTeX that tell it how to format the information stored in the `.bib` file when processed for output. And so the first command listed above is declaring which style file to use. The style file in this instance is `plain.bst` (which comes as standard with BibTeX). You do not need to add the `.bst` extension when using this command, as it is assumed. Despite its name, the plain style does a pretty good job (look at the output of this tutorial to see what I mean).

The second command is the one that actually specifies the `.bib` file you wish to use. The ones I created for this tutorial were called `sample1.bib`, `sample2.bib`, `...`, `samplen.bib`, but once again, you don't include the file extension. At the moment, the `.bib` file is in the same directory as the LaTeX document too. However, if your `.bib` file was elsewhere (which makes sense if you intend to maintain a centralized database of ref-

erences for all your research), you need to specify the path as well, e.g `\bibliography{some/where/sample}`.

Now that LaTeX and BibTeX know where to look for the appropriate files, actually citing the references is fairly trivial. The `\cite{ref_key}` is the command you need, making sure that the *ref_key* corresponds exactly to one of the entries in the .bib file. If you wish to cite more than one reference at the same time, do the following: `\cite{ref_key1, ref_key2, ..., ref_keyN}`.

5.3.7. Why won't LaTeX generate any output?

The addition of BibTeX adds extra complexity for the processing of the source to the desired output. This is largely hidden to the user, but because of all the complexity of the referencing of citations from your source LaTeX file to the database entries in another file, you actually need multiple passes to accomplish the task. This means you have to run LaTeX a number of times.

Each pass will perform a particular task until it has managed to resolve all the citation references. Here's what you need to type (into command line):

1. `latex latex_source_code.tex`
2. `bibtex latex_source_code.aux`
3. `latex latex_source_code.tex`
4. `latex latex_source_code.tex`

(Extensions are optional, if you put them note that the `bibtex` command takes the AUX file as input.)

After the first LaTeX run, you will see errors such as:

```
LaTeX Warning: Citation 'lampport94' on page 1 undefined on input line
21.
...
LaTeX Warning: There were undefined references.
```

The next step is to run `bibtex` on that same LaTeX source (or more precisely the corresponding AUX file, however not on the actual `.bib` file) to then define all the references within that document. You should see output like the following:

```
This is BibTeX, Version 0.99c (Web2C 7.3.1)
The top-level auxiliary file: latex_source_code.aux
The style file: plain.bst
Database file #1: sample.bib
```

The third step, which is invoking LaTeX for the second time will see more errors like "LaTeX Warning: Label(s) may have changed. Rerun to get cross-references right.". Don't be alarmed, it's almost complete. As you can guess, all you have to do is follow its instructions, and run LaTeX for the third time, and the document will be output as expected, without further problems.

If you want a pdf output instead of a dvi output you can use `pdflatex` instead of `latex` as follows:

1. `pdflatex latex_source_code.tex`
2. `bibtex latex_source_code.aux`
3. `pdflatex latex_source_code.tex`
4. `pdflatex latex_source_code.tex`

(Extensions are optional, if you put them note that the `bibtex` command takes the AUX file as input.)

Note that if you are editing your source in vim and attempt to use command mode and the current file shortcut (`%`) to process the document like this:

1. `:! pdflatex %`
2. `:! bibtex %`

You will get an error similar to this:

1. I couldn't open file name 'current_
file.tex.aux'

It appears that the file extension is included by default when the current file command (`%`) is executed. To process your document from within vim, you

must explicitly name the file without the file extension for `bibtex` to work, as is shown below:

```
1. :! pdflatex %
2. :! bibtex %:r (without file extension, it looks for the AUX file
   as mentioned above)
3. :! pdflatex %
4. :! pdflatex %
```

However, it is much easier to install the Vim-LaTeX plugin from here¹⁴. This allows you to simply type `\ll` when not in insert mode, and all the appropriate commands are automatically executed to compile the document. Vim-LaTeX even detects how many times it has to run `pdflatex`, and whether or not it has to run `bibtex`. This is just one of the many nice features of Vim-LaTeX, you can read the excellent Beginner's Tutorial¹⁵ for more about the many clever shortcuts Vim-LaTeX provides.

14 <http://vim-latex.sourceforge.net/>

15 <http://vim-latex.sourceforge.net/documentation/latex-suite-quickstart/>

Another option exists if you are running Unix/Linux or any other platform where you have `make`¹⁶. Then you can simply create a Makefile and use vim's `make` command or use `make` in shell. The Makefile would then look like this:

```
latex_source_code.pdf: latex_source_code.tex latex_source_-\n    code.bib\n    pdflatex latex_source_code.tex\n    bibtex latex_source_-\n    pdflatex latex_source_code.tex\n    pdflatex latex_source_code.tex
```

5.3.8. Bibliography styles

Below you can see three styles available with LaTeX:

¹⁶ http://en.wikipedia.org/wiki/Make_%28software%29

Instead of WYSIWYG editors, typesetting systems like `TeX` [1] or `LATEX` [2] can be used.

References

- [1] Paul W. Abrahams, Kathryn A. Hargreaves, and Karl Berry. *TeX for the Impatient*. 2003.
- [2] Leslie Lamport. *L^AT_EX: A Document Preparation System*. Addison-Wesley, second Edition, 1994.

Figure 2 plain

Instead of WYSIWYG editors, typesetting systems like \TeX [1] or \LaTeX [2] can be used.

References

- [1] P. W. Abrahams, K. A. Hargreaves, and K. Berry. *\TeX for the Impatient*. 2003.
- [2] L. Lamport. *\LaTeX : A Document Preparation System*. Addison-Wesley, second Edition, 1994.

Figure 3 `abbr`

Instead of WYSIWYG editors, typesetting systems like `TeX` [AHB03] or `LATEX` [Lam94] can be used.

References

- [AHB03] Paul W. Abrahams, Kathryn A. Hargreaves, and Karl Berry. *TeX for the Impatient*. 2003.
- [Lam94] Leslie Lamport. *L^AT_EX: A Document Preparation System*. Addison-Wesley, second Edition, 1994.

Figure 4 alpha

To number the references in order of appearance, rather than alphabetical order use `ieeetr`

```
\bibliographystyle{ieeetr}
```

Style Name	Author Name Format	Reference Format	Sorting
plain	Homer Jay Simpson	#ID#	by author
unsrt	Homer Jay Simpson	#ID#	as referenced
abbrv	H. J. Simpson	#ID#	by author
alpha	Homer Jay Simpson	Sim95	by author
abstract	Homer Jay Simpson	Simpson-1995a	
acn	Simpson, H. J.	#ID#	
authordate1	Simpson, Homer Jay	Simpson, 1995	
apa	Simpson, H. J. (1995)	Simpson1995	

Style Name	Author Name Format	Reference Format	Sorting
named	Homer Jay Simpson	Simpson 1995	

Web page <http://www.cs.stir.ac.uk/~kjt/software/latex/showbst.html>¹⁷ contains more examples. Examples given in PDF format <http://amath.colorado.edu/documentation/LaTeX/reference/faq/bibstyles.pdf>. More info can also be found on <http://amath.colorado.edu/documentation/LaTeX>

5.3.9. Including URLs in bibliography

As you can see, there is no field for URLs. One possibility is to include Internet addresses in howpublished field of @misc or note field of @techreport, @article, @book:

¹⁷ <http://www.cs.stir.ac.uk/~kjt/software/latex/showbst.html>

```
howpublished = "\url{http://www.example.com}"
```

Note the usage of `\url` command to ensure proper appearance of URLs¹⁸.

Another way is to use special field `url` and make bibliography style recognise it.

```
url = "http://www.example.com"
```

You need to use `\usepackage{url}` in the first case or `\usepackage{hyperref}` in the second case.

Styles provided by Natbib (see below) handle this field, other styles can be modified using `urlbst`¹⁹ program. Modifications of three standard styles (`plain`, `abbrv` and `alpha`) are provided with `urlbst`.

18 <http://en.wikibooks.org/wiki/LaTeX%2FFormatting%20Typesetting%20URLs>

19 <http://purl.org/nxg/dist/urlbst>

If you need more help about URLs in bibliography, visit FAQ of UK List of TeX²⁰.

5.3.10. Customizing bibliography appearance

One of the main advantages of BibTeX, especially for people who write many research papers, is the ability to customize your bibliography to suit the requirements of a given publication. You will notice how different publications tend to have their own style of formatting references, to which authors must adhere if they want their manuscripts published. In fact, established journals and conference organizers often will have created their own bibliography style (.bst file) for those users of BibTeX, to do all the hard work for you.

It can achieve this because of the nature of the .bib database, where all the information about your references is stored in a structured format, but nothing

20 <http://www.tex.ac.uk/cgi-bin/texfaq2html?label=citeURL>

about style. This is a common theme in LaTeX in general, where it tries as much as possible to keep content and presentation separate.

A bibliography style file (`.bst`) will tell LaTeX how to format each attribute, what order to put them in, what punctuation to use in between particular attributes etc. Unfortunately, creating such a style by hand is not a trivial task. Which is why `Makebst` (also known as *custom-bib*) is the tool we need.

`Makebst` can be used to automatically generate a `.bst` file based on your needs. It is very simple, and actually asks you a series of questions about your preferences. Once complete, it will then output the appropriate style file for you to use.

It should be installed with the LaTeX distribution (otherwise, you can download it²¹) and it's very simple to initiate. At the command line, type:

21 <http://www.mps.mpg.de/software/latex/localtex/localtex.html#makebst>

LaTeX will find the relevant file and the questioning process will begin. You will have to answer quite a few (although, note that the default answers are pretty sensible), which means it would be impractical to go through an example in this tutorial. However, it is fairly straight-forward. And if you require further guidance, then there is a comprehensive manual²² available. I'd recommend experimenting with it and seeing what the results are when applied to a LaTeX document.

If you are using a custom built .bst file, it is important that LaTeX can find it! So, make sure it's in the same directory as the LaTeX source file, *unless* you are using one of the standard style files (such as *plain* or *plainnat*, that come bundled with LaTeX - these will be automatically found in the directories that they are installed. Also, make sure the name of the .bst file you want to use

22 <http://www.mps.mpg.de/software/latex/localtex/doc/merlin.pdf>

is reflected in the `\bibliographystyle{style}` command (but don't include the `.bst` extension!).

5.3.11. Localizing bibliography appearance

When writing documents in languages other than English, you may find it desirable to adapt the appearance of your bibliography to the document language. This concerns words such as *editors*, *and*, or *in* as well as a proper typographic layout. The `babelbib` package²³ can be used here. For example, to layout the bibliography in German, add the following to the header:

```
\usepackage[fixlanguage]{babelbib}  
\selectbiblanguage{german}
```

²³ <http://tug.ctan.org/tex-archive/biblio/bibtex/contrib/babelbib/>

Alternatively, you can layout each bibliography entry according to the language of the cited document:

```
\usepackage{babelbib}
```

The language of an entry is specified as an additional field in the BibTeX entry:

```
@article{muel08,  
% ...  
language = {german}  
}
```

For babelbib to take effect, a bibliography style supported by it - one of babplain, babplai3, babalpha, babunsrt, bababbrv, bababbrv, and bababbr3 - must be used:

```
\bibliographystyle{babplain}  
\bibliography{sample}
```

5.3.12. Showing unused items

Usually Latex only display the entries which are referred to with

```
\cite
```

. It's possible to make uncited entries visible:

```
\nocite{Name89} % Show Bibliography entry of Name89  
\nocite{*} % Show all Bib-entries
```

5.3.13. Getting Bibliographic data

Many online databases provide bibliographic data in BibTeX-Format, making it easy to build your own database. For example, Google Scholar²⁴ offers

24 <http://scholar.google.com>

the option to return properly formatted output, but you must turn it on in the Preferences²⁵.

One should be alert to the fact that bibliographic databases are frequently the product of several generations of automatic processing, and so the resulting BibTex code is prone to a variety of minor errors, especially in older entries.

5.3.14. Helpful Tools

UNKNOWN TEMPLATE See also


w:en:Comparison of reference management software

²⁵ http://scholar.google.de/scholar_preferences?hl=en&lr=&output=search

Wie schreibe ich eine Seminar- oder Examensarbeit?

Daten ändern

Wolter Krämer



WIE SCHREIBE ICH EINE SEMINAR- ODER EXAMENSARBEIT?

DIN 1505

DIN 1505.2

Krämer, Wolter: Wie schreibe ich eine Seminar- oder Examensarbeit?. Frankfurt/Main: Campus Verlag, 1999. -ISBN : S. 1-260

Autor [; Autor ; ...] : Titel : Untertitel. Auflage. Ort: Verlag. Datum. -ISBN. S.

Standard 1

Standard 2

MLA

BibTex

EndNote

Figure 5 Literatur-Generator

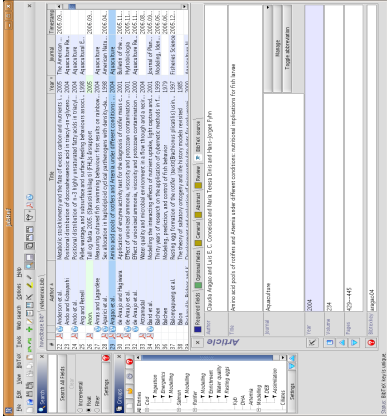


Figure 6 JabRef

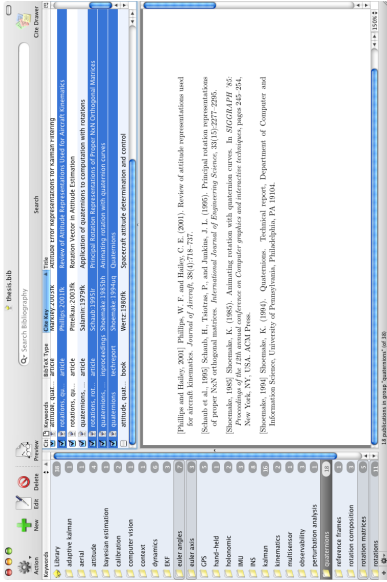


Figure 7 BibDesk

- Literatur-Generator²⁶ FAST and EASY to create your own Bibliographie (Bibtex, Endnote, Din 1505, ...). The Literatur-Generator is totally free and you do not need a registration to use the Literatur-Generator.
- Mendeley²⁷ Mendeley is cost-free academic software for managing PDFs which can manage a bibliography in Open Office and read BibTeX.
- Zotero²⁸ Zotero is a free and open reference manager working as a Firefox plugin capable of importing and exporting bib files.
- JabRef²⁹ is a Java program (under the GPL license) which lets you search many bibliographic databases such as Medline, Citeseer, IEEEXplore and arXiv and feed and manage your BibTeX local databases with your selected articles. Based on BiBTeX, JabRef can export in many other output formats such as html, MS Word or EndNote. It can be used online without being installed (<http://jabref.sourceforge.net/jws/jabref.jnlp>)

26 <http://literatur-generator.de/>

27 <http://mendeley.com>

28 <http://www.zotero.org/>

29 <http://jabref.sourceforge.net/>

- Referencer³⁰ Referencer is a Gnome application to organise documents or references, and ultimately generate a BibTeX bibliography file.
- Citavi³¹ Commercial software (with size-limited free demo version) which even searches libraries for citations and keeps all your knowledge in a database. Export of the database to all kinds of formats is possible. Works together with MS Word and Open Office Writer. Moreover plug ins for browsers and Acrobat Reader exist to automatically include references to your project.
- bibliographer³²(broken link) Bibliographer is a BibTeX bibliography database editor which aims to be easy to use. Its features include linking files to your records with indexing and searching support. The interface is designed for the easy navigation of your bibliography, and double clicking a record will open the linked file.

30 <http://icculus.org/referencer/index.html>

31 <http://www.citavi.ch>

32 <http://bibliographer.homelinux.net/>

- **cb2Bib**³³ The **cb2Bib** is a tool for rapidly extracting unformatted, or unstandardized bibliographic references from email alerts, journal Web pages, and PDF files.
- **KBibTeX**³⁴ **KBibTeX** is a **BibTeX** editor for **KDE** to edit bibliographies used with **LaTeX**. Features include comfortable input masks, starting web queries (e. g. **Google** or **PubMed**) and exporting to **PDF**, **PostScript**, **RTF** and **XML/HTML**. As **KBibTeX** is using **KDE**'s **KParts** technology, **KBibTeX** can be embedded into **Kile** or **Konqueror**.
- **KBib**³⁵ Another **BibTeX** editor for **KDE**. It has similar capabilities, and slightly different UI. Features include **BibTeX** reference generation from **PDF** files, plain text, **DOI**, **arXiv** & **PubMed** IDs. Web queries to **Google Scholar**, **PubMer**, **arXiv** and a number of other services are also supported.

33 <http://www.molspaces.com/cb2bib/>

34 <http://www.unix-ag.uni-kl.de/~fischer/kbibtex/>

35 <http://users.tpg.com.au/thachly/kbib/>

- Bibwiki³⁶ Bibwiki is a Specialpage for MediaWiki to manage BibTeX bibliographies. It offers a straightforward way to import and export bibliographic records.
- BibDesk³⁷ BibDesk is a bibliographic reference manager for Mac OS X. It features a very usable user interface and provides a number of features like smart folders based on keywords and live tex display.
- CiteULike³⁸ CiteULike is a free online service to organise academic papers. It can export citations in BibTeX format, and can "scrape" BibTeX data from many popular websites.
- Bibtex³⁹ Bibtex is a DokuWiki plugin that allows for the inclusion of bibtex formatted citations in DokuWiki pages and displays them in APA format. Note: This Plugins is vulnerable to an XSS attack -> <http://www.dokuwiki.org/plugin:bibtex>

36 <http://www.mediawiki.org/wiki/Extension:Bibwiki>

37 <http://bibdesk.sourceforge.net/>

38 <http://www.citeulike.org/>

39 <http://stat.genopole.cnrs.fr/~cambroise/doku.php?id=softwares:dokuwikibibtexplugin>

- BibSonomy⁴⁰ — A free social bookmark and publication management system based on BibTeX.
- Synapsen⁴¹ — Hypertextual Card Index / Reference Manager with special support for BiBTeX / biblatex, written in Java.

5.3.15. Summary

Although it can take a little time to get to grips with BibTeX, in the long term, it's an efficient way to handle your references. It's not uncommon to find .bib files on websites that people compile as a list of their own publications, or a survey of relevant works within a given topic, etc. Or in those huge, online bibliography databases, you often find BibTeX versions of publications, so it's a quick cut-and-paste into your own .bib file, and then no more hassle!

40 <http://www.bibsonomy.org/>

41 <http://www.verzetteln.de/synapsen/>

Having all your references in one place can be a big advantage. And having them in a structured form, that allows customizable output is another one. There are a variety of free utilities that can load your .bib files, and allow you to view them in a more efficient manner, as well as sort them and check for errors.

5.4. Bibliography in the table of contents

If you are writing a *book* or *report*, you'll likely insert your bibliography using something like:

```
\begin{thebibliography}{99}
\bibitem{bib:one_book} some information
\bibitem{bib:one_article} other information
\end{thebibliography}
```

Or, if you are using BibTeX, your references will be saved in a .bib file, and your TeX document will include the bibliography by these commands:

```
\bibliographystyle{plain}  
\bibliography{mybibtxfile}
```

Both of these examples will create a chapter-like (or section-like) output showing all your references. But even though the resulting “References” looks like a chapter or section, it will not be handled quite the same: it will not appear in the Table of Contents.

5.4.1. Using **tocbibind**

The most comfortable way of adding your bibliography to the table of contents is to use the dedicated package **tocbibind** that works with many standard document classes. Simply include this code in the preamble of your document:

```
\usepackage[nottoc,numbib]{tocbibind}
```


This will include the Bibliography in the Table of Contents with proper numbering. The **tocbibind** package can also handle including the List of Figures, List of Tables and the Table of Contents itself in the Table of Contents. It has many options for numbering, document structure etc. to fit almost any scenario. See the **tocbibind** CTAN page⁴² for detailed documentation.

5.4.2. Other methods

As unnumbered item

If you want your bibliography to be in the table of contents, just add the following two lines just before the *thebibliography* environment:

```
\clearpage
\addcontentsline{toc}{chapter}{Bibliography}
```

42 <http://www.ctan.org/tex-archive/macros/latex/contrib/tocbibind>

(OR

```
\addcontentsline{toc}{section}{Bibliography}
```

if you're writing an *article*)

The first line just terminates the current paragraph and page. If you are writing a *book*, use `\cleardoublepage` to match the style used. The second line will add a line in the Table of Contents (first option, *toc*), it will be like the ones created by chapters (second option, *chapter*), and the third argument will be printed on the corresponding line in the Table of Contents; here *Bibliography* was chosen because it's the same text the *thebibliography* environment will automatically write when you use it, but you are free to write whatever you like. If you are using separate bib file, add these lines between `\bibliographystyle` and `\bibliography`.

If you use `hyperref`⁴³ package, you should also use `\phantomsection` command to enable hyperlinking from the table of contents to bibliography.

```
\cleardoublepage
\phantomsection
\addcontentsline{toc}{chapter}{Bibliography}
```

This trick is particularly useful when you have to insert the bibliography in the Table of Contents, but it can work for anything. When LaTeX finds the code above, it will record the info as described and the current page number, inserting a new line in the Contents page.

As numbered item

If you instead want bibliography to be numbered section or chapter, you'll likely use this way:

```
\cleardoublepage % This is needed if the book class is used, to place
the anchor in the correct page,
% because the bibliography will start on its own
page.
```

```
    % Use \clearpage instead if the document class uses
the "oneside" argument
\renewcommand*{\refname}{} % This will define heading of bibliography
to be empty, so you can...
\section{Bibliography} % ...place a normal section heading before
the bibliography entries.

\begin{thebibliography}{99}
...
\end{thebibliography}
```

Another even easier solution is to use

```
\section
inside of the
\renewcommand
block:

\renewcommand{\refname}{\section{Sources}} % Using "Sources" as the
title of the section
\begin{thebibliography}{99}
```

```
...
\end{thebibliography}
```

You may wish to use

```
\renewcommand*{\refname}{\vspace*{-1em}}
```

followed by

```
\vspace*{-1em}
```

to counteract the extra space the blank

```
\refname
```

inserts.

- **Note: Use**

```
\bibname
```

instead of

```
\refname
```

if you use the book or report class.

5.5. Multiple Bibliographies

5.5.1. Using `multibib`

This package is for multiple Bibliographies for different sections in your work. For example, you can generate a bibliography for each chapter. You can find information about the package on CTAN⁴⁴

5.5.2. Using `bibtopic`

The `bibtopic`-Package⁴⁵ is created to differ the citations on more files, so that you can divide the bibliography into more parts.

```
\documentclass{article}
```

44 <http://ctan.org/pkg/multibib>

45 <http://ctan.org/pkg/bibtopic>

```
\usepackage{bibtopic}
\begin{document}

\bibliographystyle{alpha}
\section{Testing}
Let's cite all the books: \cite{ColBenh:93} and
\cite{Munt:93}; and an article: \cite{RouxSmart:95}.
```

File books.bib is use for this listing:

```
\begin{btSect}{books}
\section{References from books}
\btPrintCited
\end{btSect}
```

Here, the articles.bib is used, and the listing is in plain-format instead pf the standard alpha.

```
\begin{btSect}[plain]{articles}
\section{References from articles}
\btPrintCited
\section{Articles not cited}
\btPrintNotCited
\end{btSect}
```

Just print alll entries here with \btPrintAll

```
\begin{btSect}[plain]{internet}
\section{References from the internet}
\btPrintAll
```

```
\end{btSect}  
\end{document}
```

5.6. References

fr:LaTeX/Gestion de la bibliographie⁴⁶

⁴⁶ <http://fr.wikibooks.org/wiki/LaTeX%2FGestion%20de%20la%20bibliographie>

Part II.

Formating

6. List Structures

Convenient and predictable list formatting is one of the many advantages of using LaTeX. Users of WYSIWYG word processors can sometimes be frustrated by the software's attempts to determine when they intend lists to begin and end. As a mark-up language, LaTeX gives more control over the structure and content of lists.

6.1. List Structures

Lists often appear in documents, especially academic, as their purpose is often to present information in a clear and concise fashion. List structures in LaTeX are simply environments which essentially come in three flavors:

`itemize`

,

`enumerate`

and

`description`

.

All lists follow the basic format:

```
\begin{list_type}  
  \item The first item  
  \item The second item
```

```
\item The third etc \ldots
\end{list_type}
```

All three of these types of lists can have multiple paragraphs per item: just type the additional paragraphs in the normal way, with a blank line between each. So long as they are still contained within the enclosing environment, they will automatically be indented to follow underneath their item.

6.1.1. Itemize

This environment is for your standard bulleted list of items.

```
\begin{itemize}
  \item The first item
  \item The second item
  \item The third etc \ldots
\end{itemize}
```

- The first item
- The second item
- The third etc ...

Figure 8

6.1.2. Enumerate

The enumerate environment is for ordered lists, where by default, each item is numbered sequentially.

1. The first item
2. The second item
3. The third etc ...

```
\begin{enumerate}  
  \item The first item  
  \item The second item  
  \item The third etc \ldots  
\end{enumerate}
```

Figure 9

6.1.3. Description

The description environment is slightly different. You can specify the item label by passing it as an optional argument (although optional, it would look odd if you didn't include it!). Ideal for a series of definitions, such as a glossary.

First The first item

```
\begin{description}  
  \item[First] The first item  
  \item[Second] The second item  
  \item[Third] The third etc \ldots  
\end{description}
```

Second The second item

Third The third etc ...

Figure 10

Sometimes you want a description where the text begins on a new line. This cannot easily be done with

```
\\
```

. The trick is to use

```
\hfill
```

.

```
\begin{description}
\item[First] \hfill \\\
The first item
\item[Second] \hfill \\\
The second item
\item[Third] \hfill \\\
The third etc \ldots
\end{description}
```

First

The first item

Second

The second item

Third

The third etc ...

Figure 11

6.1.4. Compacted lists

As you may have noticed, in standard LaTeX document classes, the vertical spacing between items, and above and below the lists as a whole, is more

than between paragraphs: it may look odd if the descriptions are too short. If you want tightly-packed lists, use the `mdwlist` package (included in the `mdwtols` bundle), which provides compact, "starred" versions of the previous environments, i.e.

```
itemize*
```

```
,
```

```
enumerate*
```

```
and
```

```
description*
```

. They work exactly in the same way, but the output is more compact. Other packages providing compacted lists are `paralist` and `enumitem`.

Alternatively, use the `memoir` class and with `\tightlists`.

6.1.5. Nested Lists

Latex will happily allow you to insert a list environment into an existing one (up to a depth of four -- if you need more than four, use the `easylist` package). Simply begin the appropriate environment at the desired point within the current list. Latex will sort out the layout and any numbering for you.

1. The first item
 - (a) Nested item 1
 - (b) Nested item 2
2. The second item
3. The third etc ...

```
\begin{enumerate}  
  \item The first item  
  \begin{enumerate}  
    \item Nested item 1  
    \item Nested item 2  
  \end{enumerate}  
  \item The second item  
  \item The third etc \ldots  
\end{enumerate}
```

Figure 12

6.1.6. Customizing Lists

Customizing LaTeX is outside the beginners' domain. While not necessarily difficult in itself, because beginners are already overwhelmed with the array of commands and environments, moving on to more advanced topics runs the risk of confusion.

However, since the tutorial is on formatting, I shall still include a brief guide on customizing lists. Feel free to skip!

Customizing Line Spacing in Lists

Inside lists you can redefine some length/dimension variables of LaTeX, for example using:

```
\begin{itemize} \itemsep1pt \parskip0pt \parsep0pt
\item first item
```

```
\item second item  
\end{itemize}
```

Alternatively, to create a unified look in your document you can define your own enumerate environment:

```
\newenvironment{my_enumerate}  
{\begin{enumerate}  
  \setlength{\itemsep}{1pt}  
  \setlength{\parskip}{0pt}  
  \setlength{\parsep}{0pt}}  
\end{enumerate}}
```

Customizing Enumerated Lists

The thing people want to change most often with Enumerated lists are the counters. A quick solution to this problem is provided by the `enumerate`

package of David Carlisle¹, or the more sophisticated package `enumitem` by Javier Bezos².

To go any further and do it yourself instead, a brief introduction to LaTeX *counters* is required. For anything that LaTeX automatically numbers, such as section headers, figures, and itemized lists, there is a counter associated with it that controls the numbering.

There are four individual counters that are associated with itemized lists, each one represents the four possible levels of nesting, which are called:

`enumi`

,

`enumii`

,

¹ <http://mirrors.fe.up.pt/pub/CTAN/macros/latex/required/tools/enumerate.pdf>The enumerate package, David Carlisle 1999

² <http://mirrors.fe.up.pt/pub/CTAN/macros/latex/contrib/enumitem/enumitem.pdf>The enumitem package, Javier Bezos 2011

enumiii

,

enumiv

. In order to reset any of these counters in the middle of an enumeration simply **use**

```
\setcounter
```

. The counter is incremented by

```
\item
```

before it is printed. For example to reset

```
enumi
```

use:

```
\begin{enumerate}  
  \setcounter{enumi}{4}  
  \item fifth element  
\end{enumerate}
```

which prints as:

```
5. fifth element
```

Each counter also has a default format that dictates how it is displayed whenever LaTeX needs to print it. Such formats are specified using internal LaTeX commands:

Command	Example
<code>\arabic</code>	1, 2, 3 ...
<code>\alph</code>	a, b, c ...
<code>\Alph</code>	A, B, C ...
<code>\roman</code>	i, ii, iii ...
<code>\Roman</code>	I, II, III ...
<code>\fnsymbol</code>	Aimed at footnotes (see below), but prints a sequence of symbols.

Each counter entity holds various bits of information about itself. To get to the numbered element, simply use

```
\the
```

followed immediately (i.e., no space) by the name of the counter, e.g.,

```
\theenumi
```

. This is often referred to as the *representation* of a counter.

Now, that's most of the technicalities out of the way. To make changes to the formatting of a given level:

```
\renewcommand{\representation}{\format_command{counter}}
```

Admittedly, the generic version is not that clear, so a couple of examples will clarify:

```
%Redefine the first level
\renewcommand{\theenumi}{\Roman{enumi}}
\renewcommand{\labelenumi}{\theenumi}
```

```
%Redefine the second level
\renewcommand{\theenumii}{\Alph{enumii}}
\renewcommand{\labelenumii}{\theenumii}
```

The method used above first explicitly changes the format used by the counter. However, the element that controls the label needs to be updated to reflect the change, which is what the second line does. Another way to achieve this result is this:

```
\renewcommand{\labelenumi}{\Roman{enumi}}
```

This simply redefines the appearance of the label, which is fine, providing that you do not intend to cross-reference to a specific item within the list, in which case the reference will be printed in the previous format. This issue does not arise in the first example.

Note that you can also add other symbols, such as parentheses and periods, before and after the counter. For instance, to create a list indexed by lower

case letters with parentheses before and after the letter, you might enter the following:

```
\renewcommand{\labelenumi}{(\alph{enumi})}
```

Customizing Itemised Lists

Itemized lists are not as complex as they do not need to count. Therefore, to customize, you simply change the labels. It can be done manually for each entry with

```
\item[new symbol]
```

, eg

```
\item[ $\star$ ]
```

.

The `itemize` labels are accessed via

```
\labelitemi
```

```
,  
\labelitemi  
  
,  
\labelitemiii  
  
,  
\labelitemiv
```

, for the four respective levels.

```
\renewcommand{\labelitemi}{\textgreater}
```

The above example would set the labels for the first level to a greater than (>) symbol. Of course, the text symbols available in LaTeX are not very exciting. Why not use one of the ZapfDingbat symbols, as described in the Symbols³ section. Or use a mathematical symbol:

```
\renewcommand{\labelitemi}{$\star$}
```

Itemized list with tightly set items, that is with no vertical space between two consecutive items, can be created as follows.

```
\begin{itemize}  
  \setlength{\itemsep}{0cm}%  
  \setlength{\parskip}{0cm}%  
  \item Item opening the list  
  \item Item tightly following  
\end{itemize}
```

Details of Customizing Lists

Note that it is necessary that the

```
\renewcommand  
appears after the  
\begin{document}
```

instruction so the changes made are taken into account. This is needed for both enumerated and itemized lists.

6.1.7. Inline lists

Inline lists are a special case as they require the use of the `paralist` package which provides the

```
inparaenum
```

environment (with an optional formatting specification in square brackets):

Inline lists, which are sequential in nature, just like enumerated lists, but are *a)* formatted within their paragraph; *b)* usually labelled with letters; and *c)* usually have the final item prefixed with ‘and’ or ‘or’, like this example.

```
...
\usepackage{paralist}

\begin{document}

\textbf{\itshape Inline lists}, which are
sequential in nature, just like enumerated
lists, but are
\begin{inparaenum}[\itshape a\upshape]
\item formatted within their paragraph;
\item usually labelled with letters; and
\item usually have the final item prefixed with
‘and’ or ‘or’,
\end{inparaenum} like this example.

...
```

Figure 13

To change the styles of the counter, tokens A, a, I, i, and 1 can be used in the optional argument to produce the counter with one of the styles

,

`\alph`

,

`\Roman`

,

`\roman`

and

`\arabic`

. For example:

```
\begin{inparaenum} [ (i) ]
```

produces the labels (i), (ii), (iii) ...

Other packages providing inline lists are `shortlist` and `enumitem`.

6.2. Easylist package

The `easylist` package allows you to create list using a more convenient syntax and with infinite nested levels. It is also very customizable.

Load the package with the control character as optional argument:

```
\usepackage[ampersand]{easylist}
```

Example of custom enumerate:

```
\begin{easylist}[enumerate]  
  
\listproperties{Style2*={,Numbers=a,Numbers1=R,FinalMark=.}  
& Main item~:  
    && Sub item.  
    && Another sub item.  
\end{easylist}
```

Important note: `easylist` has some drawbacks. First if you need to put an `easylist` inside an environment using the same control character as the one specified for `easylist`, you may get an error. To circumvent it, use the following commands provided by `easylist`:

```
\Activate  
\begin{easylist}  
& . . .  
\end{easylist}  
\Deactivate
```

Besides using `easylist` along with figures may cause some trouble to the layout and the indentation. `LaTeX` lists do not have this problem.

7. Tables

In academic writing, tables are a common feature, often for summarizing results from research. It is therefore a skill that needs mastering in order to produce quality papers.

However, if there is one area about LaTeX that is the least intuitive, then this is it. Basic tables are not too taxing, but you will quickly notice that anything more advanced can take a fair bit of construction. So, we start slowly and build up from there.

Workaround: You might save lots of time by building tables using specialized software and exporting them in LaTeX format. The following plugins and libraries are available for some popular software:

- `calc2latex`¹: for OpenOffice.org Calc spreadsheets,
- `excel2latex`²: for Microsoft Office Excel,
- `matrix2latex`³: for MATLAB,
- `matrix2latex`⁴: for Python and MATLAB,
- `latex-tools`⁵: a Ruby library,
- `xtable`⁶: a library for R,

1 <http://calc2latex.sourceforge.net/>
2 <http://www.ctan.org/tex-archive/support/excel2latex/>
3 <http://www.mathworks.com/matlabcentral/fileexchange/4894-matrix2latex>
4 <https://code.google.com/p/matrix2latex/>
5 <http://rubygems.org/gems/latex-tools>
6 <http://cran.r-project.org/web/packages/xtable/index.html>

- `org-mode`⁷: for Emacs users, `org-mode` tables can be used inline in LaTeX documents, see https://www.gnu.org/software/emacs/manual/html_node/org/A-LaTeX-example.html⁸ for a tutorial.
- Emacs Align Commands⁹: while not a workaround per se, the align commands can clean up a messy LaTeX table.
- Online Table generator for L^AT_EX¹⁰: An online tool for creating simple tables within the browser. LaTeX format is directly generated as you type.

7.1. Floating with `table`

It is highly recommended to place a `tabular`-environment only in the `table`-environment, which is able to float and add a label and caption.

7 <http://orgmode.org/>

8 https://www.gnu.org/software/emacs/manual/html_node/org/A-LaTeX-example.html

9 <http://emacswiki.org/emacs/AlignCommands>

10 <http://truben.no/latex/table/>


```

\begin{table}[position specifier]
\centering
\begin{tabular}{|l|}
... your table ...
\end{tabular}
\caption{This table shows some data}
\label{tab:myfirsttable}
\end{table}

```

The captions show now up in the list of tables, if displayed.

You can set the optional parameter *position specifier* to define the position of the table, where it should be placed. The following characters are all possible placements. Using sequences of it define your "wishlist" to LaTeX.

h	here
t	top
b	bottom
p	page

Default is *tbp*, which means that it is by default placed on the top of the page. If it's not possible, its placed on the bottom and last possibility is to place it with other floating environments on an extra page. You can force LaTeX to use one given position. E.g. `[!h]` forces LaTeX to place it exactly where you place it (Except when it's really impossible, e.g you place a table *here* and this place would be the last line on a page). Centering the table horizontally works using the `\centering`-command just after opening the `table`-environment.

More informations about floating environments, captions etc. can be found in [Floats, Figures and Captions](#)¹¹.

7.2. The `tabular` environment

The `tabular` environment can be used to typeset tables with optional horizontal and vertical lines. `LaTeX` determines the width of the columns automatically.

The first line of the environment has the form:

```
\begin{tabular}[pos]{table spec}
```

The *table spec* argument tells `LaTeX` the alignment to be used in each column and the vertical lines to insert.

The number of columns does not need to be specified as it is inferred by looking at the number of arguments provided. It is also possible to add vertical lines between the columns here. The following symbols are available to describe the table columns (some of them require that the package *array* has been loaded):

c	centered column
r	right-justified column
p{width}	paragraph column with text vertically aligned at the top
m{width}	paragraph column with text vertically aligned in the middle (requires array package)
b{width}	paragraph column with text vertically aligned at the bottom (requires array package)
	vertical line
	double vertical line

By default, if the text in a column is too wide for the page, LaTeX won't automatically wrap it. Using `p{width}` you can define a special type of column which will wrap-around the text as in a normal paragraph. You can pass the width using any unit supported by LaTeX, such as pt and cm, or

command lengths, such as `\textwidth`. You can find a complete list in appendix Useful Measurement Macros¹².

The optional parameter *pos* can be used to specify the vertical position of the table relative to the baseline of the surrounding text. In most cases, you will not need this option. It becomes relevant only if your table is not in a paragraph of its own. You can use the following letters:

b	bottom
c	center (default)
t	top

In the first line you have pointed out how many columns you want, their alignment and the vertical lines to separate them. Once in the environment, you have to introduce the text you want, separating between cells and introducing new lines. The commands you have to use are the following:

<code>&</code>	column separator
<code>\\</code>	start new row (additional space may be specified after <code>\\</code> using square brackets, such as <code>\\ [6pt]</code>)
<code>\hline</code>	horizontal line
<code>\newline</code>	start a new line within a cell (in a paragraph column)
<code>\cline{i-j}</code>	partial horizontal line beginning in column <i>i</i> and ending in column <i>j</i>

Note, any white space inserted between these commands is purely down to ones' preferences. I personally add spaces between to make it easier to read.

7.2.1. Basic examples

This example shows how to create a simple table in LaTeX. It is a three-by-three table, but without any lines.

1	2	3
4	5	6
7	8	9

```
\begin{tabular}{l c r }  
1 & 2 & 3 \\  
4 & 5 & 6 \\  
7 & 8 & 9 \\  
\end{tabular}
```

Expanding upon that by including some vertical lines:

1	2	3
4	5	6
7	8	9

```

\begin{tabular}{|c|c|c|}
 1 & 2 & 3 \\
 4 & 5 & 6 \\
 7 & 8 & 9 \\
\end{tabular}

```

To add horizontal lines to the very top and bottom edges of the table:

1	2	3
4	5	6
7	8	9

```
\begin{tabular}{| l | c | | r }  
  \hline  
  1 & 2 & & 3 \\ \hline  
  4 & 5 & & 6 \\ \hline  
  7 & 8 & & 9 \\ \hline  
  \end{tabular}
```

And finally, to add lines between all rows, as well as centering (notice the use of the center environment - of course, the result of this is not obvious from the preview on this web page):

1	2	3
4	5	6
7	8	9

```
\begin{center}
\begin{tabular}{|l|c||r}
\hline
1 & 2 & 3 \\
\hline
4 & 5 & 6 \\
\hline
7 & 8 & 9 \\
\hline
\end{tabular}
\end{center}
```

```

\begin{tabular}{|r|l|}
\hline
7C0 & hexadecimal \\
3700 & octal \\
11111000000 & binary \\
\hline
1984 & decimal \\
\hline
\end{tabular}

```

7C0	hexadecimal
3700	octal
11111000000	binary
1984	decimal

Figure 14

7.2.2. Column specification using `>{\cmd}` and `<{\cmd}`

The column specification can be altered using the `array` package. This is done in the argument of the tabular environment using `>{\command}` for commands executed right *before* each column element and `<{\command}` for commands to be executed right *after* each column element. As an example: to

get a column in math mode enter: `\begin{tabular}>{\$}c<{\$}`. Another example is changing the font: `\begin{tabular}>{\small}c<` to print the column in a small font.

The argument of the `>` and `<` specifications must be correctly balanced when it comes to `{` and `}` characters. This means that `>{\bfseries}` is valid, while `>\textbf}` will not work and `>\textbf{}` is not valid. If there is the need to use the text of the table as an argument (for instance, using the `\textbf` to produce bold text), one should use the `\bgroup` and `\egroup` commands: `>\textbf\bgroup}c<\egroup}` produces the intended effect. This works only for some basic LaTeX commands. For other commands, such as `\underline` to underline text, it is necessary to temporarily store the column text in a box using `\rbox`. First, you must define such a box with `\newsavebox{\boxname}` and then you can define:

```
>{\begin{\rbox}{\boxname}}%  
l%  
<{\end{\rbox}}%
```

```
\underline{\unhbox\boxname} } %  
}
```

This stores the text in a box and afterwards, takes the text out of the box with `\unhbox` (this destroys the box, if the box is needed again one should use `\uncopy` instead) and passing it to `\underline`. (For LaTeX2e, you may want to use `\usebox{\boxname}` instead of `\unhbox\boxname`.)

This same trick done with `\raisebox` instead of `\underline` can force all lines in a table to have equal height, instead of the natural varying height that can occur when e.g. math terms or superscripts occur in the text.

Here is an example showing the use of both `p{...}` and `>\centering`

:

```
\begin{tabular}{>\centering}p{3.5cm}<\centering}p{3.5cm}  
Geometry & Algebra  
\tabularnewline  
\hline  
Points & Addition  
\tabularnewline
```

```
Spheres & Multiplication  
\end{tabular}
```

Note the use of `\tabularnewline` instead of `\\` to avoid a `Misplaced \noalign` error.

7.2.3. Text wrapping in tables

LaTeX's algorithms for formatting tables have a few shortcomings. One is that it will not automatically wrap text in cells, even if it overruns the width of the page. For columns that you know will contain a certain amount of text, then it is recommended that you use the *p* attribute and specify the desired width of the column (although it may take some trial-and-error to get the result you want). Use the *m* attribute to have the lines aligned toward the middle of the box and the *b* attribute to align along the bottom of the box.

Here is a practical example. The following code creates two tables with the same code; the only difference is that the last column of the second one has

a defined width of 5 centimeters, while in the first one we didn't specify any width. Compiling this code:

```
\documentclass{article}
\usepackage[english]{babel}
\begin{document}
```

Without specifying width for last column:

```
\begin{center}
\begin{tabular}{|l|l|l|l|l|}
\hline
Day & Min Temp & Max Temp & Summary & \hline
Monday & 11C & 22C & A clear day with lots of sunshine.
However, the strong breeze will bring down the temperatures. & \hline
Tuesday & 9C & 19C & Cloudy with rain, across many northern
regions. Clear spells
across most of Scotland and Northern Ireland,
but rain reaching the far northwest. & \hline
Wednesday & 10C & 21C & Rain will still linger for the morning.
Conditions will improve by early afternoon and continue
```

```

throughout the evening. \\
\hline
\end{tabular}
\end{center}

With width specified:
\begin{center}
\begin{tabular}{| l | l | l | p{5cm} |}
\hline
Day & Min Temp & Max Temp & Summary \\
Monday & 11C & 22C & A clear day with lots of sunshine.
However, the strong breeze will bring down the temperatures. \\
\hline
Tuesday & 9C & 19C & Cloudy with rain, across many northern
regions. Clear spells
across most of Scotland and Northern Ireland,
but rain reaching the far northwest. \\
\hline
Wednesday & 10C & 21C & Rain will still linger for the morning.
Conditions will improve by early afternoon and continue
throughout the evening. \\
\hline
\end{tabular}
\end{center}

```



```
\end{document}
```

You get the following output:

Without specifying width for last column:

Day	Min Temp	Max Temp	Summary
Monday	11C	22C	A clear day with lots of sunshine. However, the strong breeze will
Tuesday	9C	19C	Cloudy with rain, across many northern regions. Clear spells ac
Wednesday	10C	21C	Rain will still linger for the morning. Conditions will improve by

With width specified:

Day	Min Temp	Max Temp	Summary
Monday	11C	22C	A clear day with lots of sunshine. However, the strong breeze will bring down the temperatures.
Tuesday	9C	19C	Cloudy with rain, across many northern regions. Clear spells across most of Scotland and Northern Ireland, but rain reaching the far northwest.
Wednesday	10C	21C	Rain will still linger for the morning. Conditions will improve by early afternoon and continue throughout the evening.

Figure 15

Note that the first table is cropped: The output is wider than the page width.

7.2.4. Text justification in tables

On rare occasions, it might be necessary to stretch every row in a table to the natural width of its longest line, for instance when one has the same text in two languages and wishes to present these next to each other with lines syncing up. A tabular environment helps control where lines should break, but cannot justify the text, which leads to ragged right edges. The `eqparbox` package provides the command `\eqmakebox` which is like `\makebox` but instead of a *width* argument, it takes a tag. During compilation it bookkeeps which `\eqmakebox` with a certain tag contains the widest text and can stretch all `\eqmakeboxes` with the same tag to that width. Combined with the `array` package, one can define a column specifier that justifies the text in all lines: (See the documentation of the `eqparbox` package for more details.)

```
\newsavebox{\tstretchbox}
```

```

\newcolumntype{S}[1]{%
>{\begin{lrbox}{\tstretchbox}}%
1%
<{\end{lrbox}}%
\egmakebox[#1][s]{\unhcopy\tstretchbox}}%
}

```

7.2.5. Other environments inside tables

If you use some LaTeX environments inside table cells, like `verbatim` or `enumerate`¹³

```

\begin{tabular}{|c|c|}
\hline
\begin{verbatim}
code
\end{verbatim}
& description

```

¹³ [http://en.wikibooks.org/wiki/LaTeX%2FFormatting%](http://en.wikibooks.org/wiki/LaTeX%2FFormatting%23Enumerate)

```
\\ \hline
\end{tabular}
```

you might encounter errors similar to

```
! LaTeX Error: Something's wrong--perhaps a missing \item.
```

To solve this problem, change column specifier¹⁴ to "paragraph" (p, m or b).

```
\begin{tabular}{| m{5cm} | c |}
```

7.2.6. Defining multiple columns

It is possible to define many identical columns at once using the `*{num}{str}` syntax.

This is particularly useful when your table has many columns.

Here is a table with six centered columns flanked by a single column on each side:

```
\begin{tabular}{l*{6}{c}r}
Team          & P & W & D & L & F & A & Pts \\
\hline
Manchester United & 6 & 4 & 0 & 2 & 10 & 5 & 12 \\
Celtic          & 6 & 3 & 0 & 3 & 8 & 9 & 9 \\
Benfica        & 6 & 2 & 1 & 3 & 7 & 8 & 7 \\
FC Copenhagen  & 6 & 2 & 1 & 2 & 5 & 8 & 7 \\
\end{tabular}
```

Team	P	W	D	L	F	A	Pts
Manchester United	6	4	0	2	10	5	12
Celtic	6	3	0	3	8	9	9
Benfica	6	2	1	3	7	8	7
FC Copenhagen	6	2	1	2	5	8	7

Figure 16

7.2.7. @-expressions

The column separator can be specified with the `@{ . . . }` construct.

It typically takes some text as its argument, and when appended to a column, it will automatically insert that text into each cell in that column before the actual data for that cell. This command kills the inter-column space and

replaces it with whatever is between the curly braces. To add space, use `@{\hspace{width}}`.

Admittedly, this is not that clear, and so will require a few examples to clarify. Sometimes, it is desirable in scientific tables to have the numbers aligned on the decimal point. This can be achieved by doing the following:

```
3.14159
16.2
123.456
```

```
\begin{tabular}{r@{.}l}
3 & 14159 \\
16 & 2 \\
123 & 456 \\
\end{tabular}
```

Note that the headers should be enclosed in `\multicolumn{2}{1}{HEADER}`.

Alternatively, to center the column on the decimal separator the `dcolumn` package may be used, which provides a new column specifier for floating point data.

The space-suppressing qualities of the `@`-expression actually make it quite useful for manipulating the horizontal spacing between columns. Given a basic table, and varying the column descriptions:

```
\begin{tabular}{|l|l|l|}
\hline
stuff & stuff & \ \ \hline
stuff & stuff & \ \
\hline
\end{tabular}
```


{ | | | | }

stuff	stuff
stuff	stuff

Figure 17

{ | @ { } | | | @ { } | }

stuff	stuff
stuff	stuff

Figure 18

{ | @ { } | | @ { } | | }

stuff	stuff
stuff	stuff

Figure 19

```
{ | @ { } | @ { } | @ { } | }
```

stuff	stuff
stuff	stuff

Figure 20

7.3. Spanning

To complete this tutorial, we take a quick look at how to generate slightly more complex tables. Unsurprisingly, the commands necessary have to be embedded within the table data itself.

7.3.1. Rows spanning multiple columns

The command for this looks like this: `\multicolumn{num_sequent_columns}{alignment}{contents}`. *num_cols* is the number of subsequent columns to merge; *alignment* is either l, c, r, or to have text wrapping specify a width `p{5.0cm}`. And *contents* is simply the actual data you want to be contained within that cell. A simple example:

Team sheet	
GK	Paul Robinson
LB	Lucus Radebe
DC	Michael Duberry
DC	Dominic Matteo
RB	Dider Domi
MC	David Batty
MC	Eirik Bakke
MC	Jody Morris
FW	Jamie McMaster
ST	Alan Smith
ST	Mark Viduka

```

\begin{tabular}{|l|l|}
\hline
\multicolumn{2}{|c|}{Team sheet} \\
\hline
GK & Paul Robinson \\
LB & Lucus Radebe \\
DC & Michael Duberry \\
DC & Dominic Matteo \\
RB & Dider Domi \\
MC & David Batty \\
MC & Eirik Bakke \\
MC & Jody Morris \\
FW & Jamie McMaster \\
ST & Alan Smith \\
ST & Mark Viduka \\
\hline
\end{tabular}

```

Figure 21

7.3.2. Columns spanning multiple rows

The first thing you need to do is add `\usepackage{multirow}` to the preamble¹⁵. This then provides the command needed for spanning rows: `\multirow{num_rows}{width}{contents}`. The arguments are pretty simple to deduce (`*` for the *width* means the content's natural width).

¹⁵ Package `multirow` on CTAN `^{\http://www.ctan.org/tex-archive/macros/latex/contrib/multirow/}`

Team sheet	
Goalkeeper	GK Paul Robinson
Defenders	LB Lucus Radebe
	DC Michael Duberry
	DC Dominic Matteo
Midfielders	RB Didier Domi
	MC David Batty
	MC Eirik Bakke
Forward	MC Jody Morris
	FW Jamie McMaster
	ST Alan Smith
Strikers	ST Mark Viduka

```

...
\usepackage{multirow}
...
\begin{tabular}{|l|l|l|l|}
\hline
\multicolumn{3}{|c|}{Team sheet} \\
\hline
Goalkeeper & GK & Paul Robinson \\
\multirow{4}{*}{Defenders} & LB & Lucus Radebe \\
& DC & Michael Duberry \\
& DC & Dominic Matteo \\
& RB & Didier Domi \\
\multirow{3}{*}{Midfielders} & MC & David Batty \\
& MC & Eirik Bakke \\
& MC & Jody Morris \\
\multirow{3}{*}{Strikers} & FW & Jamie McMaster \\
& ST & Alan Smith \\
& ST & Mark Viduka \\
\hline
\end{tabular}

```

Figure 22

Figure 22 shows the output of the code above.

The main thing to note when using `\multirrow` is that a blank entry must be inserted for each appropriate cell in each subsequent row to be spanned.

If there is no data for a cell, just don't type anything, but you still need the "&" separating it from the next column's data. The astute reader will already have deduced that for a table of n columns, there must always be $n - 1$ ampersands in each row. The exception to this is when `\multicolumn` and `\multirrow` are used to create cells which span multiple columns or rows.

7.3.3. Spanning in both directions simultaneously

Here is a nontrivial example of how to use spanning in both directions simultaneously and have the borders of the cells drawn correctly:

Here is another example exploiting the same ideas to

The command

18 <http://www.ctan.org/tex-archive/macros/latex/contrib/ctable/>

19 **Chapter 13.2.9 on page 433**

20 <http://www.ctan.org/tex-archive/macros/latex/contrib/caption/>

21 <http://tug.ctan.org/pkg/hhline>

22 <http://tug.ctan.org/pkg/array>

23 <http://tug.ctan.org/pkg/colortbl>

24 <http://tug.ctan.org/pkg/supertabular>

25 <http://tug.ctan.org/pkg/longtable>

26 <http://ctan.org/tex-archive/macros/latex/contrib/threeparttable>

27 <http://tug.ctan.org/pkg/xtab>

28 <http://tug.ctan.org/pkg/tabulary>

29 <http://tug.ctan.org/pkg/arydshln>

30 <http://tug.ctan.org/pkg/ctable>

31 <http://tug.ctan.org/pkg/slashbox>

32 <http://tug.ctan.org/pkg/dcolumn>

33 <http://tug.ctan.org/pkg/rccol>

34 <http://tug.ctan.org/pkg/numprint>

35 <http://www.ctan.org/pkg/spreadtab>

36 <http://ctan.org/tex-archive/macros/latex/contrib/siunitx>

-
- 37 <http://www.ctan.org/tex-archive/graphics/pgf/contrib/pgfplots>
- 38 http://fr.wikibooks.org/wiki/LaTeX%2FFaire_des_tableaux
- 39 <http://nl.wikibooks.org/wiki/LaTeX%2FTabellen>
- 40 <http://pl.wikibooks.org/wiki/LaTeX%2FTabele>

`\resizebox{width}{height}{object}` can be used with `tabular` to specify the height and width of a table. The following example shows how to resize a table to 8cm width while maintaining the original width/height ratio.

```
\resizebox{8cm}{!}{  
  \begin{tabular}...\br/>  \end{tabular}  
}
```

Alternatively you can use `\scalebox{ratio}{object}` in the same way but with ratios rather than fixed sizes:

```
\scalebox{0.7}{  
  \begin{tabular}...\br/>  \end{tabular}  
}
```

Both `\resizebox` and `\scalebox` require the `graphicx` package. To tweak the space between columns (LaTeX will by default

There are several situations where you might want to create a title in a custom format, rather than in the format natively supported by LaTeX. For shorter documents such as basic articles, the output of `\maketitle` is often adequate, but longer documents (such as books and reports) often require more involved formatting. While it is possible to change the output of `\maketitle`, it can be complicated even with minor changes to the title. In such cases it is often better to create the title from scratch, and this section will show you how to accomplish this.

8.1. Standard Title Pages

Many document classes will form a title page for you. One must specify what to fill it with using these commands placed in the top matter¹:

```
\title{The Triangulation of Titling Data in
```

¹ Chapter 3.1.2 on page 81

```
Non-Linear Gaussian Fashion via  $\rho$  Series}
\date{October 31, 475}
\author{John Doe\ Magic Department, Richard Miles University
       \and Richard Row, \LaTeX\ Academy}
```

Commonly the date is excluded from the title page by using `\date{}`. It defaults to `\today` if not in the source file.

To form a title page, use

```
\maketitle
```

This should go after the preceding commands. For most document styles, this will form a separate page, while the `article` document style will place the title on the top of the first page. Note that the `abstract2` environment should precede the `\maketitle` command in AMS documents.

² <http://en.wikibooks.org/wiki/LaTeX%2FFormatting%23Abstracts>

Footnotes within the title page can be specified with the `\thanks` command. For example, one may add

```
\author{John Doe\thanks{Funded by NASA Grant \#42}}
```

The `\thanks` command can also be used in the `\title`.

It is dependent on the document class which commands are used in the title page generated by `\maketitle`. For example, the `amsart` uses commands such as `\address`, `\dedicatory`, `\email` and more in the title page while other classes may only use `\title`.

8.2. Custom Title Pages

8.2.1. Create the title

Normally, the benefit of using LaTeX instead of traditional word processing programs is that LaTeX frees you to concentrate on content by handling margins, justification, and other typesetting concerns. On the other hand, if you want to write your own title format, it is exactly the opposite: you have to take care of everything--this time LaTeX will do nothing to help you. It can be challenging to create your own title format since LaTeX was not designed to be graphically interactive in the adjustment of layout. The process is similar to working with raw HTML with the added step that each time you want to see how your changes look, you have to re-compile the source. While this may seem like a major inconvenience, the benefit is that once the customized title format has been written, it serves as a template for all other documents that would use the title format you have just made. In other words, once you have a

layout you like, you can use it for any other documents where you would like the same layout without any additional fiddling with layout.

First step: since you'll be working only on the first page of your document and you'll have to compile very often, you don't have to compile the whole document each time, you only need to take a look at the first page. That is why we'll first create a dummy document for preparing the title and then we'll simply include it within the existing big document we are writing. Call the dummy document `test_title.tex` and put the following code in it:

```
\documentclass[pdftex,12pt,a4paper]{report}
\usepackage[pdftex]{graphicx}
\newcommand{\HRule}{\rule{\linewidth}{0.5mm}}
\begin{document}
\input{./title.tex}
\end{document}
```

It is meant to be compiled with `pdflatex` to create a PDF in output. It is a very basic document, but take care that it has the same settings of the document you are writing, so the output won't change when you include the title in your document. In this case (see the first line) the font size is set to 12pt and the paper size is an A4. The package `graphicx` is included to insert an image in the title. Then a command is defined called `\HRule`; it will just insert a horizontal line whose length is like the size of the paper and whose thickness is 0.5 mm. If you want you can change its settings in the definition. Finally the document starts and it simply includes the `title.tex` file, that must be placed in the same directory of our dummy file `test_title.tex`.

Now create the `title.tex` and write in it:

```
\begin{titlepage}  
  
\end{titlepage}
```

all the things you want to put in the title must be inside the `titlepage` environment. Now if you compile `test_title.tex` you will see a preview of your title in the `test_title.pdf` file. Here is what you need to know to write your title:

Alignment

if you want to center some text just use `\begin{center} ... \end{center}`. If you want to align it differently you can use the environment `flushright` for *right*-alignment and `flushleft` for *left*-alignment.

Images

the command for including images is the following (the example is for a small logo, but you can introduce any image of any size): `\includegraphics[width=0.15\textwidth]{./logo}`. There is no `\begin{figure}` as you usually do because you don't want it

to be floating³, you just want it there where you placed it. When handling it, remember that it is considered like a big box by the TeX engine.

Text size

If you want to change the size of some text just place it within brackets, *{like this}*, and you can use the following commands (in order of size): `\Huge`, `\huge`, `\LARGE`, `\Large`, `\small`, `\footnotesize`, `\tiny`. So for example:

```
{\large this text is slightly bigger than normal}, this one is not
```

```
\normalsize is used to create text at the default size for the document.
```

New lines

you can force the start of a new line by `\\`. If you want to add more vertical space you don't need to use several new-line commands, just insert some

vertical space. For example, this way `\[1cm]` you start a new line after having left 1 cm of empty space.

Date

you can insert the date of the current day with the command `\today`. If you do not wish to insert any date, keep it blank e.g. `\date{}`

Filling the page

the command `\vfill` keeps on adding empty spaces until the page is full. If you put it in the page, you are sure that all the following text will be placed at the bottom of the page.

8.2.2. A practical example

All these tips might have made you confused. Then, here is a practical example. Get the `test_title.tex` described above and here is an example of a

title.tex. On the right you can see the output after you compile test_
title.tex in PDF:



UNIVERSITY OF BEER

FINAL YEAR PROJECT

Lager brewing techniques

Authors:
John Smith

Supervisor:
Dr. Mark Brown

```
% Upper part of the page
\includegraphics[width=0.15\textwidth]{./logo}\[1cm]
```

```
\textsc{\LARGE University of Beer}\[1.5cm]
```

```
\textsc{\Large Final year project}\[0.5cm]
```

April 17, 2012

```
% Title
\HRule \[0.4cm]
{ \huge \bfseries Lager brewing techniques}\[0.4cm]
```

```
\HRule \[1.5cm]
```

```
% Author and supervisor
\begin{minipage}{0.4\textwidth}
\begin{flushleft} \large
\emph{Author:}\[
```

Figure 35

The picture is from a file called `logo.png` that is in the same directory of both `title.tex` and `test_title.tex`. Since I wanted to insert both the author and supervisor names properly aligned I used a trick: I created two small minipages, one on left and one on the right. Their width is a bit less than half of page width (as you can see, they are exactly 40% of the text width). Within the minipages I have used different alignments. Using `\vfill` I could write the date exactly at the bottom of the page.

As you can see, the code looks "dirtier" than standard LaTeX source because you have to take care of the output as well. If you start changing font's output it will get more confused, but you can do it: it's only for the title and your complicated code will be isolated from all the rest within its own file `title.tex`.

8.2.3. Integrating the title page

Assuming that your title page is now contained in a file named `title.tex`, it must be placed in the same directory as the main document. In order to integrate it, the input command must be used by placing `\input{./title.tex}` at the top of the document. Don't forget to add the commands `\usepackage[pdfTeX]{graphicx}` and `\newcommand{\HRule}{\rule{\linewidth}{0.5mm}}` in the preamble section as well.

For example, the top section of your document would look like:

```
...
\usepackage[pdfTeX]{graphicx}
\newcommand{\HRule}{\rule{\linewidth}{0.5mm}}
\begin{document}
\input{./title.tex}
```

\tableofcontents

...

8.2.4. Additional documentation and packages

The `titlpages` package presents many styles of designs for title pages. Italian users may also want to use the `frontespizio` package.

9. Page Layout

Latex and the document class will normally take care of page layout issues for you. For submission to an academic publication, this entire topic will be out of your hands, as the publishers want to control the presentation. However, for your own documents, there are some obvious settings that you may wish to change: margins, page orientation and columns, to name but three. The purpose of this tutorial is to show you how to configure your pages.

9.1. Page dimensions

A page in Latex is defined by myriad internal parameters. Each parameter corresponds to the length of an element of the page, for example, `\paperheight` is the physical height of the page. Here you can see a diagram showing all the variables defining the page. All sizes are given in Postscript Points (pt), there are 72pt in an inch or $1\text{pt} = 0.3527\text{mm}$.

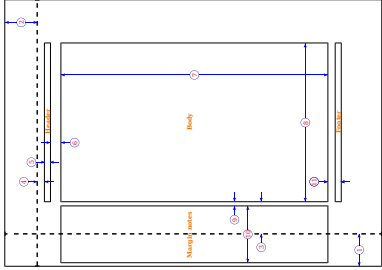


Figure 36

1. `one inch + \hoffset`
2. `one inch + \voffset`
3. `\oddsidemargin = 31pt`
4. `\topmargin = 20pt`

5. `\headheight = 12pt`
 6. `\headsep = 25pt`
 7. `\textheight = 592pt`
 8. `\textwidth = 390pt`
-
1. `\marginparsep = 10pt`
 2. `\marginparwidth = 35pt`
 3. `\footskip = 30pt`
-
- `\marginparpush = 7pt` (not shown)
 - `\hoffset = 0pt`
 - `\voffset = 0pt`
 - `\paperwidth = 597pt`
 - `\paperheight = 845pt`

It will not have been immediately obvious - because it doesn't really cause any serious problems - that the default page size for all standard document classes is *US letter*. This is shorter by 18 mm (about 3/4 inch), and slightly wider by 8 mm (about 1/4 inch), compared to A4 (which is the standard in almost all the

rest of the world). While this is not a serious issue (most printers will print the document without any problems), it is possible to specify alternative sizes:

```
\documentclass[a4paper]{article}
```

The above example illustrates how to pass the optional argument to the `\documentclass`, which will then modify the page dimensions accordingly. The standard document classes that are a part of LaTeX are built to be fairly generic, which is why you have the flexibility of specifying the page size. Other classes may have different options (or none at all). Normally, 3rd party classes come with some documentation to let you know.

Readers from a word processing background are probably thinking why there is so much white space surrounding the text. There is a good reason, and it's all down to readability. Have a look in a few books, and pick a few lines at random. Count the number of characters per line. I bet the average is about 66. Studies have shown that it's easier to read text when there are 60-70 characters per line - and it would seem that 66 is the optimal number. Therefore, the page margins are set to ensure that readability remains as good as possible.

Also, white space is often left in the inner margin for the assumption that the document will be bound.

However, those who want to read on iPads or other handheld digital devices need to create documents without the extra whitespace. In order to create PDF documents with optimal handheld viewing, not only must the text field and margins be adjusted, so must the page size. To do this, use the setlength command to adjust the parameters "pdfpagewidth" and "pdfpagelength" to the appropriate dimensions. If you are looking for a sensible dimension, consider following the paper size used by the Supreme Court of the United States, 441pt. You by 666pt (or 6.125 inches by 9.25 inches), which looks great on the iPad. You could also use the Supreme Court's text field size of 297 pt by 513 pt, but this is too wide for font's other than Century Schoolbook, the font required by the Supreme Court.

If you wish to change the margins of your document, there are many ways to do so:

- Simply use the `fullpage` package for somewhat standardized smaller margins (around an inch):

```
\usepackage{fullpage}
```

For an even greater effect give it the `cm` option (around 1.5cm):

```
\usepackage[cm]{fullpage}
```

- Use the `a4wide` package for a page with A4 document size with smaller margins.
- Use the `geometry` package. This package allows you to specify the 4 margins without needing to remember the particular page dimensions commands. You can enter the measures in centimeters and inches as well. Use `cm` for centimeters and `in` for inches after each value (e.g: 1.0in or 2.54cm). Note that by default (i.e. without any options) this package already reduces the margins, so for a 'standard layout' you may not need to specify anything. These values are relative to the edge of paper (0in) and go inward it. It may be implemented as follows:

```
\usepackage[top=tlength, bottom=blength, left=llength,
right=rlength]{geometry}
```

- **Edit individual page dimension variables** described above, using the `\addtolength` and `\setlength` commands. For instance,

```
\oddsidemargin=-1cm
\setlength{\textwidth}{6.5in}
\addtolength{\voffset}{-5pt}
```

Additionally, there are several packages designed to solve the problem of varying page sizes, which override any defaults setup by the document class. One of the most versatile packages for page layout is the `geometry` package. For instance, to set the page size, add the following to your preamble:

```
\usepackage[a4paper]{geometry}
```

The `geometry` package has many pre-defined page sizes, like `a4paper`, built in. Others include: `a0paper`, `a1paper`, ..., `a6paper`,

b0paper, b1paper, ..., b6paper, letterpaper, legalpaper, executivepaper.

To explicitly change the paper dimensions using the `geometry` package, the `paperwidth` and `paperheight` options can be used. For example:

```
\usepackage[margin=1in, paperwidth=5.5in,  
paperheight=8.5in]{geometry}
```

- A quick way to eliminate the difference in position between even and odd numbered pages would be setting the values to **evensidemargin** and **oddsidemargin** to the half of odd's default:

```
\setlength{\oddsidemargin}{15.5pt}  
\setlength{\evensidemargin}{15.5pt}
```

The value of **evensidemargin** is larger than **oddsidemargin** in the two-sided default layout, as one could wish to write notes on the side of the page. The side for the large margin is chosen opposite to the side where pages are conjoined together.

9.1.1. Top margin above Chapter

The top margin above a chapter can be changed using the `titlesec` package.
Example: <http://www.ctex.org/documents/packages/layout/titlesec.pdf>¹

```
\usepackage{titlesec}
\titlespacing*{\chapter}{0pt}{-50pt}{20pt}
\titlefo
rmat{\chapter}[display]{\normalfont\huge\bfseries}{\chaptertitlename}
\thechapter}{20pt}{\Huge}
```

The command `\titleformat` must be used when the spacing of a chapter is changed. In case of a section this command can be omitted.

9.1.2. Page size issues

If you intend to get a pdf in the end, there are basically three ways:

¹ <http://www.ctex.org/documents/packages/layout/titlesec.pdf>

```
TeX => PDF
TeX => DVI => PDF
TeX => DVI => PS => PDF
```

Which are in general obtained with

```
pdflatex myfile          # TeX => PDF
latex myfile             # TeX => DVI
dvi2pdf myfile           # DVI => PDF
dvips myfile -o myfile.ps # DVI => PS
ps2pdf myfile.ps myfile.pdf # PS => PDF
```

With all the available Ghostscript versions, the safest way to always get the right paper size in the end is to add

```
\documentclass[... ,a4paper,...]{...}
\special{papersize=210mm,297mm}
```

to the tex file, `-t a4` after `dvips` and `-sPAPERSIZE=a4` after the `ps2pdf`. For `pdflatex` to work fine, using the package `geometry` usually works.

If you want US Letter instead, replace "210mm,297mm" by "8.5in,11in" and "a4" by "letter".

9.2. Page orientation

When you talk about changing page orientation, it usually means changing to landscape mode, since portrait is the default. I shall introduce two slightly different styles of changing orientation.

The first is for when you want all of your document to be in landscape from the very beginning. There are various packages available to achieve this, but the one I prefer is the `geometry` package. All you need to do is call the package, with *landscape* as an option:

```
\usepackage[landscape]{geometry}
```

Although, if you intend to use `geometry` to set your paper size, don't add the `\usepackage` commands twice, simply string all the options together, separating with a comma:

```
\usepackage[a4paper,landscape]{geometry}
```

9.2.1. Change orientation of specific part

The second method is for when you are writing a document in portrait, but you have some contents, like a large diagram or table that would be displayed better on a landscape page. However, you still want the consistency of your headers and footers appearing the same place as the other pages.

The `landscape` package is for this very purpose. It supplies a `landscape` environment, and anything inside is basically rotated. No actual page dimensions are changed. This approach is more applicable to books or reports than to typical academic publications. Using `pdflandscape` instead of `landscape` when

generating a PDF document will make the page appear right side up when viewed: the single page that is in landscape format will be rotated, while the rest will be left in portrait orientation.

Also, to get a table to appear correctly on a landscaped page, one must place the `tabular` environment inside a `table` environment, which is itself inside the landscape environment. e.g., it should look like this:

```
\begin{landscape}  
\begin{table}  
\centering % optional, probably makes it look better to have it  
centered on the page  
\begin{tabular}{...}  
.....  
\end{tabular}  
\end{table}  
\end{landscape}
```

9.2.2. Change orientation of floating environment

If you use the above code, you will see that the table is inserted where it is in the code. It will not be floated! To fix this you need the package `rotating`

```
\begin{table}[p]
  \centering
  \rotatebox{90}{
    \begin{minipage}{\textheight}
    \begin{tabular}{|l|l|l|l|}
      \dots
    \end{tabular}
    \end{minipage}
  }
\end{table}
```

9.3. Page styles

Page styles in Latex terms refers not to page dimensions, but to the running headers and footers of a document. These headers typically contain document titles, chapter or section numbers/names, and page numbers.

9.3.1. Standard page styles

The possibilities of changing the headers in plain Latex are actually quite limited. There are two commands available: `\pagestyle{style}` will apply the specified style to the current and all subsequent pages, and `\thispagestyle{style}` will only affect the current page. The possible styles are:

empty

Both header and footer are clear

plain

Header is clear, but the footer contains the page number in the center.

headings

Footer is blank, header displays information according to document class (e.g., section name) and page number top right.

myheadings

Page number is top right, and it is possible to control the rest of the header.

With `myheadings`, the commands `\markright` (in the standard document classes, book, report and article) and `\markboth` (only in the book class) are used to control the headings. The following commands placed at the beginning of an article document will set the header of all pages to contain "John Smith" top left, "On page styles" centered and the page number top right:

```
\pagestyle{myheadings}
\markright{John Smith\hfill On page styles\hfill}
```

An issue to look out for is that the major sectioning commands (`\part`, `\chapter` or `\maketitle`) specify a `\thispagestyle{plain}`. So, if you wish to suppress all styles by inserting a `\pagestyle{empty}` at the beginning of your document, then the style command at each section will override your initial rule, for those pages only. To achieve the intended result one can follow the new section commands with `\thispagestyle{empty}`. The `\part` command, however, cannot be fixed this way, because it sets the page style, but also advances to the next page, so that `\thispagestyle{}` cannot be applied to that page. Another approach is to simply write `\usepackage{nopageno}` in the preamble. This package will make `\pagestyle{plain}` have the same effect as `\pagestyle{empty}`, effectively suppressing page numbering when it is used.

9.3.2. Customising with `fancyhdr`

To get better control over the headers, one can use the package `fancyhdr` written by Piet van Oostrum. It provides several commands that allow you to customize the header and footer lines of your document. For a more complete guide, the author of the package produced this `documentation2`.

The tricky problem when customizing headers and footers is to get things like running section and chapter names in there. Standard LaTeX accomplishes this with a two-stage approach (although an alternative one-stage mechanism is provided by the package `titleps`). In the header and footer definition, you use the commands

```
\rightmark
```

and

```
\leftmark
```

² <http://www.ctan.org/tex-archive/macros/latex/contrib/fancyhdr/fancyhdr.pdf>

to represent the current section and chapter heading, respectively. The values of these two commands are overwritten whenever a chapter or section command is processed. For ultimate flexibility, the

`\chapter`

command and its friends do not redefine

`\rightmark`

and

`\leftmark`

themselves. They call yet another command (

`\chaptermark`

,

`\sectionmark`

, or

`\subsectionmark`

) that is responsible for redefining

`\rightmark`

and

`\leftmark`

, except if they are starred -- in such a case,

`\chaptermark`

, etc., must be used after the sectioning command if header and footer lines are to be updated.

If you want to change the look of the chapter name in the header line, you need only "renew" the

`\chaptermark`

command.

To begin, add the following lines to your preamble:

```
\usepackage{fancyhdr}
\setlength{\headheight}{15.2pt}
\pagestyle{fancy}
```


The second line will prevent LaTeX from giving a warning. Both the header and footer comprise three elements each according to its horizontal position (left, centre or right). To set their values, the following commands are available:

```
\lhead[<lh-even>] {<lh-odd>\head[<ch-even>] {<ch-odd>\head[<rh-even>] {<rh-odd>
\lfoot[<lf-even>] {<lf-odd>\foot[<cf-even>] {<cf-odd>\foot[<rf-even>] {<rf-odd>
```

Hopefully, the behaviour of the above commands is fairly intuitive: if it has *head* in it, it affects the head etc, and obviously, *l*, *c* and *r* means left, centre and right respectively. Documents can be either one- or two-sided. Articles are by default one-sided, books are two-sided. Two-sided documents differentiate the left (even) and right (odd) pages, whereas one-sided do not.

Watch out: if you provide long text in two different "parts" only in the footer or only in the header, you might see overlapping text. There are special commands you can use as arguments:

number of the current page

`\thepage`

current chapter name printed like
"CHAPTER 3. THIS IS THE
CHAPTER TITLE"

`\leftmark`

current section name printed like
"1.6. THIS IS THE SECTION
TITLE"

`\rightmark`

the name *chapter* in the current language. If this is English, it will display "Chapter"

`\chaptername`

current chapter number

`\thechapter`

current section number

`\thesection`

Note that

`\leftmark`

and

`\rightmark`

convert the names to uppercase, whichever was the formatting of the text. If you want them to print the actual name of the chapter without converting it to uppercase use the following command:

```
\renewcommand{\chaptermark}[1]{\markboth{#1}}{}}
\renewcommand{\sectionmark}[1]{\markright{#1}}{}}
```

now

```
\leftmark
```

and

```
\rightmark
```

will just print the name of the chapter and section, without number and without affecting the formatting. Note that these redefinitions must be inserted *after* the first call of

```
\pagestyle{fancy}
```

. The standard book formatting of the

```
\chaptermark
```

is:

```
\renewcommand{\chaptermark}[1]{\markboth{\MakeUppercase{\chaptername}\
\thechapter.\ #1}}{}}
```

Moreover, with the following commands you can define the thickness of the decorative lines on both the header and the footer:

```
\renewcommand{\headrulewidth}{0.5pt}  
\renewcommand{\footrulewidth}{0pt}
```

The first line for the header, the second for the footer. Setting it to zero means that there will be no line.

An example:

```
\fancyhf{}  
  
\lhead{Andrew Roberts}  
\rhead{\today}  
\rfoot{\thepage}
```

It is often necessary to clear any defaults or a previous style definition, and the first line of the above example will do this. The commands are an alternative

interface to customising the headers/footers that `fancyhdr` offers, and so by not passing anything to them, it assumes that you want it all blank.

The result of these commands will put my name at the top left, todays date at the top right, and the current page number at the bottom right of the page. Even if the document class was two-sided, because no optional text has been supplied for the even pages, the style will be used for all pages.

This approach has a serious bad point: some pages like the title or the beginning of each chapter have no header or footer, but with the code we have shown *every* page will get the same output. There is a way to solve this problem: you can use the *fancyplain* style. If you do so, you can use the command

```
\fancyplain{...}{...}
```

inside

```
\thead{...}
```

etc.

When LaTeX wants to create a page with an empty style, it will insert the first argument of `fancyplain`, in all the other cases it will use the second argument. So, an improved version of the previous code would be:

```
\pagestyle{fancyplain}
\fancyhf{}
\lhead{\fancyplain}{Andrew Roberts}
\rhead{\fancyplain}{\today}
\rfoot{\fancyplain}{\thepage}
```

It has the same behavior of the previous code, but you will get empty header and footer in the title and at the beginning of chapters.

For two-sided, it's common to mirror the style of opposite pages, you tend to think in terms of *inner* and *outer*. So, the same example as above for two-sided is:

```
\lhead[Andrew Roberts]{}
\rhead[Andrew Roberts]{}
\lfoot[Andrew Roberts]{}
\rfoot[Andrew Roberts]{}
\pagestyle{fancyplain}
```

```
\rhead[] {Andrew Roberts}
\lhead[] {\today}
\rhead[] {\today} {}
\lfoot[] {\thepage} {}
\rfoot[] {\thepage}
```

This is effectively saying my name is top outer, todays date is top inner, and current page number is bottom outer. You can use the `fancyplain` command within them for two-sided documents, too.

As an example, here is the complete code of a basic configuration you could use for a real document:

```
\usepackage{fancyhdr}
\setlength{\headheight}{15pt}

\pagestyle{fancyplain}
\renewcommand{\chaptermark}[1]{\markboth{#1}{} }

\lhead{\fancyplain}{\thepage}
\chead{}
\rhead{\fancyplain}{\textit{\leftmark}}
```



```
\lfoot{}  
\cfoot{}  
\rfoot{}
```

NB. If you want to make the article class two-sided, use `\documentclass[twoside]{article}`.

9.3.3. Another approach with `fancyhdr`

If you want to get different style for even and odd pages, there is another possible way, still using `fancyhdr`. Start again with:

```
\fancyhf{}
```

it will just delete the current heading/footer configuration, so you can make your own. Now you can create what you want using the command `\fancyhead` for header and `\fancyfoot` for footer. They work in the same way, so we'll explain only the first one. The syntax is:

```
\fancyhead[selectors]{output you want}
```

The selectors are the following:

E	even page
O	odd page
L	left side
C	centered
R	right side

so **CE** will refer to the center of the even pages and **RO** to the right side of the odd pages. Whether it is header or footer, depends if you are using `\fancyhead` or `\fancyfoot`. You can use multiple selectors separated by a comma. Here is an example:

```
\fancyhead[CE]{Author's Name}  
\fancyhead[CO]{\today}  
\fancyfoot[LE,RO]{\thepage}
```

it will print author's name on the center of the header of the even pages, the date of the current day on the center of the odd pages and the current page number on the left side of even pages *and* on the right size of the odd pages. Finally, in order to have the pages at the beginning of any chapter really plain, you could redefine the *plain* style, for example to have a really plain page when you want. The command to use is `\fancypagestyle{plain}{...}` and the argument can contain all the commands explained before. An example is the following:

```
\fancypagestyle{plain}{%  
\fancyhf{}% remove everything  
\renewcommand{\headrulewidth}{0pt}% remove lines as well  
\renewcommand{\footrulewidth}{0pt}}
```

Finally, here is the complete code of a possible style you could use for a two-sided document:

```
\usepackage{fancyhdr}  
\setlength{\headheight}{15pt}
```

```

\pagestyle{fancy}
\renewcommand{\chaptermark}[1]{\markboth{#1}{}}
\renewcommand{\sectionmark}[1]{\markright{#1}{}}

\fancyhf{}
\fancyhead[LE,RO]{\thepage}
\fancyhead[RE]{\textit{\nouppercase{\leftmark}}}
\fancyhead[LO]{\textit{\nouppercase{\rightmark}}}

\fancypagestyle{plain}{%
\fancyhf{} % remove everything
\renewcommand{\headrulewidth}{0pt} % remove lines as well
\renewcommand{\footrulewidth}{0pt}}

```

9.3.4. Page n of m

Some people like to put the current page number in context with the whole document. LaTeX only provides access to the current page number. However, you can use the `lastpage` package to find the total number of pages, like this:

```
\usepackage{lastpage}  
...  
\cfoot{\thepage\ of \pageref{LastPage}}
```

Note the capital letters. Also, add a backslash after `\thepage` to ensure adequate space between the page number and 'of'. And recall, when using references, that you have to run LaTeX an extra time to resolve the cross-references.

9.3.5. Alternative packages

Other packages for page styles are `scrpage2`, very similar to `fancyhdr`, and `titlps`, which takes a one-stage approach, without having to use `\leftmark` or `\rightmark`.

9.4. Multi-column pages

It is common to see articles and conference proceedings formatted with two columns of text. However, such publishers will usually provide you with their own document class, which automatically implements this format, without you having to do anything. It is very easy to format your page in this way. If you are using a standard LaTeX document class, then you can simply pass the optional argument *twocolumn* to the document class: `\documentclass[twocolumn]{article}` which will give the desired effect.

While this simple addition will do the job 9 out of 10 times, it is widely acknowledged that there are many limitations of this approach, and that the `multicol` package is much more useful for handling multiple columns. It has several advantages:

- Can support up to ten columns.

- Implements a *multicols* environment, therefore, it is possible to mix the number of columns within a document.
- Additionally, the environment can be nested inside other environments, such as `figure`.
- `Multicol` outputs *balanced* columns, whereby the columns on the final page will be of roughly equal length.
- Vertical rules between columns can be customised.
- Column environments can be easily customised locally or globally.

Floats are not fully supported by this environment. It can only cope if you use the starred forms of the float commands (e.g., `\begin{figure*}`) which makes the float span all columns. This is not hugely problematic, since floats of the same width as a column may be too small, and you would probably want to span them anyway. See this section³ for a more detailed discussion.

To create a typical two-column layout:

```
\begin{multicols}{2}  
lots of text  
\end{multicols}
```

The parameter `\columnseprule` holds the width of the vertical rules. By default, the lines are omitted as this parameter is set to a length of 0pt. Do the following before the beginning of the environment:

```
\setlength{\columnseprule}{1pt}
```

This will draw a thin line of 1pt in width. A thick line would not look very pleasing, however, you are free to put in any length of your choosing. Also, to change the horizontal space in between columns (the default is set at 10pt, which is quite narrow) then you need to change the `\columnsep` parameter:

```
\setlength{\columnsep}{20pt}
```

To force a break in a column, the command `\columnbreak` is used in a `multicols` environment.

9.5. Manual page formatting

There may be instances, especially in very long documents, such as books, that LaTeX will not get all page breaks looking as good as it could. It may, therefore, be necessary to manually tweak the page formatting. Of course, you should only do this at the very final stage of producing your document, once all the content is complete. LaTeX offers the following:

<code>\newline</code>	Breaks the line at the point of the command.
<code>//</code>	Breaks the line at the point of the command; it is a shorter version of the previous command, but it does exactly the same thing.

<code>*</code>	<p>Breaks the line at the point of the command and additionally prohibits a page break after the forced line break</p>
<code>\linebreak [number]</code>	<p>Breaks the line at the point of the command. The <i>number</i> you provide as an argument represents the priority of the command in a range from 0 (it will be easily ignored) to 4 (do it anyway). LaTeX will try to produce the best line breaks possible. If it cannot, it will decide whether including the linebreak or not according to the priority you have provided.</p>
<code>\newpage</code>	<p>Ends the current page and starts a new one.</p>

<code>\pagebreak [number]</code>	Breaks the current page at the point of the command. The optional <i>number</i> argument sets the priority in a scale from 0 to 4.
<code>\nopagebreak [number]</code>	Stops the page being broken at the point of the command. The optional <i>number</i> argument sets the priority in a scale from 0 to 4.
<code>\clearpage</code>	Ends the current page and causes any floats encountered in the input, but yet to appear, to be printed.

9.6. Widows and orphans

w:Widows and orphans⁴ In professional books, it's not desirable to have single lines at the beginning or end of a page. In typesetting such situations are called 'widows' and 'orphans'. Normally it is possible that widows and orphans appear in LaTeX documents. You can try to deal with them using manual page formatting, but there's also an automatic solution.

LaTeX has a parameter for 'penalty' for widows and orphans ('club lines' in LaTeX terminology). With the greater penalty LaTeX will try more to avoid widows and orphans. You can try to increase these penalties by putting following commands in your document preamble:

```
\widowpenalty=300  
\clubpenalty=300
```

4 <http://en.wikipedia.org/wiki/Widows%20and%20orphans>

If this does not help, you can try increasing these values even more, to a maximum of 10000. However, it is not recommended to set this value too high, as setting it to 10000 forbids LaTeX from doing this altogether, which might result in strange behavior.

It also helps to have rubber band values for the space between paragraphs:

```
\setlength{\parskip}{3ex plus 2ex minus 2ex}
```

Alternatively, you can use the `needspace` package to *reserve* some lines and thus to prevent page breaking for those lines.

```
\needspace{5\baselineskip}  
Some  
text  
on  
5  
lines.
```

9.7. Summary

This tutorial is relatively short, largely due to the fact that the whole LaTeX ethos is to concentrate on the content, and let LaTeX (and/or other typographers who have developed suitable document classes) decide on the best presentation. The next step to achieve greater control of page layout is to set about designing your own class. Unfortunately, that is not a straightforward task, and is often best left to the professionals!

10. Paragraph Formatting

Altering the paragraph formatting is not often required, especially in academic writing. However, it is useful to know, and applications tend to be for formatting text in floats, or other more exotic documents.

10.1. Paragraph Alignment

Paragraphs in LaTeX are usually fully justified (i.e., flush with both the left and right margins). For whatever reason, should you wish to alter the justification

of a paragraph, there are three environments at hand, and also LaTeX command equivalents.

Alignment	Environment	Command
Left justified	flushleft	\raggedright
Right justified	flushright	\raggedleft
Center	center	\centering

All text between the

`\begin`

and

`\end`

of the specified environment will be justified appropriately. The commands listed are for use within other environments. For example,

P

(paragraph) columns in



Warning

There is no way to set full justification. It means that if you do not enclose the previous 3 commands into a group, the rest of the document will be affected. So the right way of doing this with commands is

```
{  
  \raggedleft  
  Some text flushed right.  
}
```

10.2. Paragraph Indents

By default, the first paragraph after a heading follows the standard Anglo-American publishers' practice of no indentation. The size of subsequent paragraph indents are determined by a parameter called

```
\parindent
```

. The default length that this constant holds is set by the document class that you use. It is possible to override using the

```
\setlength
```

command.

```
\setlength{\parindent}{1cm}
```

This will set paragraph indents to 1cm.

Be careful, however, if you decide to set the indent to zero, then it means you will need a vertical space between paragraphs in order to make them clear. The space between paragraphs is held in

`\parskip`

, which could be altered in a similar fashion as above. However, this parameter is used elsewhere too, such as in lists, which means you run the risk of making various parts of your document look very untidy by changing this setting. If you want to use the style of having no indentation with a space between paragraphs, use the `\parskip` package, which does this for you, while making adjustments to the spacing of lists and other structures which use paragraph spacing, so they don't get too far apart. Add this to the preamble:

```
\usepackage{parskip}
```

To indent subsequent lines of a paragraph, use the TeX command

```
\hangindent
```

. (While the default behaviour is to apply the hanging indent after the first line, this may be changed with the

```
\hangafter
```

command.) An example follows.

`\hangindent=0.7cm` This paragraph has an extra indentation at the left.

The TeX commands

`\leftskip`

and

`\rightskip`

add additional space to the left and right sides of each line, allowing the formatting for subsequent paragraphs to differ from the overall document margins. This space is in addition to the indentation added by

`\parindent`

and

`\hangindent`

.

To change the indentation of the last line in a paragraph, use the TeX command

`\parfillskip`

White-space in LaTeX can also be made flexible (what Lamport calls "rubber" lengths). This means that values such as extra vertical space inserted before a paragraph

```
\parskip
```

can have a default dimension plus an amount of expansion minus an amount of contraction. This is useful on pages in complex documents where not every page may be an exact number of fixed-height lines long, so some give-and-take in vertical space is useful. You specify this in a

```
\setlength
```

command like this:

```
\setlength{\parskip}{1cm plus4mm minus3mm}
```

If you want to indent a paragraph that is not indented, you can use

```
\indent
```

at the beginning of the paragraph. Obviously, this will only have an effect when `\parindent` is not set to zero. If you want to indent the beginning of every section, you can use the `indentfirst` package, see the chapter about LaTeX/Packages¹ for more information.

To create a non-indented paragraph, you can use

```
\noindent
```

as the first command of the paragraph. This might come in handy when you start a document with body text and not with a sectioning command.

10.3. Line Spacing

To change line spacing in the whole document use the command

```
\linespread
```

covered in LaTeX/Text_Formatting#Spacing².

To change line spacing in specific environments do the following:

1. Add

```
\usepackage{setspace}
```

to the document preamble.

2. This then provides the following environments to use within your document:

- `doublespace`
 - all lines are double spaced.
 - `onehalfspace`
 - line spacing set to one-and-half spacing.
 - `singlespace`
 - normal line spacing.
3. To change line spacing to custom values, use the command

```
\setstretch{baselinestretch}
```


in the preamble to change line spacing for the whole document or use the environment

```
\begin{spacing}{2.5} ... \end{spacing}
```

to change the line spacing to whatever is greater, the document-wide or environment-wide one.

After declaring the package in the preamble the use of the command

```
\singlespacing
```

,

```
\doublespacing
```

, or

```
\onehalfspacing
```

will specify the line spacing for all sections and paragraphs until another command is used.

See the section on customizing lists³ below for information on how to change the line spacing in lists.

10.4. Special Paragraphs

10.4.1. Verbatim Text

There are several ways to introduce text that won't be interpreted by the compiler. If you use the

```
verbatim
```

environment, everything input between the *begin* and *end* commands are processed as if by a typewriter. All spaces and new lines are reproduced as given, and the text is displayed in an appropriate fixed-width font. Any LaTeX command will be ignored and handled as plain text. This is ideal for typesetting program source code. Here is an example:

```
\begin{verbatim}
The verbatim environment
  simply reproduces every
  character you input,
  including all s p a c e s!
\end{verbatim}
```

The `verbatim` environment simply reproduces every character you input, including all s p a c e s!

Figure 37

Note: once in the

```
verbatim
```

environment, the only command that will be recognized is

```
\end{verbatim}
```

. Any others will be output. The font size in the `verbatim` environment can be adjusted by placing a font size command⁴ before

```
\begin{verbatim}
```

. If this is an issue, then you can use the `alltt` package instead, providing an environment with the same name:

```
Verbatim extended with the ability  
to use normal commands. Therefore, it  
is possible to emphasize words in  
this environment, for example.
```

```
\begin{alltt}  
Verbatim extended with the ability  
to use normal commands. Therefore, Figure 38  
is possible to \emph{emphasize} words in  
this environment, for example.  
\end{alltt}
```

Remember to add

```
\usepackage{alltt}
```

to your preamble to use it though! Within the

```
alltt
```

environment, you can use the command

```
\normalfont
```

to get back the normal font. To write equations within the

```
alltt
```

environment, you can use

```
\(
```

and

```
\)
```

to enclose them, instead of the usual

```
$
```

```
.
```

When using

```
\textbf{ }
```

inside the

```
alltt
```

environment, note that the standard font has no bold TT font. Ttfonts has bold fonts: just add

```
\renewcommand{\ttdefault}{ttxtt}
```

after

```
\usepackage{alltt}
```

.

If you just want to introduce a short verbatim phrase, you don't need to use the whole environment, but you have the

```
\verb
```

command:

```
\verb+my text+
```

The first character following

```
\verb
```

is the delimiter: here we have used "+", but you can use any character you like but * and space;

`\verb`

will print verbatim all the text after it until it finds the next delimiter. For example, the code:

```
\verb|\textbf{Hi mate!}|
```

will print `\textbf{Hi mate!}`, ignoring the effect

`\textbf`

should have on text.

For more control over formatting, however, you can try the `fancyvrb` package, which provides a

`Verbatim`

environment (note the capital letter) which lets you draw a rule round the verbatim text, change the font size, and even have typographic effects inside the

`Verbatim`

environment. It can also be used in conjunction with the `fancybox` package and it can add reference line numbers (useful for chunks of data or programming), and it can even include entire external files.

Typesetting URLs

One of either the `hyperref` or `url` packages provides the

```
\url
```

command, which properly typesets URLs, for example:

```
Go to \url{http://www.uni.edu/~myname/best-website-ever.html} for my  
website.
```

will show this URL exactly as typed (similar to the

```
\verb
```

command), but the

```
\url
```


command also performs a hyphenless break at punctuation characters (only in PDFLaTeX, not in plain LaTeX+ dvips). It was designed for Web URLs, so it understands their syntax and will never break mid-way through an unpunctuated word, only at slashes and full stops. Bear in mind, however, that spaces are forbidden in URLs, so using spaces in

```
\url
```

arguments will fail, as will using other non-URL-valid characters.

When using this command through the `hyperref` package, the URL is "clickable" in the PDF document, whereas it is not linked to the web when using only the `url` package. Also when using the `hyperref` package, to remove the border placed around a URL, insert

```
pdfborder = {0 0 0 0}
```

(alternately

```
pdfborder = {0 0 0}
```

might work if the four zeroes does not) inside the

```
\hypersetup{}
```

You can put the following code into your preamble to change the style, how URLs are displayed to the normal font:

```
\urlstyle{same}
```

See also [LaTeX/Hyperlinks#Customization](#)⁵

Listing Environment

This is also an extension of the verbatim environment provided by the `moreverb` package. The extra functionality it provides is that it can add line numbers along side the text. The command:

```
\begin{listing}[step]{first line}
```

. The mandatory *first line* argument is for specifying which line the numbering shall commence. The optional *step* is the step between numbered lines (the default is 1, which means every line will be numbered).

⁵ Chapter 14.3 on page 445

To use this environment, remember to add

```
\usepackage{moreverb}
```

to the document preamble.

10.4.2. Multi-line comments

As we have seen, the only way LaTeX allows you to add comments is by using the special character `%`, that will comment out all the rest of the line after itself. This approach is really time-consuming if you want to insert long comments or just comment out a part of your document that you want to improve later, unless you're using an editor⁶ that automates this process. Alternatively, you can use the `verbatim` package, to be loaded in the preamble as usual:

```
\usepackage{verbatim}
```

(you can also use the `comment` package instead) you can use an environment called `comment` that will comment out everything within itself. Here is an example:

```
This is another
\begin{comment}
  rather stupid,
  but helpful
\end{comment}
example for embedding
comments in your document.
```

This is another example for embedding comments in your document.

Note that this won't work inside complex environments, like `math` for example. You may be wondering, why should I load a package called `verbatim` to have

the possibility to add comments? The answer is straightforward: commented text is interpreted by the compiler just like verbatim text, the only difference is that verbatim text is introduced within the document, while the comment is just dropped.

Alternatively, you can define a `\comment{}` command, by adding the following to the document's preamble:

```
\newcommand{\comment}[1]{}
```

Then, to comment out text, simply do something like this:

```
\comment{ This is a long comment and can extend over multiple lines,  
etc. }
```

10.4.3. Quoting text

LaTeX provides several environments for quoting text; they have small differences and they are aimed for different types of quotations. All of them are indented on either margin, and you will need to add your own quotation marks if you want them. The provided environments are:

quote

for a short quotation, or a series of small quotes, separated by blank lines.

quotation

for use with longer quotations, of more than one paragraph, because it indents the first line of each paragraph.

verse

is for quotations where line breaks are important, such as poetry. Once in, new stanzas are created with a blank line, and new lines within a stanza are indicated using the `newline` command,

```
//  
 . If a line takes up more than one line on the page, then all subsequent lines  
   are indented until explicitly separated with  
  
//  
 .
```

10.4.4. Abstracts

In scientific publications it is customary to start with an abstract which gives the reader a quick overview of what to expect. LaTeX provides the

```
abstract  
environment for this purpose. It is available in
```

```
article
```

```
and
```

```
report
```

document classes; it's not available in the

, but it's quite simple to create your own if you really need it.

10.5. Notes and References

This page uses material from Andy Roberts' Getting to grips with LaTeX with permission from the author.

11. Importing Graphics

Strictly speaking, LaTeX cannot manage pictures directly: in order to introduce graphics within documents, LaTeX just creates a box with the same size as the image you want to include and embeds the picture, without any other processing. This means you will have to take care that the images you want to include are in the right format to be included. This is not such a hard task because LaTeX supports the most common picture formats around.

11.1. The `graphicx` package

As stated before, LaTeX can't manage pictures directly, so we will need some extra help: we have to load the `graphicx` package in the preamble of our document:

```
\usepackage{graphicx}
```

This package accepts as an argument the external driver to be used to manage pictures; however, the latest version of this package takes care of everything by itself, changing the driver according to the compiler you are using, so you don't have to worry about this. Still, just in case you want to understand better how it works, here are the possible options you can pass to the package:

- `dvips` (default if compiling with *latex*), if you are compiling with *latex* to get a DVI and you want to see your document with a DVI or PS viewer.
- `dvipdfm`, if you are compiling with *latex* to get a DVI that you want to convert to PDF using *dvipdfm*, to see your document with any PDF viewer.

- `pdftex` (default if compiling with *pdf_latex*), if you are compiling with *pdf_ltex* to get a PDF that you will see with any PDF viewer.

but, again, you don't need to pass any option to the package because the default settings are fine in most of the cases.

In many respects, importing your images into your document using LaTeX is fairly simple... *once* you have your images in the right format that is! Therefore, I fear for many people the biggest effort will be the process of converting their graphics files. Now we will see which formats we can include and then we will see how to do it.

11.1.1. Document Options

The graphics and graphics packages recognize the "draft" and "final" options¹ given in the `\documentclass2[...]{...}` command at the start of the

1 Chapter 2.4.3 on page 75

2 Chapter 2.2.1 on page 46

file. Using "draft" as the option will suppress the inclusion of the image in the output file and will replace the contents with the name of the image file that would have been seen. Using "final" will result in the image being placed in the output file. The default is "final".

11.1.2. Supported image formats

As explained before, the image formats you can use depend on the driver that `graphicx` is using but, since the driver is automatically chosen according to the compiler, then the allowed image formats will depend on the compiler you are using.

Compiling with *latex*

The only format you can include while compiling with *latex* is Encapsulated PostScript³ (**EPS**).

The EPS format was defined by Adobe Systems for making it easy for applications to import postscript-based graphics into documents. Because an EPS file declares the size of the image, it makes it easy for systems like LaTeX to arrange the text and the graphics in the best way. EPS is a vector format⁴—this means that it can have very high quality if it is created properly, with programs that are able to manage vector graphics. It is also possible to store bit-map pictures within EPS, but they will need *a lot* of disk space.

Many graphics software packages have the ability to save images in the EPS format (extension is normally `.eps`). Here are some examples of software that can output EPS formats:

3 <http://en.wikipedia.org/wiki/Encapsulated%20PostScript>

4 <http://en.wikipedia.org/wiki/Vector%20graphics>

- Printing in an EPS file:
- Under Windows, PDFCreator⁵ is an open source software that can create PDF as well as EPS files. It installs a virtual printer that can be accessed from other software having a "print..." entry in their menu (virtually any program).
- Creating and converting vector graphics:
 - Commercial vector graphics software, such as Adobe Illustrator, CorelDRAW, and FreeHand are commonly used and can *read* and *write* EPS figures. However, these products are limited to Windows and Mac OS platforms.
 - Inkscape⁶ can save in vector EPS format, and it can run on multiple platforms. Inkscape cannot open EPS figures directly; however, with the *epstopdf* utility⁷ one can convert EPS into PDF and Inkscape can import

5 <http://sourceforge.net/projects/pdfcreator/>

6 <http://www.inkscape.org>

7 <http://www.ctan.org/tex-archive/support/epstopdf/>

PDF. From version 0.48, Inkscape has a special PDF+LaTeX output option (and for EPS/PS too). See Inkscape website⁸.

- Dia⁹ is a cross platform diagramming utility which can export eps images, or generate tex drawn using the *tikz* package.
- Creating and converting raster-only graphics to EPS:
- Commercial product Adobe Photoshop can save to EPS.
- GIMP¹⁰, has a graphical user interface, and it is multi-platform.
- For command-line:
 - Sam2p¹¹ (`convert`) or
 - ImageMagick¹² (`convert`) or
 - GraphicsMagick¹³ (`gm convert`).

8 <http://wiki.inkscape.org/wiki/index.php/LaTeX>

9 <http://live.gnome.org/Dia>

10 <http://www.gimp.org>

11 <http://pts.szit.bme.hu/sam2p/>

12 <http://www.imagemagick.org/>

13 <http://www.graphicsmagick.org/>

- These three programs operate much the same way, and can convert between most graphics formats. Sam2p however is the most recent of the three and seems to offer both the best quality and to result in the smallest files.
 - `imgtops`¹⁴. A lightweight graphics utility.
 - Creating publication-quality vector-based plots and charts:
 - `Gnuplot`¹⁵, producing scientific graphics since 1986.
 - `R`¹⁶, statistical and scientific figures.
 - `Generic Mapping Tools (GMT)`¹⁷, maps and a wide range of highly customizable plots.
 - `Gnumeric`¹⁸, spreadsheets has SVG, EPS, PDF export
 - `matplotlib`¹⁹, plotting library written in python, with PDF and EPS export.

14 <http://imgtops.sourceforge.net/>

15 <http://www.gnuplot.info>

16 <http://www.r-project.org/>

17 <http://gmt.soest.hawaii.edu/>

18 <http://projects.gnome.org/gnumeric/>

19 <http://matplotlib.sourceforge.net/>

There are some tricks to be able to import formats other than EPS into your DVI document, but they're very complicated. On the other hand, converting any image to EPS is very simple, so it's not worth considering them.

Compiling with `pdf \LaTeX`

If you are compiling with `pdf \LaTeX` to produce a PDF, you have a wider choice. You can insert:

- **JPG**, widely used on Internet, digital cameras, etc. They are the best choice if you want to insert photos
- **PNG**, a very common format (even if not as much as JPG); it's a lossless²⁰ format and it's the best choice for diagrams (if you were not able to generate a vector²¹ version) and screenshots

20 <http://en.wikipedia.org/wiki/lossless>

21 <http://en.wikipedia.org/wiki/Vector%20graphics>

- **PDF**, it is widely used for documents but can be used to store images as well. It supports both vector and bit-map²² images, but it's not recommended for the latter, as JPG or PNG will provide the same result using less disk space.
- **Vector formats** can be used with the help of Inkscape. There are instructions²³ on how to save your vector images in a PDF format understood by LaTeX and have LaTeX manage the text styles and sizes in the image automatically.
- **EPS** can be used with the help of the epstopdf package. Please see these instructions²⁴.

JPG and PNG are supported by any image processing program, so you just have to use the one you prefer. If you want to create high quality vector PDF to embed within your PDF document, you can use Inkscape²⁵: it supports many vector formats and so you can use it to convert from one to another. You

22 <http://en.wikipedia.org/wiki/Raster%20graphics>

23 <http://mirrors.ctan.org/info/svg-inkscape/InkscapePDFLaTeX.pdf>

24 <http://dirkraffel.com/2007/11/19/include-eps-files-in-latex>

25 <http://www.inkscape.org>

could also create your graphics directly with Inkscape. If you want to make mathematical plots, then Gnuplot²⁶ can save in any format.

Note, that EPS files cannot be used with pdflatex, however they can be converted to PDF using the epstopdf utility²⁷, included in most LaTeX distributions. This can be called automatically by LaTeX using the epstopdf package²⁸. In Windows, multiple files can be converted by placing the following line in a batch file²⁹ (a text file with a .BAT extension) in the same directory as the images:

```
for %%f in (*.eps) do epstopdf %%f
```

which can then be run from the command line.

26 <http://www.gnuplot.info>

27 <http://www.ctan.org/tex-archive/support/epstopdf/>

28 <http://www.ctan.org/tex-archive/help/Catalogue/entries/epstopdf-pkg.html>

29 <http://en.wikipedia.org/wiki/Batch%20file>

In Bash this can be done by:

```
$ for i in *.eps; do epstopdf "$i"; done
```

If `epstopdf`³⁰ produces whole page with your small graphics somewhere on it, use

```
$ epstopdf --gsopt=-dEPCrop foo.eps
```

or try using `ps2pdf`³¹ utility

```
$ ps2pdf -dEPCrop foo.eps
```

to crop final PDF.

30 <http://www.ctan.org/tex-archive/support/epstopdf/>

31 <http://svn.ghostscript.com/ghostscript/trunk/gs/doc/Ps2pdf.htm>

Images can be saved in multiple formats for different purposes. For example, a directory can have "diagram.pdf" for high-resolution printing, while "diagram.png" can be used for previewing on the monitor. You can specify which image file is to be used by `pdf\latex` through the preamble command:

```
\DeclareGraphicsExtensions{.pdf,.png,.jpg}
```

which specifies the files to include in the document, if files with the same basename exist, but with different extensions.

Acrobat Reader sometimes has problems with displaying colors correctly if you include graphics in PNG format with alpha channel. You can solve this problem by dropping the alpha channel. On Linux it can be achieved with `convert` from the ImageMagick³² program:

```
convert -alpha off input.png output.png
```

³² <http://en.wikipedia.org/wiki/ImageMagick>

11.1.1.3. Including graphics

Now that we have seen which formats we can include and how we could manage those formats, it's time to learn how to include them in our document. After you have loaded the `graphicx` package in your preamble, you can include images with `\includegraphics`, whose syntax is the following:

```
\includegraphics[attr1=val1, attr2=val2, ..., attrn=valn]{imagename}
```

As you should hopefully be aware by now, arguments in square brackets are optional, whereas arguments in curly braces are compulsory. The argument in the curly braces is the name of the image. Write it *without* the extension. This way the LaTeX compiler will look for any supported image format in that directory and will take the best one (EPS if the output is DVI; JPEG, PNG or PDF if the output is PDF). The variety of possible attributes that can be set is fairly large, so only the most common are covered below:

<code>width=xx</code>	Specify the preferred width of the imported image to <code>xx</code>	NB. Only <i>specifying either</i>
-----------------------	--	---

<code>height=xx</code>	Specify the preferred height of the imported image to <i>xx</i> .
<code>keepaspectratio</code>	This can be set to either <i>true</i> or <i>false</i> . When <i>true</i> , it will scale the image according to both height and width, but will not distort the image, so that neither width nor height are exceeded.
<code>scale=xx</code>	Scales the image by the desired scale factor. e.g, 0.5 to reduce by half, or 2 to double.
<code>angle=xx</code>	This option can rotate the image by <i>xx</i> degrees (counter-clockwise)
<code>trim=l b r t</code>	This option will crop the imported image by <i>l</i> from the left, <i>b</i> from the bottom, <i>r</i> from the right, and <i>t</i> from the top. Where <i>l</i> , <i>b</i> , <i>r</i> and <i>t</i> are lengths.
<code>clip</code>	For the <code>trim</code> option to work, you must set <code>clip=true</code> .

page=x

If the image file is a pdf file with multiple pages, this parameter allows you to use a different page than the first.

In order to use more than one option at a time, simply separate each with a comma. The order you give the options matters. E.g you should first rotate your graphic (with angle) and then specify its width.

Included graphics will be inserted just *there*, where you placed the code, and the compiler will handle them as "big boxes". As we will see in the next section³³, this can disrupt the layout; you'll probably want to place graphics inside floating objects.

Also note that the trim option does not work with XeLaTeX.

The *star* version of the command will work for .eps files only. For a more portable solution, the standard way should take precedence. The star command will take the crop dimension as extra parameter:

```
\includegraphics*[100,100][300,300]{mypicture}
```

11.1.4. Examples

OK, it's time to see `graphicx` in action. Here are some examples:

```
\includegraphics{chick}
```

This simply imports the image, without any other processing. However, it is very large (so I won't display it here!). So, let's scale it down:

```
\includegraphics[scale=0.5]{chick}
```



Figure 39

This has now reduced it by half. If you wish to be more specific and give actual lengths of the image dimensions, this is how to go about it:

```
\includegraphics[width=2.5cm]{chick}
```



Figure 40

One can also specify the scale with respect to the width of a line in the local environment (`\linewidth`), the width of the text on a page (`\textwidth`) or the height of the text on a page (`\textheight`) (pictures not shown):

```
\includegraphics[width=\linewidth]{chick}
```

```
\includegraphics[width=\textwidth]{chick}
```

```
\includegraphics[height=\textheight]{chick}
```

To rotate (I also scaled the image down):

```
\includegraphics[scale=0.5, angle=180]{
```

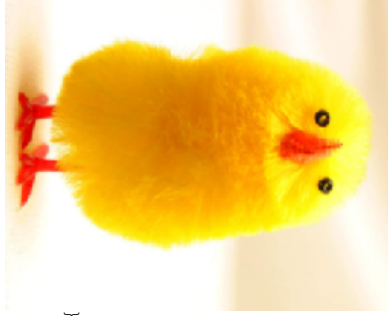


Figure 41

And finally, an example of how to crop an image should you wish to focus on one particular area of interest:

```
%trim option's parameter order: left
\includegraphics[trim = 10mm 80mm 2
```



Figure 42

Note the presence of `clip`, as the `trim` operation will not work without it.

Trick: You can also use negative trim values to add blank space to your graphics, in cases where you need some manual alignment.

As you may have noticed, the file name of the picture is always without the extension: LaTeX will take care of getting the right version for us. Consider

the following situation: you have added some pictures to your document in JPG and you have successfully compiled it in PDF. Now you want to compile it in DVI, you run *latex* and you get a lot of errors... because you forgot to provide the EPS versions of the pictures you want to insert. At the beginning of this book, we had stated that the same LaTeX source can be compiled in both DVI and PDF without any change. This is true, as long as you don't use particular packages, and `graphicx` is one of those. In any case, you can still use both compilers with documents with pictures as well, as long as you always remember to provide the pictures in two formats (EPS and one of JPG, PNG or PDF).

11.1.5. Borders

It is possible to have LaTeX create a border around your image by using `fbox`:

```
\setlength\fboxsep{0pt}  
\setlength\fboxrule{0.5pt}  
\fbox{\includegraphics{chick}}
```


You can control the border padding with the `\setlength\fbboxsep{0pt}` command, in this case I set it to 0pt to avoid any padding, so the border will be placed tightly around the image. You can control the thickness of the border by adjusting the `\setlength\fbboxrule{0.5pt}` command.

11.1.6. Graphics storage

There is a way to tell LaTeX where to look for images: for example, it can be useful if you store images centrally for use in many different documents. The answer is in the command `\graphicspath` which you supply with an argument giving the name of an additional directory path you want searched when a file uses the `\includegraphics` command, here are some examples (trailing / is required):

```
\graphicspath{{c:\mypict~1\camera}}
\graphicspath{{c:/mypict~1/camera/}} % works well in Win XP
\graphicspath{{/var/lib/images/}}
```

```
\graphicspath{{images_folder/}{other_folder/}{third_folder/}}  
\graphicspath{{./images/}}
```

Please see <http://www.ctan.org/tex-archive/macros/latex/required/graphics/grfguide>
In the last example shown you would have a directory named "images" in the same directory as your main tex file, i.e. this is RELATIVE addressing.

As you may have noticed, in the first example I've used the "safe" (MS-DOS) form of the Windows *MyPictures* folder because it's a bad idea to use directory names containing spaces. Using absolute paths, `\graphicspath` does make your file less portable, while using relative paths (like the last example), you shouldn't have any problem with portability, but remember not to use spaces in file-names. Alternatively, if you are using PDFLaTeX, you can use the package `grffile`³⁵ which will then allow you to use spaces in file names.

34 <http://www.ctan.org/tex-archive/macros/latex/required/graphics/grfguide.pdf>.

35 <http://en.wikibooks.org/wiki/grffile>

Note that you cannot make the `graphicx` package search directories recursively. Under Linux/Unix, you can achieve a recursive search using the environment variable `TEXINPUTS`, e.g., by setting it to

```
export TEXINPUTS=./images//:./Snapshots//
```

before running `latex/pdflatex` or your TeX-IDE. (But this, of course, is not a portable method.)

11.1.7. Images as Figures

There are many scenarios where you might want to accompany an image with a caption and possibly a cross-reference. This is done using the `figure` environment. The following code sample shows the bare minimum required to use an image as a figure.

```
\begin{figure}[htb]
```

```
\includegraphics{image.png}  
\end{figure}
```

The above code extract is relatively trivial, and doesn't offer much functionality. The following code sample shows an extended use of the figure environment which is almost universally useful, offering a caption and label, centering the image and scaling it to 80% of the width of the text.

```
\begin{figure}[htb]  
\centering  
\includegraphics[width=0.8\textwidth]{image.png}  
\caption{Awesome Image}  
\label{fig:awesome_image}  
\end{figure}
```

The figure environment is not exclusively used for images. More information on the figure environment and how to use it can be found in Floats, Figures and Captions³⁶.

11.1.8. Text Wrapping around Images

Text can also be wrapped around images. (This is especially useful if you include tall pictures.)

```
%import section
\usepackage{wrapfig}

% content section
\begin{wrapfigure}{r}{8cm} % "placement and width parameter for the
width of the image space.
\centering
\includegraphics[height=80mm]{Abb/bluesnipenr.jpg}
\caption{Selbstgebaute „Bluesnipen“ um Bluetooth-Geräte aus über 1 km
Entfernung anzugreifen. (Stand: 2004)}
\label{bluesnipenr}
\end{wrapfigure}
```

Placement

There are overall eight possible positioning targets:

r	R	right side of the text
l	L	left side of the text
i	I	inside edge—near the binding (in a two-side document)
o	O	outside edge—far from the binding

The uppercase-character allows the figure to float, while the lowercase version means "exactly here".³⁷

37 <http://ftp.univie.ac.at/packages/tex/macros/latex/contrib/wrapfig/wrapfig-doc.pdf>

11.1.1.9. Including full PDF pages

There is a great package for including full pages of PDF files: `pdfpages`³⁸. It is capable of inserting full pages as is and more pages per one page in any layout (e.g. 2x3).

Package Options

```
\usepackage[ options ]{pdfpages}
```

Options:

- **final**: Inserts pages. This is the default.
- **draft**: Does not insert pages, but prints a box and the filename instead.

38 <http://www.ctan.org/tex-archive/macros/latex/contrib/pdfpages>

- **enable-survey**: Activates survey functionalities. (experimental, subject to change)

Commands

includepdf

```
\includepdf[ key=val ]{ filename }
```

Options for **key=val** (A comma separated list of options using the key = value syntax)

pages

Selects pages to insert. The argument is a comma separated list, containing page numbers (pages={3,5,6,8}), ranges of page numbers (pages={4-9}) or any combination. To insert empty pages use {}. E.g.:

```
pages={3, {}, 8-11, 15}
```

will insert page 3, an empty page, and pages 8, 9, 10, 11, and 15.

Actually not only links but all kinds of PDF annotations will get lost. Page ranges are specified by the following syntax: $m - n$.

This selects all pages from m to n . Omitting n defaults to the first

angle

You can use the angle-option for turning the included page, for example for turning a landscape document when the latex-document is portrait. E.g.:

```
angle=90
```

addtolist

Adds an entry to the list of figures, the list of tables, or any other list (e.g. from float.sty). This option requires four arguments, separated by commas:

```
addtolist={ page number , type , he
```

page number : Page number of the inserted page.

- **type**: Name of a floating environment. (figure, table, etc.)
- **heading**: Title inserted into LoF, LoT, etc.
- **label**: Name of the label. This label can be referred to with \ref and \pageref.

Like addtotoc, addtolist accepts

<p>pagecommand</p>	<p>Declares LaTeX-commands, which are executed on each sheet of paper. (Default: <i>pagecommand={thispagestyle{empty}}</i>)</p> <pre>pagecommand={\label{fig:mylabel}}</pre>
---------------------------	--

includepdfmerge

Inserts pages of several external PDF documents.

```
\includepdfmerge[ key=val ]{ file-page-list }
```

Layout

Several PDFs can be placed table-like on one page.

See more information in its documentation³⁹.

11.2. Creating Vector Graphics

11.2.1. TikZ/PGF

More thorough introduction to TikZ is available at the PGF/TikZ⁴⁰ chapter

You can draw graphics directly with TeX commands using the tikz package: <http://ftp.dante.de/tex-archive/help/Catalogue/entries/pgf.html> It comes with very good documentation with many examples.

```
% This needs \usepackage{tikz} in the preamble
\begin{figure}
```

39 <http://www.ctan.org/tex-archive/macros/latex/contrib/pdftpages/pdftpages.pdf>

40 <http://en.wikibooks.org/wiki/LaTeX%2FFPGF%2FTikZ>

```

\centering
\begin{tikzpicture}
  \draw[thick,rounded corners=8pt]
    (0,0) -- (0,2) -- (1,3.25) -- (2,2) -- (2,0) -- (0,2) -- (2,2) --
    (0,0) -- (2,0);
\end{tikzpicture}
\caption{This is the caption of my figure}
\label{fig:test}
\end{figure}

```

An extensive collection of examples can be found here:
<http://www.texample.net/tikz/>

Further packages which are based on TikZ (e.g. for drawing electrical circuits), can be found here: <http://ftp.dante.de/tex-archive/help/Catalogue/bytopic.html#pgftikzsection>

11.2.2. Xfig

Vector graphics can be created using the vector painting program Xfig (see Installation⁴¹), and exported for LaTeX. In Xfig, once your graphic is saved as a file `test.fig`, you need to export it using the **File > Export** drop down menu from the main Xfig window and then select the "Combined PS/Latex (both parts)" in the language drop down list. If you don't change any other settings, two files will be created in the same directory as the `test.fig` file, such as: `test.pstex_t` and `test.pstex`. The figure can then be placed in a LaTeX document:

```
\begin{figure}
\centering
\input{./xfig/test.pstex_t}
\caption{This is the caption of my figure}
\label{fig:test}
\end{figure}
```

11.2.3. Ipe

The Ipe⁴² extensible drawing editor is a free vector graphics editor for creating figures in PDF or EPS format. Unlike Xfig, Ipe represents LaTeX⁴³ fonts in their correct size on the screen which makes it easier to place text labels at the right spot. Ipe also has various snapping modes (for example, snapping to points, lines, or intersections) that can be used for geometric constructions.

11.2.4. Inkscape

Another program for creating vector graphics is Inkscape⁴⁴. It works with Scalable Vector Graphics (SVG)⁴⁵ files, although it can export to many formats

42 <http://en.wikipedia.org/wiki/Ipe%20%28program%29>

43 <http://en.wikibooks.org/wiki/LaTeX>

44 <http://www.inkscape.org/>

45 <http://www.w3.org/Graphics/SVG/>

that can be included in LaTeX⁴⁶ files, such as EPS and PDF. From version 0.48, there is a combined PDF/EPS/PS+LaTeX output option, like XFig has.

11.2.5. Editing EPS graphics

As described above, graphics content can be imported into LaTeX⁴⁷ from outside programs as EPS files. But sometimes you want to edit or retouch these graphics files. An EPS file can be edited with any text editor since it is formatted as ASCII. In a text editor, you can achieve simple operations like replacing strings or moving items slightly, but anything further becomes cumbersome.

To properly edit an EPS file, you can convert it to an *editable* format using `pstoedit`⁴⁸. For instance, to get an Xfig-editable file, do:

46 <http://en.wikibooks.org/wiki/LaTeX>

47 <http://en.wikibooks.org/wiki/LaTeX>

48 <http://www.pstoedit.net/>

```
$ pstoeedit -f fig input.eps output.fig
```

And to get an SVG file for Inkscape you can do:

```
$ pstoeedit -f plot-svg input.eps output.svg
```

Sometimes pstoeedit fails to create the target format (for example when the EPS file contains clipping information). A more robust way to edit EPS files is achieved by converting it first to PDF and then importing the resulting PDF in Inkscape. Inkscape uses the Cairo library that achieves a high-quality transformation of the original EPS figure:

```
$ epstopdf input.eps  
$ inkscape input.pdf
```

When all of the above fails, one can simplify the EPS file before attempting other conversions, by using the `eps2eps`⁴⁹ tool (also see next section):

49 http://linuxcommand.org/man_pages/eps2eps1.html

```
$ eps2eps input.eps input-e2.eps
```

This will convert all the fonts to pre-drawn images, which is sometimes desirable when submitting manuscripts for publication. However, on the downside, the fonts are NOT converted to lines, but instead to bitmaps, which reduces the quality of the fonts.

11.2.6. Converting a color EPS to grayscale

Sometimes color EPS figures need to be converted to black-and-white or grayscale to meet publication requirements. This can be achieved with the `eps2eps`⁵⁰ of the Ghostscript⁵¹ package and <http://www.pa.op.dlr.de/Patrick-Joeckel/pscol/index.html>⁵² programs:

50 http://linuxcommand.org/man_pages/eps2eps1.html

51 <http://ghostscript.com/>

52 <http://www.pa.op.dlr.de/~PatrickJoeckel/pscol/index.html>

```
$ eps2eps input.eps input-e2.eps  
$ pscoll -0gray input-e2.eps input-gray.eps
```

11.3. References

12. Creating Graphics

In the Importing Graphics¹ chapter, you learned that you can import or link graphics into LaTeX, such as graphics that you have created in another program or obtained elsewhere. In this chapter, you will learn how to create or embed graphics directly in a LaTeX document. The graphics is marked up using commands similar to those for typesetting bold text or creating mathematical formulas, as the following example of embedded graphics shows:

```
\begin{displaymath}
\ymatrix{ \bullet \ar[r] \ar@{
}\end{displaymath}
```



Figure 43

There are several packages supporting the creation of graphics directly in LaTeX, including `picture`², `xy-pic`³ and `PGF/TikZ`⁴, described in the following sections.

2 <http://en.wikibooks.org/wiki/LaTeX%2FPicture>

3 <http://en.wikibooks.org/wiki/LaTeX%2FXy-pic>

4 <http://en.wikibooks.org/wiki/LaTeX%2FPGF%2FTikZ>

12.1. Overview

The `picture` environment allows programming pictures directly in LaTeX. On the one hand, there are rather severe constraints, as the slopes of line segments as well as the radii of circles are restricted to a narrow choice of values. On the other hand, the `picture` environment of LaTeX2e brings with it the `\qbezier` command, "q" meaning *quadratic*. Many frequently-used curves such as circles, ellipses, and catenaries⁵ can be satisfactorily approximated by quadratic Bézier curves, although this may require some mathematical toil. If a programming language like Java is used to generate `\qbezier` blocks of LaTeX input files, the `picture` environment becomes quite powerful.

Although programming pictures directly in LaTeX is severely restricted, and often rather tiresome, there are still reasons for doing so. The documents thus

5 <http://en.wikipedia.org/wiki/catenary>

produced are "small" with respect to bytes, and there are no additional graphics files to be dragged along.

Packages like `epic`, `eepic` or `pstricks` enhance the original picture environment, and greatly strengthen the graphical power of LaTeX.

While the former two packages just enhance the picture environment, the `pstricks` package has its own drawing environment, `pspicture`. The power of `pstricks` stems from the fact that this package makes extensive use of PostScript possibilities. Unfortunately it has one big shortcoming: it doesn't work together with pdfLaTeX, as such to generate a PDF document form TeX source you have to go `TeX`→`DVI`→`PDF`; losing hyperlinks, metadata and microtypographic features of `pdflatex`.

In addition, numerous packages have been written for specific purposes. One of them is *XY-pic*, described at the end of this chapter. A wide variety of these packages are described in detail in *The LaTeX Graphics Companion* (not to be confused with *The LaTeX Companion*).

Perhaps the most powerful graphical tool related with LaTeX is MetaPost⁶, the twin of Donald E. Knuth's METAFONT⁷. MetaPost has the very powerful and mathematically sophisticated programming language of METAFONT. Contrary to METAFONT, which generates bitmaps, MetaPost generates encapsulated PostScript files, which can be imported in LaTeX. For an introduction, see *A User's Manual for MetaPost*. A very thorough discussion of LaTeX and TEX strategies for graphics (and fonts) can be found in *TEX Unbound*.

The last but certainly not least is the PGF/TikZ system. While the previous systems (`picture`, `epic`, `pstricks` or `metapost`) focus on the *how* to draw, TikZ focuses more on the *what* to draw. One could say that TikZ is to drawing in LaTeX as LaTeX is to digital typesetting. It's recommended to use it if your LaTeX distribution includes it.

6 <http://en.wikipedia.org/wiki/MetaPost>

7 <http://en.wikipedia.org/wiki/METAFONT>

12.2. Alternatives

In many cases, especially for more advanced diagrams, it may be easier to draw the graphics using external vector graphics software, and then import the file into the document (see `./Importing Graphics`⁸). However most software does not support LaTeX fonts or mathematical notation, which can result in not suitable and inconsistent graphics. There are several solutions to this problem.

The easiest solution is to use the `picture` environment and then simply use the `"put"` command to put a graphics file inside the `picture`, along with any other desired LaTeX element. For example:

```

\setlength{\unitlength}{0.8cm}
\begin{picture}(6,5)
\put(3.5,0.4){$\displaystyle
s:=\frac{a+b+c}{2}$}
\put(1,1){\includegraphics[
width=2cm,height=2cm]{picture.eps},
\end{picture}

```

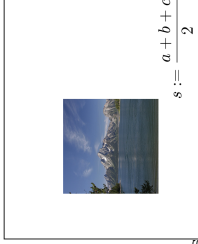


Figure 44

Note that the border around the picture in the above example was added by using `\fbox9`, so the contents of the border is the picture as generated by the above code.

Another solution is to use `textext10`, a plug-in for Inkscape¹¹ which allows one to insert small LaTeX objects into .SVG images. These images can then

9 Chapter 11.1.5 on page 365

10 <http://pav.iki.fi/software/textext/>

11 <http://www.inkscape.org/>

be saved as .EPS (or .PDF) files which may then be imported into the LaTeX document proper.

Yet another solution is provided by `lpic`¹², which allows TeX annotations to imported graphics.

12.2.1. Using `xfig` to create pictures

An option that allows significantly more flexibility while creating graphics that are consistent with LaTeX is to use `xfig`¹³. `xfig` is a drawing program, which allows exports into various formats from which it's possible to import into LaTeX. While the software is designed for Linux, it can be run using Macports¹⁴ for Macs and under any X-Window System for Windows.

12 <http://www.math.uni-leipzig.de/~matveyev/lpic/>

13 <http://www.xfig.org>

14 <http://www.macports.org>

There are many ways to use `xfig` to create graphics for LaTeX documents. One method is to export the drawing as a LaTeX document. This method, however, suffers from various drawbacks: lines can be drawn only at angles that are multiples of 30 and 45 degrees, lines with arrows can only be drawn at angles that are multiples of 45 degrees, several curves are not supported, etc.

Exporting a file as PDF/LaTeX or PS/LaTeX, on the other hand, offers a good deal more flexibility in drawing. Here's how it's done:

1. Create the drawing in `xfig`. Wherever you need LaTeX text, such as a mathematical formula, enter a LaTeX string in a textbox. Use the Edit tool to open the properties of each of those textboxes, and change the option on the "Special Flag" field to Special. This tells LaTeX to interpret these textboxes when it opens the figure.
2. Go to File -> Export and export the file as PDF/LaTeX (both parts) or PS/LaTeX (both parts), depending on whether you are using `pdflatex` or `pslatex` to compile your file. In your LaTeX document, where the picture should be, use the following, where "test" is replaced by the name of the image:

```
\begin{figure}[htbp]
\centering
\input{test.pdf_t}
\caption{Your figure}
\label{figure:example}
\end{figure}
```

Observe that this is just like including a picture, except that rather than using `\includegraphics`, we use `\input`. If the export was into PS/LaTeX, the file extension to include would be `.pstex_t` instead of `.pdf_t`.

3. Make sure to include packages `graphicx` and `color` in the file, with the `usepackage` command right below the `documentclass` command, like this:

```
\usepackage{graphicx}
\usepackage{color}
```

And you're done!

For more details on using xfig with LaTeX, this chapter¹⁵ of the xfig User Manual¹⁶ may prove helpful.

15 http://www-epb.lbl.gov/xfig/latex_and_xfig.html

16 <http://www-epb.lbl.gov/xfig/contents.html>

13. Floats, Figures and Captions

The previous¹ chapter introduced importing graphics. However, just having a `picture` stuck in between paragraphs does not look professional. For starters, we want a way of adding captions, and to be able to cross-reference. What we need is a way of defining *figures*. It would also be good if LaTeX could apply principles similar to when it arranges text to look its best to arranging pictures as well. This is where *floats* come into play.

13.1. Floats

Floats are containers for things in a document that cannot be broken over a page. LaTeX by default recognizes "table" and "figure" floats, but you can define new ones of your own (see Custom Floats² below). Floats are there to deal with the problem of the object that won't fit on the present page, and to help when you really don't want the object here just now.

Floats are not part of the normal stream of text, but separate entities, positioned in a part of the page to themselves (top, middle, bottom, left, right, or wherever the designer specifies). They always have a caption describing them and they are always numbered so they can be referred to from elsewhere in the text. LaTeX automatically floats Tables and Figures, depending on how much space is left on the page at the point that they are processed. If there is not enough room on the current page, the float is moved to the top of the next page. This can be changed by moving the Table or Figure definition to an earlier or

2 Chapter 13.2.8 on page 426

later point in the text, or by adjusting some of the parameters which control automatic floating.

Authors sometimes have many floats occurring in rapid succession, which raises the problem of how they are supposed to fit on the page and still leave room for text. In this case, LaTeX stacks them all up and prints them together if possible, or leaves them to the end of the chapter in protest. The skill is to space them out within your text so that they intrude neither on the thread of your argument or discussion, nor on the visual balance of the typeset pages.

13.1.1. Figures

To create a figure that floats, use the `figure` environment.

```
\begin{figure}[placement specifier]
... figure contents ...
\end{figure}
```

The previous section mentioned how floats are used to allow LaTeX to handle figures, while maintaining the best possible presentation. However, there may be times when you disagree, and a typical example is with its positioning of figures. The *placement specifier* parameter exists as a compromise, and its purpose is to give the author a greater degree of control over where certain floats are placed.

Specifier	Permission
h	Place the float <i>here</i> , i.e., <i>approximately</i> at the same point it occurs in the source text (however, not <i>exactly</i> at the spot)
t	Position at the <i>top</i> of the page.
b	Position at the <i>bottom</i> of the page.
p	Put on a <i>special page</i> for floats only.

Specifier	Permission
!	Override internal parameters LaTeX uses for determining "good" float positions.
H	Places the float at precisely the location in the LaTeX code. Requires the <code>float</code> package, ³ e.g., <code>\usepackage{float}</code> . This is somewhat equivalent to <code>h!</code> .

What you do with these *placement permissions* is to list which of the options you wish to make available to LaTeX. These are simply possibilities, and LaTeX will decide when typesetting your document which of your supplied specifiers it thinks is best.

3 <http://www.ctan.org/tex-archive/macros/latex/contrib/float/>

Use `\listoffigures` to add a list of the figures in the beginning of the document. To change the name used in the caption from **Figure** to **Example**, use `\renewcommand{\figurename}{Example}` in the figure contents.

13.1.2. Figures with borders

It's possible to get a thin border around all figures. You have to write the following once at the beginning of the document:

```
\usepackage{float}  
\floatstyle{boxed}  
\restylefloat{figure}
```

The border will not include the caption.

13.1.3. Tables

Although tables have already been covered⁴, it was only the internal syntax that was discussed. The `tabular` environment that was used to construct the tables is not a float by default. Therefore, for tables you wish to float, wrap the `tabular` environment within a `table` environment, like this:

```
\begin{table}
  \begin{tabular}{...}
    ... table data ...
  \end{tabular}
\end{table}
```

You may feel that it is a bit long winded, but such distinctions are necessary, because you may not want all tables to be treated as a float.

Use `\listoftables` to add a list of the tables in the beginning of the document.

13.1.4. Keeping floats in their place

The `placeins`⁵ package provides the command `\FloatBarrier`, which can be used to prevent floats from being moved over it. This can, e.g., be useful at the beginning of each section. The package even provides an option to change the definition of `\section` to automatically include a `\FloatBarrier`. This can be set by loading the package with the option `[section]` (`\usepackage[section]{placeins}`). `\FloatBarrier` may also be useful to prevent floats intruding on lists created using `itemize` or `enumerate`. The `flafter` package can be used to force floats to appear after they are defined, and the `endfloat`⁶ package can be used to place all floats at the end of a document.

⁵ <http://ctan.org/tex-archive/macros/latex/contrib/placeins>

⁶ <http://www.ctan.org/pkg/endfloat>

13.2. Captions

It is always good practice to add a caption to any figure or table. Fortunately, this is very simple in LaTeX. All you need to do is use the `\caption{text}` command within the float environment. Because of how LaTeX deals sensibly with logical structure, it will automatically keep track of the numbering of figures, so you do not need to include this within the caption text.

The location of the caption is traditionally underneath the float. However, it is up to you to therefore insert the caption command after the actual contents of the float (but still within the environment). If you place it before, then the caption will appear above the float. Try out the following example to demonstrate this effect:

```

\documentclass[a4paper,12pt]{article}

\usepackage[english]{babel}
\usepackage{graphicx}

\begin{document}

\begin{figure}[h!]
\caption{A picture of a gull.}
\centering
\includegraphics[width=0.5\textwidth]{gull}
\end{figure}

\begin{figure}[h!]
\centering
\reflectbox{%
\includegraphics[width=0.5\textwidth]{gull}}
\caption{A picture of the same gull
looking the other way!}
\end{figure}

\begin{table}[h!]
\begin{center}

```

Figure 1: A picture of a gull.



Figure 2: A picture of the same gull looking the other way!

1	2	3
4	5	6
7	8	9

Table 1: A simple table

Notice how the tables and figures have independent counters.

Figure 45

note that the command `\reflectbox{...}` flips its content horizontally.

13.2.1. Lists of figures and tables

Captions can be listed at the beginning of a paper or report in a "List of Tables" or a "List of Figures" section by using the `\listoftables` or `\listoffigures` commands, respectively. The caption used for each figure will appear in these lists, along with the figure numbers, and page numbers that they appear on.

The `\caption` command also has an optional parameter, `\caption[short]{long}` which is used for the *List of Tables* or *List of Figures*. Typically the short description is for the caption listing, and the long description will be placed beside the figure or table. This is particularly useful if the caption is long, and only a "one-liner" is desired in the figure/table listing. Here is an example of this usage:

List of Figures

1 Close up of *Hemidactylus* sp. 1

```
\documentclass[12pt]{article}
\usepackage{graphicx}
```

1 Introduction

```
\begin{document}

\listoffigures

\section{Introduction}
```

```
\begin{figure}[hb]
```

```
\centering
```

```
\includegraphics[width=4in]{gecko}
\caption[Close up of Hemidactylus sp.]
```

```
{Close up of Hemidactylus sp., which is
```

```
part the genus of the gecko family. It is the
```

```
second most speciose genus in the family.}
```

```
\end{figure}
```

```
\end{document}
```



Figure 1: Close up of *Hemidactylus* sp., which is part the genus of the gecko family. It is the second most speciose genus in the family.

Figure 46

13.2.2. Side captions

It is sometimes desirable to have a caption appear on the side of a float, rather than above or below. The `sidecap` package can be used to place a caption beside a figure or table. The following example demonstrates this for a figure by using a `SCfigure` environment in place of the `figure` environment.

```

\documentclass{article}

\usepackage[pdftex]{graphicx}
\usepackage{sidecap}

\begin{document}

\begin{SCfigure}
\centering
\includegraphics[width=0.5\textwidth]{Giraff_picture}% picture filename
\caption{... caption text ...}
\end{SCfigure}

\end{document}

```



Figure 1: The giraffe (*Giraffa camelopardalis*) is an African even-toed ungulate mammal, the tallest of all land-living animal species. Males can be 4.8 to 5.5 metres tall and weigh up to 1,360 kilograms. The record-sized bull was 5.87 m tall and weighed approximately 2,000 kg. Females are generally slightly shorter and weigh less than the males do.

Figure 47

13.2.3. Labels and Cross-referencing

Labels and cross-references work fairly similarly to the general case - see the Labels and Cross-referencing⁷ section for more information.

Warning: If you want to label a figure so that you can reference it later, you have to add the label **after the caption** (inside seems to work in LaTeX 2e) but **inside the floating environment**. If it is declared outside, it will give the section number. If the label picks up the section or list number instead of the figure number, put the label inside the caption to ensure correct numbering. If you get an error when the label is inside the caption, use `\protect` in front of the `\label` command.

13.2.4. Wrapping text around figures

Although not normally the case in academic writing, an author may prefer that some floats do not break the flow of text, but instead allow text to wrap around it. (Obviously, this effect only looks decent when the figure in question is significantly narrower than the text width.)

A word of warning: Wrapping figures in LaTeX will require a lot of manual adjustment of your document. There are several packages available for the task, but none of them work perfectly. Before you make the choice of including figures with text wrapping in your document, make sure you have considered all the options. For example, you could use a layout with two columns for your documents and have no text-wrapping at all.

Anyway, we will look at the package `wrapfig`. (Note: `wrapfig` may not come with the default installation of LaTeX; you might need to install additional packages manually.⁸)

To use `wrapfig`, you must first add this to the preamble:

```
\usepackage{wrapfig}
```

This then gives you access to:

```
\begin{wrapfigure}[lineheight]{alignment}{width}
```

Alignment can normally be either *l* for left, or *r* for right. Lowercase *l* or *r* forces the figure to start precisely where specified (and may cause it to run over page breaks), while capital *L* or *R* allows the figure to float. If you defined your document as twosided, the alignment can also be *i* for inside or *o* for outside, as well as *I* or *O*. The *width* is, of course, the width of the figure. An example:

8 http://en.wikibooks.org/wiki/LaTeX/Packages/Installing_Extra_Packages

Gulls are birds in the family Laridae. They are most closely related to the terns (family Sternidae), auks and skimmers, and more distantly to the waders. Most gulls belong to the large genus *Larus*.

They are in general medium to large birds, typically grey or white, often with black markings on the head or wings. They have stout, longish bills and webbed feet.

Most gulls, particularly *Larus* species, are ground nesting carnivores, which will take live food or scavenge opportunistically. The live food often includes crabs and small fish. Apart from the kittiwakes, gulls are typically coastal or inland species, rarely venturing far out to sea. The large species take up to four years to attain full adult plumage, but two years is typical for small gulls.

Gulls the larger species in particular are resourceful and highly-intelligent birds, demonstrating complex methods of communication and a highly-developed social structure. Certain species (e.g. the Herring Gull) have exhibited tool

```
\begin{wrapfigure}{r}{0.5\textwidth}
\begin{center}
\includegraphics[width=0.48\textwidth]{center}
\end{center}
\caption{A gull}
\end{wrapfigure}
```



Figure 1: A gull

Figure 48

Note that we have specified a size for both the `wrapfigure` environment and the image we have included. We did it in terms of the text width: it is always better to use relative sizes in LaTeX, let LaTeX do the work for you! The "wrap" is slightly bigger than the picture, so the compiler will not return any strange warning and you will have a small white frame between the image

and the surrounding text. You can change it to get a better result, but if you don't keep the image smaller than the "wrap", you will see the image *over* the text.

The wrapfig package can also be used with user-defined floats with float package. See below in the section on custom floats⁹.

13.2.5. Tip for figures with too much white space

It happens that you'll generate figures with too much (or too little) white space on the top or bottom. In such a case, you can simply make use of the optional argument `[lineheight]`. It specifies the height of the figure in number of lines of text. Also remember that the environment `center` adds some extra white space at its top and bottom; consider using the command `\centering` instead.

Another possibility is adding space within the float using the `\vspace{...}` command. The argument is the size of the space you want to add, you can use any unit you want, including pt, mm, in, etc. If you provide a negative argument, it will add a *negative* space, thus removing some white space. Using `\vspace` tends to move the caption relative to the float while the `[lineheight]` argument does not. Here is an example using the `\vspace` command, the code is exactly the one of the previous case, we just added some negative vertical spaces to shrink everything up:

Gulls are birds in the family Laridae. They are most closely related to the terns (family Sternidae), auks and skimmers, and more distantly to the waders. Most gulls belong to the large genus *Larus*.

They are in general medium to large birds, typically grey or white, often with black markings on the head or wings. They have stout, longish bills and webbed feet.

Most gulls, particularly *Larus* species, are ground nesting carnivores, which will take live food or scavenge opportunistically. The live food often includes crabs and small fish. Apart from the kittiwakes, gulls are typically coastal or inland species, rarely venturing far out to sea. The large species take up to four years to attain full adult plumage, but two years is typical for small gulls.

Gulls the larger species in particular are resourceful and highly-intelligent birds, demonstrating complex methods of communication and a highly-developed social structure. Certain species (e.g. the Herring Gull) have exhibited tool use behaviour. Many species of gull have learned to co-exist successfully with man and have thrived in human habitats. Others rely on kleptoparasitism to get their food.

```
\begin{wrapfigure}{r}{0.5\textwidth}
\vspace{-20pt}
\begin{center}
\includegraphics[width=0.48\textwidth]{center}
\end{center}
\vspace{-20pt}
\caption{A gull}
\vspace{-10pt}
\end{wrapfigure}
```

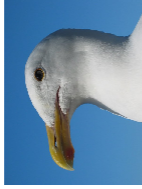


Figure 1: A gull

Figure 49 336

In this case it may look too shrunk, but you can manage spaces the way you like. In general, it is best not to add any space at all: let LaTeX do the formatting work!

(In this case, the problem is the use of `\begin{center}` to center the image. The `center` environment adds extra space that can be avoided if `\centering` is used instead.)

Alternatively you might use the `picins` package instead of the `wrapfigure` package which produces a correct version without the excess white space out of the box without any hand tuning.

There is also an alternative to `wrapfig`: the package `floatflt` <http://tug.ctan.org/tex-archive/macros/latex/contrib/floatflt/10>
- for documentation see [http://www.ctan.org/tex-](http://www.ctan.org/tex-archive/macros/latex/contrib/floatflt/floatflt.pdf)

10 <http://tug.ctan.org/tex-archive/macros/latex/contrib/floatflt/>

11 <http://www.ctan.org/tex-archive/macros/latex/contrib/floatflt/floatflt.pdf>

13.2.6. Subfloats

A useful extension is the `subcaption` package [<ftp://ftp.dante.de/tex-archive/macros/latex/contrib/caption/subcaption.pdf>] (the `subfigure` and `subfig` packages are deprecated and shouldn't be used any more), which uses subfloats within a single float. This gives the author the ability to have subfigures within figures, or subtables within table floats. Subfloats have their own caption, and an optional global caption. An example will best illustrate the usage of this package:

```
\usepackage{graphicx}
\usepackage{caption}
\usepackage{subcaption}

\begin{figure}
  \begin{subfigure}[b]{0.3\textwidth}
    \centering
    \includegraphics[width=\textwidth]{gull}
    \caption{A gull}
    \label{fig:gull}
  \end{subfigure}
  \end{subfigure}
%
```



```
~ %add desired spacing between images, e. g. ~, \quad, \qqquad
etc.
    % (or a blank line to force the subfigure onto a new line)
\begin{subfigure}[b]{0.3\textwidth}
\centering
\includegraphics[width=\textwidth]{tiger}
\caption{A tiger}
\label{fig:tiger}
\end{subfigure}
~ %add desired spacing between images, e. g. ~, \quad, \qqquad
```

etc.

```
    % (or a blank line to force the subfigure onto a new line)
\begin{subfigure}[b]{0.3\textwidth}
\centering
\includegraphics[width=\textwidth]{mouse}
\caption{A mouse}
\label{fig:mouse}
\end{subfigure}
\caption{Pictures of animals}\label{fig:animals}
\end{figure}
```



(a) A gull



(b) A tiger



(c) A mouse

Figure 1: Pictures of animals

Figure 50

You will notice that the figure environment is set up as usual. You may also use a table environment for subtables. For each subfloat, you need to use:

```
\begin{subfigure}[<placement specifier>]{<width>}  
  \centering  
  ... figure or table ...
```

```
\caption{<sub caption>}  
\end{subfigure}
```

If you intend to cross-reference any of the subfloats, see where the label is inserted; `\caption` outside the `subfigure`-environment will provide the global caption.

`subcaption` will arrange the figures or tables side-by-side providing they can fit, otherwise, it will automatically shift subfloats below. This effect can be added manually, by putting the newline command (`\`) before the figure you wish to move to a newline.

Horizontal spaces between figures are controlled by one of several commands, which are placed in between `\begin{subfigure}` and `\end{subfigure}`:

- A non-breaking space (specified by `~` as in the example above) can be used to insert a space in between the subfigs.

- Math spaces¹²: `\qquad`, `\quad`, `\;` , and `\,`
- Generic space: `\hspace{length}`
- Automatically expanding/contracting space: `\hfill`

13.2.7. Wide figures in two column documents

If you are writing a document using two columns (i.e. you started your document with something like `\documentclass[twocolumn]{article}`), you might have noticed that you can't use floating elements that are wider than the width of a column (using a LaTeX notation, wider than `0.5\textwidth`), otherwise you will see the image overlapping with text. If you really have to use such wide elements, the only solution is to use the "starred" variants of the floating environments, that are `{figure*}` and `{table*}`. Those "starred" versions work exactly like the standard ones, but they will be as wide as the page, so you will get no overlapping.

12 Chapter 17.14 on page 562

A bad point of those environments is that they can be placed only at the top of the page or on their own page. If you try to specify their position using modifiers like *b* or *h* they will be ignored. Add `\usepackage{dblfloatfix}` to the preamble in order to alleviate this problem with regard to placing these floats at the bottom of a page, using the optional specifier `[b]`. Default is `[tbp]`. However, *h* still does not work.

To prevent the figures from being placed out-of-order with respect to their "non-starred" counterparts, the package `fixltx2e`¹³ should be used (e.g. `\usepackage{fixltx2e}`).

13.2.8. Custom Floats

If tables and figures are not adequate for your needs, then you always have the option to create your own! Examples of such instances could be source code

¹³ <http://www.tex.ac.uk/cgi-bin/texfaq2html?label=2colfloatorder>

examples, or maps. For a program float example, one might therefore wish to create a float named `program`. The package `float` is your friend for this task. *All commands to set up the new float must be placed in the preamble, and not within the document.*

1. Add `\usepackage{float}` to the preamble of your document
2. Declare your new float using: `\newfloat{type}{placement}{examplecounter}`, where:
 - *type* - the new name you wish to call your float, in this instance, 'program'.
 - *placement* - t, b, p, or h (as previously described in Placement¹⁴), where letters enumerate permitted placements.
 - *ext* - the file name extension of an auxiliary file for the list of figures (or whatever). Latex writes the captions to this file.
 - *outer counter* - the presence of this parameter indicates that the counter associated with this new float should depend on outer counter, for example 'chapter'.

3. The default name that appears at the start of the caption is the type. If you wish to alter this, use `\floatname{type}{floatname}`
4. Changing float style can be issued with `\floatstyle{style}` (Works on all subsequent `\newfloat` commands, therefore, must be inserted before `\newfloat` to be effective).
 - `plain` - the normal style for Latex floats, but the caption is always below the content.
 - `plaintop` - the normal style for Latex floats, but the caption is always above the content.
 - `boxed` - a box is drawn that surrounds the float, and the caption is printed below.
 - `ruled` - the caption appears above the float, with rules immediately above and below. Then the float contents, followed by a final horizontal rule.

Float styles can also be customized as the second example below illustrates.

An example document using a new program float type:

```
\documentclass{article}

\usepackage{float}

\floatstyle{ruled}
\newfloat{program}{thp}{lop}
\floatname{program}{Program}

\begin{document}

\begin{program}
\begin{verbatim}

class HelloWorldApp {
public static void main(String[] args) {
//Display the string
System.out.println("Hello World!");
}
}
\end{verbatim}
\caption{The Hello World! program in Java.}
\end{program}

\end{document}
```


The `verbatim` environment is an environment that is already part of LaTeX. Although not introduced so far, its name is fairly intuitive! LaTeX will reproduce everything you give it, including new lines, spaces, etc. It is good for source code, but if you want to introduce a lot of code you might consider using the `listings`¹⁵ package, that was made just for it.

While this is useful, one should be careful when embedding the float within another float. In particular, the error `not in outer par mode` may occur. One solution might be to use the `[H]` option (not any other) on the inner float, as this option "pins" the inner float to the outer one.

Newly created floats with *newfloat* can also be used in combination with the *wrapfig* package from above. E.g. the following code creates a floating text box, which floats in the text on the right side of the page and is complete with caption, numbering, an index file with the extension `.lob` and a customization of the float's visual layout:

15 <http://en.wikibooks.org/wiki/LaTeX%2FPackages%2FListings>

```

\documentclass{article}

% have hyperref package before float in order to get strange errors
with .\theHfloatbox
\usepackage[pdfTeX]{hyperref}

\usepackage{float}

% allows use of "@" before \begin{document}
\makeatletter

% this creates a custom and simpler ruled box style
\newcommand\floatc@simplerule[2]{\@fs@cfont #1 #2}\par}
\newcommand\if
s@simplerule{\def\@fs@cfont{\bfseries}\let\@fs@capt\floatc@simplerule
\def\@fs@pre{\hrule height.8pt depth0pt \kern4pt}%
\def\@fs@post{\kern4pt\hrule height.8pt depth0pt \kern4pt \relax}%
\def\@fs@mid{\kern8pt}%
\let\@fs@iftopcapt\iftrue}

% this code block defines the new and custom floatbox float
environment
\floatstyle{simplerule}
\newfloat{floatbox}{thp}{lob}[section]

```

```
\floatname{floatbox}{Text Box}

\begin{document}

\begin{floatbox}{r}{
\textit{Bootstrapping} is a resampling technique used
for robustly estimating statistical quantities, such as
the model fit  $R^2$ . It offers some protection against
the sampling bias.
\caption{Bootstrapping}
}
\end{floatbox}

\end{document}
```

13.2.9. Caption Styles

To change the appearance of captions, use the `\caption` command from the `caption` package.¹⁶ For example, to make all caption labels small and bold:

```
\usepackage[font=small,labelfont=bf]{caption}
```

The KOMA script packages <http://www.komascript.de/>¹⁷ have their own caption customizing features with e.g. `\captionabove`, `\captionformat` and `\setcapwidth`. However these definitions have limited effect on newly created float environments with the `wrapfig` package.

Alternatively, you can redefine the `\thefigure` command:

¹⁶ <http://mirror.ctan.org/macros/latex/contrib/caption/caption-eng.pdf>

¹⁷ <http://www.komascript.de/>

```
\renewcommand{\thefigure}{\arabic{section}.\arabic{figure}}
```

See this page¹⁸ for more information on counters.

13.3. Labels in the figures

There is a LaTeX package `lpic`¹⁹ to put LaTeX on top of included graphics, thus allowing to add TeX annotations to imported graphics. It defines a convenient interface to put TeX over included graphics, and allows for drawing a white background under the typeset material to overshadow the graphics. It is a better alternative for labels inside of graphics; you do not have to change text size when rescaling pictures, and all LaTeX power is available for labels.

18 Chapter 6.1.6 on page 203

19 <http://www.math.uni-leipzig.de/~matveyev/lpic/>

A very similar package, with somewhat different syntax, is `pinlabel`²⁰. The link given also points to the packages `psfrag` and `overpic`.

A much more complicated package which can be used in the same way is `tikz`²¹. `tikz` is a front-end to a drawing library called `pgf` (which is used to make beamer). It can be used to label figures by adding text nodes on top of an image node.

13.3.1. Summary

That concludes all the fundamentals of floats. You will hopefully see how much easier it is to let LaTeX do all the hard work and tweak the page layouts in order to get your figures in the best place. As always, the fact that LaTeX takes care of all caption and reference numbering is a great time saver.

²⁰ <http://www.ctan.org/tex-archive/help/Catalogue/entries/pinlabel.html>

²¹ <http://www.ctan.org/pkg/pgf>

13.4. References

14. Hyperlinks

LaTeX enables typesetting of hyperlinks, useful when the resulting format is PDF, and the hyperlinks can be followed. It does so using the package *hyperref*.

14.1. Hyperref

The package `hyperref` provides LaTeX the ability to create hyperlinks within the document. It works with *pdflatex* and also with standard "latex" used with `dvips` and `ghostscript` or `dvipdfm` to build a PDF file. If you load it,

you will have the possibility to include interactive external links and all your internal references will be turned to hyperlinks. The compiler *pdf_{flat}ex* makes it possible to create PDF files directly from the LaTeX source, and PDF supports more features than DVI. In particular PDF supports hyperlinks, and the only way to introduce them in LaTeX is using `hyperref`. Moreover, PDF can contain other information about a document such as the title, the author, etc., and you can edit those using this same package.

14.2. Usage

The basic usage with the standard settings is straightforward. Just load the package in the preamble, at the end of **all** the other packages but prior to other settings:

```
\usepackage{hyperref}
```

This will automatically turn all your internal references into hyperlinks. It won't affect the way to write your documents: just keep on using the standard

`\label`

-

`\ref`

system (discussed in the chapter on Labels and Cross-referencing¹); with `hyperref` those "connections" will become links and you will be able to click on them to be redirected to the right page. Moreover the table of contents, list of figures/tables and index will be made of hyperlinks, too.

14.2.1. Commands

The package provides three useful commands for inserting links pointing outside the document.

\hyperref

Usage:

```
\hyperref[label_name]{'link text'}
```

This will have the same effect as

```
\ref{label_name}
```

but will make the text *link text* a full link, instead. The two can be combined, for example in

```
we use \hyperref[mainlemma]{lemma \ref{*{mainlemma}}}
```

Note the * after

```
\ref
```

for avoiding nested hyperlinks.

If the lemma labelled as "mainlemma" was number 4.1.1, then the outputted text would be "we use lemma 4.1.1" with the hyperlink as expected.

`\url`

Usage:

```
\url{'my_url'}
```

It will show the URL using a mono-spaced font and, if you click on it, your browser will be opened pointing at it.

`\href`

Usage:

```
\href{'my_url'}{'description'}
```

It will show the string "description" using standard document font but, if you click on it, your browser will be opened pointing at "my_url". Here is an example:

```
\url{http://www.wikibooks.org}
\href{http://www.wikibooks.org}{wikibooks home}
```

Both point at the same page, but in the first case the URL will be shown, while in the second case the URL will be hidden. Note that, if you print your document, the link stored using

```
\href
```

will not be shown anywhere in the document.

14.2.2. Other possibilities

Apart from linking to websites discussed above, `\hyperref` can be used to provide `\mailto` links, links to local files, and links to anywhere within the PDF output file.

Mail address

A possible way to insert email links is by

```
\href{mailto:my_address@wikibooks.org}{my_address@wikibooks.org}
```

it just shows your email address (so people can know it even if the document is printed on paper) but, if the reader clicks on it, (s)he can easily send you an email. Or, to incorporate the `url` package's formatting and line breaking abilities into the displayed text, use²

```
\href{mailto:my_address@wikibooks.org}{\nolinkurl{my_address@wikibooks.org}}
```

When using this form, note that the

```
\nolinkurl
```

command is fragile and if the hyperlink is inside of a moving argument, it must be preceded by a

² Email link with `hyperref`, `url` packages ³ . . Retrieved

`\protect`

command.

Local file

Files can also be linked using the `url` or the `href` commands. You simply have to add the string `run` : at the beginning of the link string:

```
\url{run:/path/to/my/file.ext}
\href{run:/path/to/my/file.ext}{text displayed}
```

You can use relative paths to link documents near the location of your current document; in order to do so, use the standard Unix-like notation (`.` / is the current directory, `..` / is the previous directory, etc.)

14.2.3. Hyperlink and Hyeptarget

It is also possible to create an anchor anywhere in the document (with or without caption) and to link to it, with:

```
\hyperlink{label}{anchor caption}
```

and

```
\hypertarget{label}{link caption}
```

14.3. Customization

The standard settings should be fine for most users, but if you want to change something, you can easily do it. There are several variables you can change and there are two methods to pass those to the package. You can pass the options as an argument of the package when you load it (that's the standard way packages work), or you can use the `\hypersetup` command:


```
\hypersetup{'option1' [, ...]}
```

you can pass as many options as you want; separate them with a comma. Options have to be in the form:

```
variable_name=new_value
```

exactly the same format has to be used if you pass those options to the package while loading it, like this:

```
\usepackage['option1, option2']{hyperref}
```

Here is a list of the possible variables you can change (for the complete list, see the official documentation). The default values are written in an upright font:

Checkout 3.8 Big list at [hyperref-manual](http://www.tug.org/hyperref-manual) at [tug.org](http://www.tug.org) ⁴

⁴ <http://www.tug.org/applications/hyperref/manual.html#x1-120003.8>

variable	values	comment
bookmarks	= <i>true, false</i>	show or hide the bookmarks bar when displaying the document
unicode	= <i>false, true</i>	allows to use characters of non-Latin based languages in Acrobat's bookmarks

variable	values	comment
pdfborder	<pre>= { RadiusH RadiusV Width [Dash-Pattern] }</pre>	<p>set the style of the border around a link. The first two parameters (RadiusH, RadiusV) have no effect in most pdf viewers. <i>Width</i> defines the thickness of the border. <i>Dash-Pattern</i> is a series of numbers separated by space and enclosed by box-brackets. It is an optional parameter to specify the length of each line & gap in the dash pattern. For example, {0 0 0.5 [3</p>

variable	values	comment
pdftoolbar	=true, false	show or hide Acrobat's toolbar
pdfmenubar	=true, false	show or hide Acrobat's menu
pdfwindow	=true, false	resize document window to fit document size
pdfstartview	={FitH}, {FitV}, et	fit ⁵ the width of the page to the window
pdftitle	= {text}	define the title that gets displayed in the "Document Info" window of Acrobat

5 Other possible values are defined in the `hyperref` manual ^{<http://mirror.switch.ch/ftp/mirror/tex/macros/latex/contrib/hyperref/doc/manual.html#TBL-7-40-1>}

variable	values	comment
pdfauthor	= {text}	the name of the PDF's author, it works like the one above
pdfsubject	= {text}	subject of the document, it works like the one above
pdfcreator	= {text}	creator of the document, it works like the one above
pdfproducer	= {text}	producer of the document, it works like the one above
pdfkeywords	= {text}	list of keywords, separated by brackets, example below

variable	values	comment
pdfnewwindow	$(=true, false)$	define if a new window should get opened when a link leads out of the current document
pagebackref	$(=false, true)$	activate back references inside bibliography. Must be specified as part of the <code>\usepackage{}</code> statement.

variable	values	comment
colorlinks	(=false, true)	surround the links by color frames (false) or colors the text of the links (true). The color of these links can be configured using the following options (default colors are shown):
hidelinks		hide links (removing color and border)
linkcolor	=red	color of internal links (sections, pages, etc.)

variable	values	comment
linktoc	=none, section, page	defines which part of an entry in the table of contents is made into a link
citecolor	=green	color of citation links (bibliography)
filecolor	=magenta	color of file links
urlcolor	=cyan	color of URL links (mail, web)
linkbordercolor	={1 0 0}	color of frame around internal links (if colorlinks=false)
citebordercolor	={0 1 0}	color of frame around citations
urlbordercolor	={0 1 1}	color of frame around URL links

Please note, that explicit RGB specification is only allowed for the border colors (like `linkbordercolor` etc.), while the others may only assigned to named colors (which you can define your own, see `Colors`⁶). In order to speed up your customization process, here is a list with the variables with their default value. Copy it in your document and make the changes you want. Next to the variables, there is a short explanations of their meaning:

```
\hypersetup{
  bookmarks=true,           % show bookmarks bar?
  unicode=false,          % non-Latin characters in Acrobat's
  bookmarks                %
  pdftoolbar=true,        % show Acrobat's toolbar?
  pdfmenubar=true,       % show Acrobat's menu?
  pdfwindow=false,       % window fit to page when opened
  pdfstartview={FitH},   % fits the width of the page to the
  window
  pdftitle={My title},   % title
  pdfauthor={Author},    % author
  pdfsubject={Subject},  % subject of the document
  pdfcreator={Creator},  % creator of the document
}
```

```

pdfproducer={Producer}, % producer of the document
pdfkeywords={keyword1} {key2} {key3}, % list of keywords
pdfnewwindow=true, % links in new window
colorlinks=false, % false: boxed links; true: colored links
linkcolor=red, % color of internal links
citecolor=green, % color of links to bibliography
filecolor=magenta, % color of file links
urlcolor=cyan % color of external links
}

```

If you don't need such a high customization, here are some smaller but useful examples. When creating PDFs destined for printing, colored links are not a good thing as they end up in gray in the final output, making it difficult to read. You can use color frames, which are not printed:

```

\usepackage{hyperref}
\hypersetup{colorlinks=false}

```

or make links black:

```

\usepackage{hyperref}

```

```
\hypersetup{
  colorlinks,%
  citecolor=black,%
  filecolor=black,%
  linkcolor=black,%
  urlcolor=black
}
```

When you just want to provide information for the Document Info section of the PDF file, as well as enabling back references inside bibliography:

```
\usepackage[pdfauthor={Author's name},%
pdftitle={Document Title},%
pagebackref=true,%
pdfutex]{hyperref}
```

By default, URLs are printed using mono-spaced fonts. If you don't like it and you want them to be printed with the same style of the rest of the text, you can use this:

```
\urlstyle{same}
```

14.4. Problems with Links and Equations

Messages like the following

```
! pdfTeX warning (ext4): destination with the same identifier (name{
equation.1.7.7.30}) has been already used, duplicate ignored
```

appear, when you have made something like

```
\begin{eqnarray}a=b\nonumber\end{eqnarray}
```

The error disappears, if you use instead this form:

```
\begin{eqnarray*}a=b\end{eqnarray*}
```

Beware that the shown line number is often completely different from the erroneous line.

Possible solution: Place the `amsmath` package before the `hyperref` package.

14.5. Problems with Links and Pages

Messages like the following:

```
! pdfTeX warning (ext4): destination with the same
  identifier (name{page.1}) has been already used,
  duplicate ignored
```

appear when a counter gets reinitialized, for example by using the command

```
\mainmatter
```

provided by the book document class. It resets the page number counter to 1 prior to the first chapter of the book. But as the preface of the book also has a page number 1 all links to "page 1" would not be unique anymore, hence the notice that "duplicate has been ignored." The counter measure

consists of putting `plainpages=false` into the `hyperref` options. This unfortunately only helps with the page counter. An even more radical solution is to use the option `hypertexnames=false`, but this will cause the page links in the index to stop working.

The best solution is to give each page a unique name by using the

```
\pagenumbering
```

command:

```
\pagenumbering{alph} % a, b, c, ...  
... titlepage, other front matter ...  
\pagenumbering{roman} % i, ii, iii, iv, ...  
... table of contents, table of figures, ...  
\pagenumbering{arabic} % 1, 2, 3, 4, ...  
... beginning of the main matter (chapter 1) ...
```

Another solution is to use

```
\pagenumbering{alph}
```

before the command

```
\maketitle
```

, which will give the title page the label page.a. Since the page number is suppressed, it won't make a difference to the output.

By changing the page numbering every time before the counter is reset, each page gets a unique name. In this case, the pages would be numbered a, b, c, i, ii, iii, iv, v, 1, 2, 3, 4, 5, ...

If you don't want the page numbers to be visible (for example, during the front matter part), use

```
\pagestyle{empty} ... \pagestyle{plain}
```

. The important point is that although the numbers are not visible, each page will have a unique name.

Another more flexible approach is to set the counter to something negative:

```
\setcounter{page}{-100}  
... titlepage, other front matter ...  
\pagenumbering{roman} % i, ii, iii, iv, ...
```

```
... table of contents, table of figures, ...
\pagenumbering{arabic} % 1, 2, 3, 4, ...
... beginning of the main matter (chapter 1) ...
```

which will give the first pages a unique negative number.

The problem can also occur with the `algorithms` package: because each algorithm uses the same line-numbering scheme, the line identifiers for the second and follow-on algorithms will be duplicates of the first.

The problem occurs with equation identifiers if you use

```
\nonumber
```

on every line of an `eqnarray` environment. In this case, use the `*`'ed form instead, e.g.

```
\begin{eqnarray*} ... \end{eqnarray*}
```

(which is an unnumbered equation array), and remove the now unnecessary

```
\nonumber
```

commands.

If your url's are too long and running off of the page, try using the `breakurl` package to split the url over multiple lines. This is especially important in a multicolumn environment where the line width is greatly shortened.

14.6. Problems with Bookmarks

The text displayed by bookmarks does not always look like you expect it to look. Because bookmarks are "just text", much fewer characters are available for bookmarks than for normal LaTeX text. Hyperref will normally notice such problems and put up a warning:

```
Package hyperref Warning:
Token not allowed in a PDFDocEncoded string:
```

You can now work around this problem by providing a text string for the bookmarks, which replaces the offending text:

```
\texorpdfstring{'TEX text'}{'Bookmark Text'}
```

Math expressions are a prime candidate for this kind of problem:

```
\section{\texorpdfstring{$E=mc^2$}{E=mc2}}
```

which turns

```
\section{$E=mc^2$}
```

to $E=mc^2$ in the bookmark area. Color changes also do not travel well into bookmarks:

```
\section{\textcolor{red}{Red !}}
```

produces the string "redRed!". The command

```
\textcolor
```

gets ignored but its argument (red) gets printed. If you use:

```
\section{\texorpdfstring{\textcolor{red}{Red !}}{Red\ !}}
```

the result will be much more legible.

If you write your document in unicode and use the `unicode` option for the `hyperref` package you can use unicode characters in bookmarks. This will give you a much larger selection of characters to pick from when using

```
\texorpdfstring
```

.

14.7. Problems with tables and figures

The links created by `hyperref` point to the label created within the `float` environment, which, as previously described⁷, must always be set after the caption. Since the caption is usually below a figure or table,

the figure or table itself will not be visible upon clicking the link⁸. A workaround exists by using the package `hypcap` <http://www.ctan.org/tex-archive/macros/latex/contrib/oberdiek/hypcap.pdf>⁹ with:

```
\usepackage[all]{hypcap}
```

Be sure to call this package *after* loading `hyperref`, which should otherwise be loaded last.

If you use the `wrapping` package mentioned in the "Wrapping text around figures¹⁰" section of the "Floats, Figures and Captions" chapter, or other similar packages that define their own environments, you will need to manually include

```
\capstart
```

in those environments, e.g.:

-
- 8 <http://www.ctan.org/tex-archive/macros/latex/contrib/hyperref/README>
 - 9 <http://www.ctan.org/tex-archive/macros/latex/contrib/oberdiek/hypcap.pdf>
 - 10 Chapter 13.2.4 on page 414

```
\begin{wrapfigure}{R}{0.5\textwidth}
\capstart
\begin{center}
\includegraphics[width=0.48\textwidth]{filename}
\end{center}
\caption{\label{labelname}a figure}
\end{wrapfigure}
```

14.8. Problems with long caption and \listoffigures or long title

There is an issue when using

```
\listoffigures
```

with `\hyperref` for long captions or long titles. This happens when the captions (or the titles) are longer than the page width (about 7-9 words depending on your settings). To fix this, you need to use the option `breaklinks` when first declaring:

```
\usepackage[breaklinks]{hyperref}
```

This will then cause the links in the

```
\listoffigures
```

to word wrap properly.

14.9. Problems with already existing `.toc`, `.lof` and similar files

The format of some of the auxilliary files generated by latex changes when you include the `hyperref` package. One can therefore encounter errors like

```
! Argument of \Hy@setref@link has an extra }. when
```

the document is typeset with `hyperref` for the first time and these files already exist. The solution to the problem is to delete all the files that latex uses to get references right and typeset again.

14.10. Problems with footnotes and special characters

See the relevant section¹¹.

14.11. Problems with Beamer

Using the command

```
\hyperref[some_label]{some text}
```

is broken when pointed at a label. Instead of sending the user to the desired label, upon clicking the user will be sent to the first frame. A simple work around exists;

instead of using

```
\phantomsection\label{some_label}
```

¹¹ http://en.wikibooks.org/wiki/LaTeX%2FFormatting%23Common_problems_and_workarounds

to label your frames, use

```
\hypertarget{some_label}{}
```

and reference it with

```
\hyperlink{some_label}{some text}
```

.

14.12. References

15. Colors

Adding colors to your text is supported by the `color` package. Using this package, you can set the color of the font of the text, and set the background color of the page. You can use one of the predefined colors such as *white*, *red*, or *yellow*, or you can define your own named colors. It's also possible to color formulas in math-environments.

15.1. Adding the color package

To make use of these color features the color package must be inserted into the preamble.

```
\usepackage{color}
```

You may also want to add to the preamble after the previous line:

```
\usepackage[usenames,dvipsnames,svgnames,table]{xcolor}
```

The `\usepackage` is obvious, but the initialization of additional commands like *usenames* allows you to use names of the default colors, the same 16 base colors as used in HTML. The *dvipsnames* allows you access to more colors, another 64, and *svgnames* allows access to about 150 colors. The initialization of "table" allows colors to be added to tables by placing the color command just before the table. The package loaded here is the "xcolor" package.

If you need more colors, then you may also want to look at adding the *xl1names* to the initialization section as well, this offers more than 300 colors, but you need to make sure your `xcolor` package is the most recent you can download.

15.2. Entering colored text

The simplest way to type colored text is by:

```
\textcolor{declared-color}{text}
```

where *declared-color* is a color that was defined before by

```
\definecolor
```

.

Another possible way is by

```
{\color{declared-color} text}
```

that will switch the standard text color to the color you want. It will work until the end of the current TeX group. For example:

```
\emph{some black text, \color{red} followed by a red fragment}, going  
black again.
```

some black text, followed by a red fragment, going black again.

Figure 51

The difference between

```
\textcolor
```

and

```
\color
```

is the same as that between

```
\textttt
```

and

```
\ttfamily
```

, you can use the one you prefer.

You can change the background color of the whole page by:

```
\pagecolor{declared-color}
```

15.3. Entering colored background for the text

```
\colorbox{declared-color}{text}
```

If the background color and the text color is changed, then:

```
\colorbox{declared-color1}{\color{declared-color2}{text}}
```

There is also `\fcolorbox` to make framed background color in yet another color:

```
\colorbox{declared-color-frame}{declared-color-background}{text}
```

15.4. Predefined colors

The predefined color names are `white`, `black`, `red`, `green`, `blue`, `cyan`, `magenta`, `yellow`. There may be other pre-defined colors on your system, but these should be available on all systems.

If you would like a color not pre-defined, you can use one of the 68 dvips colors, or define your own. These options are discussed in the following sections

15.5. The 68 standard colors known to dvips

Invoke the package with the `usenames` and `dvipsnames` option. If you are using TikZ package you must declare the color package before that, otherwise it will not work.

```
\usepackage[usenames,dvipsnames]{xcolor}
```



Apricot



Aquamarine



Bittersweet



Black



Blue



BlueGreen



BlueViolet



BrickRed



Brown



BurntOrange



CadetBlue



CarnationPink



Cerulean



CornflowerBlue



Cyan



Dandelion



DarkOrchid



Emerald



ForestGreen



Fuchsia



Goldenrod



Gray



Green



GreenYellow



JungleGreen



Lavender



LimeGreen



Magenta



Mahogany



Maroon



Melon



MidnightBlue



Mulberry



NavyBlue



OliveGreen



Orange



OrangeRed



Orchid



Peach



Periwinkle



PineGreen



Plum



ProcessBlue



Purple



RawSienna



Red



RedOrange



RedViolet



Rhodamine



RoyalBlue



RoyalPurple



RubineRed



Salmon



SeaGreen



Sepia



SkyBlue



SpringGreen



Tan



TealBlue



Thistle



Turquoise



Violet



VioletRed

White



WildStrawberry



Yellow



YellowGreen



YellowOrange

15.6. Defining new colors

If the predefined colors are not adequate, you may wish to define your own.

15.6.1. Place

Define the colors in the *preamble* of your document. (Reason: Do so in the preamble, so that you can already refer to them in the preamble, which is useful, for instance, in an argument of another package that supports colors as arguments, such as listings¹ package.)

¹ <http://en.wikibooks.org/wiki/LaTeX%2FPackages%2FListings>

15.6.2. Method

To define a new color, follow the following example, which defines orange for you, by setting the red to the maximum, the green to one half (0.5), and the blue to the minimum:

```
\definecolor{orange}{rgb}{1,0.5,0}
```

You can use small letters **rgb** and choose a value between 0 and 1 or use capital letters **RGB** and choose a value between 0 and 255. The following code should give a similar results to the last code chunk.

```
\definecolor{orange}{RGB}{255,127,0}
```

Trick : When surfing on the web, you can get hexadecimal code for each color on a web page using the *colorzilla* extension to Firefox.

If you loaded the `xcolor` package, you can define colors upon previously defined ones:

```
\colorlet{notgreen}{blue!50!yellow}
```

In the abstract, the colors are defined following this scheme:

```
\definecolor{name''}{model''}{'color-spec''}
```

where:

- *name* is the name of the color; you can call it as you like
- *model* is the way you *describe* the color, and is one of *gray*, *rgb* and *cmyk*.
- *color-spec* is the description of the color

15.6.3. Color Models

Among the models you can use to describe the color are the following (several more are described in the `xcolor` manual²):

Color Models		
Model Description	Color Specification	Example
<code>gray</code> Shades of gray.	just one number between 0 (black) and 1 (white), so 0.95 will be very light gray, 0.30 will be dark gray	<code>\definecolor{light-gray</code>

² <http://mirror.ctan.org/macros/latex/contrib/xcolor/xcolor.pdf>

Color Models

	Model Description	Color Specification	Example
rgb	Red, Green, Blue	three numbers given in the form <i>red, green, blue</i> ; the quantity of each color is represented with a number between 0 and 1	\definecolor{orange}{rgb

Color Models

Model Description	Color Specification	Example
RGB Red, Green, Blue	three numbers given in the form <i>red, green, blue</i> ; the quantity of each color is represented with a number between 0 and 255	\definecolor{orange}{RGB
HTML Red, Green, Blue	six hexadecimal numbers given in the form <i>RRGGBB</i> ; similar to what is used in HTML	\definecolor{orange}{HTML

Color Models		
Model Description	Color Specification	Example
cmyk Cyan, Ma- genta, Yellow, Black	four numbers given in the form <i>cyan,magenta,yellow,black</i> ; the quantity of each color is represented with a number between 0 and 1	<pre>\definecolor{orange}{cmyk</pre>

15.7. Advanced color settings

The `xcolor` package provides extended versions of the described color related commands. Tints can be defined as follow:

```
\color{blue!20}  
\color{blue!20!black}  
\color{blue!20!black!30!green}
```

The first specifies 20 percent blue, the second is a mixture of 20 percent blue and 80 percent black and the last one is a mixture of $(20*0.3)$ percent blue, $(80*0.3)$ percent black and 70 percent green.

Other features include: support of hsb color model, html style notation, special row coloring support in tables and more.

15.8. Sources

- The xcolor manual³

³ <http://mirror.ctan.org/macros/latex/contrib/xcolor/xcolor.pdf>

16. Fonts

In order to select a font other than the default typeface in LaTeX environment, it is necessary to include some commands in the preamble of the document.

For example:

```
\usepackage[T1]{fontenc}  
\usepackage[light,math]{iwona}
```

Important: This only works for fonts that are already prepared for use with LaTeX. A TrueType¹ font (tff) or similar must first be converted and made available to LaTeX. The external links section below has some useful resources.

16.0.1. Example

Below is an example found at the Google discussion group latexlovers. The example demonstrates how to select different fonts in a simple document.

¹ <http://en.wikipedia.org/wiki/TrueType>

```
% default font (\familydefault = \rmdefault = Computer Modern Roman)
Lorem ipsum dolor sit amet, consectetur adipiscing elit.

% Palatino font
\renewcommand*\rmdefault{ppl}\normalfont\upshape
Lorem ipsum dolor sit amet, consectetur adipiscing elit.

% Iwona font
\renewcommand*\rmdefault{iwona}\normalfont\upshape
Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
```

Figure 52

16.1. XeTeX

If you use the XeTeX² or LuaTeX³ engine and the fontspec⁴ package, you'll be able to use any font installed in the system effortlessly. XeTeX also allows using OpenType⁵ technology of modern fonts like specifying alternate glyphs and optical size variants. XeTeX also uses Unicode⁶ by default, which might be helpful for font issues.

To use the fonts, simply load the fontspec package and set the font:

```
\documentclass{article}
\usepackage{fontspec}


---


2 http://en.wikipedia.org/wiki/XeTeX
3 http://en.wikipedia.org/wiki/LuaTeX
4 http://www.ctan.org/tex-archive/help/Catalogue/entries/
   fontspec.html
5 http://en.wikipedia.org/wiki/OpenType
6 http://en.wikipedia.org/wiki/Unicode
```

```
\setmainfont{Arial}  
  
\begin{document}  
Lorem ipsum...  
\end{document}
```

Then compile the document with XeLaTeX or LuaLaTeX. Note that you can only generate .pdf files, and that you need a sufficiently new TeX distribution (TeX Live 2009 should work for XeTeX and Tex Live 2010 for LuaTeX). Also you should NOT load the inputenc or fontenc package. To make support both pdflatex and XeTeX you can use the `\ifxetex` macro from the `ifxetex7` package.

```
\documentclass{article}  
\usepackage{ifxetex}  
  
\ifxetex  
  \usepackage{fontspec}  
  \usepackage{xunicode}  
  \defaultfontfeatures{Mapping=tex-text} % To support LaTeX quoting
```

⁷ <http://www.ctan.org/tex-archive/macros/generic/ifxetex/>

```
style
\setromanfont{Hoefler Text}
\else
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\fi

\begin{document}
Lorem ipsum...
\end{document}
```

16.2. Useful websites

- The Latex Font Catalogue⁸
- LaTeX font commands⁹
- How to change fonts in Latex¹⁰

⁸ <http://www.tug.dk/FontCatalogue/>

⁹ <http://www.cl.cam.ac.uk/~rf10/pstex/latexcommands.htm>

¹⁰ <http://www.ee.iitb.ac.in/~trivedi/LatexHelp/latexfont.htm>

- [\[ftp://tug.ctan.org/tex-archive/fonts/utilities/fontinst/doc/talks/et99-font-tutorial.pdf\]](http://tug.ctan.org/tex-archive/fonts/utilities/fontinst/doc/talks/et99-font-tutorial.pdf) Understanding the world of T_EX fonts and mastering the basics of fontinst]
- Font installation the shallow way¹¹ "*For one-off projects, you can cut corners with font installation (i.e. fontinst) and end up with a more manageable set of files and a cleaner T_EX installation. This article shows how and why*"

16.2.1. TrueType (ttf) fonts

- Step-by-step guide to manually install a ttf-font for PdfTeX¹²
- A bash script for installing a LaTeX font family¹³ (MikTeX¹⁴ / TeXLive¹⁵)

11 <http://www.tug.org/TUGboat/Articles/tb27-1/tb86kroonenberg-fonts.pdf>

12 <http://c.caignaert.free.fr/Install-ttf-Font.pdf>

13 <http://www.tex.ac.uk/ctan/support/installfont/installfont.pdf>

14 <http://latex.josef-kleber.de/download/installfont-tl>

15 <http://latex.josef-kleber.de/download/installfont-tl>

- LaTeX And TrueType Font¹⁶
- True Type Fonts with LaTeX under Linux + MiKTeX 2.5¹⁷
- Unicode Truetype font installer for LaTeX under Windows + MikTeX¹⁸
- Using TrueType fonts with TeX (LaTeX) and pdfTeX (pdfLaTeX)¹⁹ (for MikTeX)

16 <http://xpt.sourceforge.net/techdocs/language/latex/latex33-LaTeXAndTrueTypeFont>

17 <http://fachschaft.physik.uni-greifswald.de/~stitch/ttf.html>

18 <http://william.famille-blum.org/software/latex/ttf/index.html>

19 <http://www.radamir.com/tex/ttf-tex.htm>

Part III.

Technical Texts

17. Mathematics

One of the greatest motivating forces for Donald Knuth when he began developing the original TeX system was to create something that allowed simple construction of mathematical formulas, whilst looking professional when printed. The fact that he succeeded was most probably why TeX (and later on, LaTeX) became so popular within the scientific community. Typesetting mathematics is one of LaTeX's greatest strengths. It is also a large topic due to the existence of so much mathematical notation.

If your document requires only a few simple mathematical formulas, plain LaTeX has most of the tools that you will need. If you are writing a scientific document that contains numerous complicated formulas, the `amsmath`

package¹ introduces several new commands that are more powerful and flexible than the ones provided by LaTeX. The `mathtools` package fixes some `amsmath` quirks and adds some useful settings, symbols, and environments to `amsmath`.² To use either package, include:

```
\usepackage{amsmath}
```

or

```
\usepackage{mathtools}
```

in the preamble of the document. The `mathtools` package loads the `amsmath` package and hence there is no need to `\usepackage{amsmath}` in the preamble if `mathtools` is used.

1 <http://www.ams.org/publications/authors/tex/amslatex>

2 <http://www.tex.ac.uk/ctan/macros/latex/contrib/mh/mathtools.pdf>

17.1. Mathematics environments

LaTeX needs to know beforehand that the subsequent text does in fact contain mathematical elements. This is because LaTeX typesets maths notation differently than normal text. Therefore, special environments have been declared for this purpose. They can be distinguished into two categories depending on how they are presented:

- *text* - text formulas are displayed in-line, that is, within the body of text where it is declared. e.g., I can say that $a + a = 2a$ within this sentence.
- *displayed* - displayed formulas are separate from the main text.

As maths require special environments, there are naturally the appropriate environment names you can use in the standard way. Unlike most other environments, however, there are some handy shorthands to declaring your formulas. The following table summarizes them:

Type	Environment	LaTeX short-hand	TeX short-hand
Text	<code>\begin{math}...\end{math}</code>	<code>\(...\)</code>	<code>\$...\$</code>
Displayed	<code>\begin{displaymath}...\end{displaymath}</code>	<code>\end{displaymath}</code>	<code>\\$ath}</code>
	or <code>\begin{equation*}...\end{equation*}</code> 3		

Note: Using the `$$...$$` should be avoided, as it may cause problems, particularly with the AMS-LaTeX macros. Furthermore, should a problem occur, the error messages may not be helpful.

Additionally, there is a second possible environment for the *displayed* type of formulas: `equation`. The difference between `this` and `displaymath` is that `equation` also adds sequential equation numbers by the side.

If you are typing text normally, you are said to be in *text mode*, while you are typing within one of those mathematical environments, you are said to be in *math mode*, that has some differences compared to the *text mode*:

1. Most spaces and line breaks do not have any significance, as all spaces are either derived logically from the mathematical expressions, or have to be specified with special commands such as `\quad`
2. Empty lines are not allowed. Only one paragraph per formula.
3. Each letter is considered to be the name of a variable and will be typeset as such. If you want to typeset normal text within a formula (normal upright font and normal spacing) then you have to enter the text using dedicated commands.⁴

17.1.1. Inserting "Displayed" maths inside blocks of text

In order for some operators, such as `\lim` or `\sum` to be displayed correctly inside some math environments (read `$ $`), it might be convenient to write the `\displaystyle` class inside the environment. Doing so might cause the line to be taller, but will cause exponents and indices to be displayed correctly for some math operators. For example, the `\sum` will print a smaller Σ and `$\displaystyle \sum$` will print a bigger one \sum , like in equations (This only works with AMSMATH package).

17.2. Symbols

Mathematics has lots and lots of symbols! If there is one aspect of maths that is difficult in LaTeX it is trying to remember how to produce them. There are of course a set of symbols that can be accessed directly from the keyboard:

+ - = ! / () [] < > | ' :

Beyond those listed above, distinct commands must be issued in order to display the desired symbols. And there are *a lot!* of Greek letters, set and relations symbols, arrows, binary operators, etc. For example:

```
\[
\forall x \in X, \quad \exists y \leq \epsilon
\]
```

$\forall x \in X, \quad \exists y \leq \epsilon$

Fortunately, there's a tool that can greatly simplify the search for the command for a specific symbol. Look for "Detexify" in the external links⁵ section below.

Another option would be to look in the "The Comprehensive LaTeX Symbol List" in the external links⁶ section below.

17.3. Greek letters

Greek letters are commonly used in mathematics, and they are very easy to type in *math mode*. You just have to type the name of the letter after a backslash: if the first letter is lowercase, you will get a lowercase Greek letter, if the first letter is uppercase (and only the first letter), then you will get an uppercase letter. Note that some uppercase Greek letters look like Latin ones, so they are not provided by LaTeX (e.g. uppercase *Alpha* and *Beta* are just "A" and "B" respectively). Lowercase epsilon, theta, phi, pi, rho, and sigma are provided in two different versions. The alternate, or *variant*, version is created by adding "var" before the name of the letter:

17.4. Operators

An operator is a function that is written as a word: e.g. trigonometric functions (sin, cos, tan), logarithms and exponentials (log, exp). LaTeX has many of these defined as commands:

```
\[
\cos (2\theta) = \cos^2 \theta - \sin^2 \theta
\]
```

$$\cos(2\theta) = \cos^2\theta - \sin^2\theta$$

For certain operators such as limits⁸, the subscript is placed underneath the operator:

⁸ <http://en.wikipedia.org/wiki/Limit%20%28mathematics%29>

```
\[
\lim_{x \to \infty} \exp(-x) = 0
\]
```

$\lim_{x \rightarrow \infty} \exp(-x) = 0$

For the modular operator⁹ there are two commands: `\bmod` and `\pmod`:

```
\[
a \bmod b
\]
```

a mod b

⁹ <http://en.wikipedia.org/wiki/Modular%20arithmetic>

```
\[
x \equiv a \pmod b
\]
```

$x \equiv a \pmod b$

To use operators which are not pre-defined, such as `argmax`¹⁰, see custom operators¹¹

17.5. Powers and indices

Powers and indices are equivalent to superscripts and subscripts in normal text mode. The caret (`^`) character is used to raise something, and the underscore

¹⁰ <http://en.wikipedia.org/wiki/argmax>

¹¹ Chapter 18.6 on page 628

() is for lowering. If more than one expression is raised or lowered, they should be grouped using curly braces (`{ }`).

$$\backslash[k_{\{n+1\}} = n^2 + k_{n^2} - k_{\{n-1\}} \backslash]$$

$$k_{n+1} = n^2 + k_n^2 - k_{n-1}$$

An underscore (`_`) can be used with a vertical bar (`|`) to denote evaluation using subscript notation in mathematics:

$$\backslash[f(n) = n^5 + 4n^2 + 2_{\{n=17\}} \backslash]$$

$$f(n) = n^5 + 4n^2 + 2|_{n=17}$$

17.6. Fractions and Binomials

A fraction is created using the `\frac{numerator}{denominator}` command. (for those who need their memories refreshed, that's the *top* and *bottom* respectively!). Likewise, the binomial coefficient¹² (aka the Choose function) may be written using the `\binom` command¹³:

```
\[
  \frac{n!}{k!(n-k)!} = \binom{n}{k}
\]
```

$$\frac{n!}{k!(n-k)!} = \binom{n}{k}$$

¹² <http://en.wikipedia.org/wiki/Binomial%20coefficient>

¹³ requires the `amsmath` package

It is also possible to use the `\choose` command without the `amsmath` package:

```
\[
\frac{n!}{k!(n-k)!} = {n \choose k}
\]
```

$$\frac{n!}{k!(n-k)!} = \binom{n}{k}$$

You can embed fractions within fractions:

```
\[
\frac{\frac{1}{x}+\frac{1}{y}}{y-z}
\]
```

$$\frac{\frac{1}{x} + \frac{1}{y}}{y - z}$$

Note that when appearing inside another fraction, or in inline text $\frac{a}{b}$, a fraction is noticeably smaller than in displayed mathematics. The `\frac` and `\dfrac` commands¹⁴ force the use of the respective styles, `\textstyle` and `\displaystyle`. Similarly, the `\tbinom` and `\dbinom` commands typeset the binomial coefficient.

Another way to write fractions is to use the `\over` command without the `amsmath` package:

14 requires the `amsmath` package

$$\frac{n!}{k!(n-k)!} = \binom{n}{k}$$

$$\frac{n!}{k!(n-k)!} = \binom{n}{k}$$

For relatively simple fractions, it may be more aesthetically pleasing to use powers and indices¹⁵:

$$x^3 / y^7$$

If you use them throughout the document, usage of `xfrac` package is recommended. This package provides `\sfrac` command to create slanted fractions. Usage:

```
Take \sfrac{1}{2} cup of sugar, \dots
\l
3\times\sfrac{1}{2}{2}=1\sfrac{1}{2}
\l
```

```
Take ${}^1/_2$ cup of sugar, \dots
\l
3\times{}^1/_2=1{}^1/_2
\l
```

Take $\frac{1}{2}$ cup of sugar, ...

$$3 \times \frac{1}{2} = 1\frac{1}{2}$$

Take $\frac{1}{2}$ cup of sugar, ...

$$3 \times \frac{1}{2} = 1\frac{1}{2}$$

Figure 53

Alternatively, the `nicefrac` package provides the `\nicefrac` command, whose usage is similar to `\frac`.

17.6.1. Continued fractions

Continued fractions should be written using `\cfrac` command¹⁶:

```
\begin{equation}
x = a_0 + \cfrac{1}{a_1}
      + \cfrac{1}{a_2}
      + \cfrac{1}{a_3 + a_4} <!------> <!------>
\end{equation}
```

$$x = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + a_4}}}$$

16 requires the `amsmath` package

17.6.2. Multiplication of two numbers

To make multiplication visually similar to a fraction, a nested array can be used, for example multiplication of numbers written one below the other.

```
\begin{equation}
\frac{
\begin{array}[b]{r}
\left( x_1 x_2 \right) \\
\times \left( x'_1 x'_2 \right)
\end{array}
}{
\left( y_1 y_2 y_3 y_4 \right)
}
\end{equation}
```

$$\frac{(x_1 x_2) \times (x'_1 x'_2)}{(y_1 y_2 y_3 y_4)}$$

17.7. Roots

The `\sqrt` command creates a square root surrounding an expression. It accepts an optional argument specified in square brackets (`[and]`) to change magnitude:

```
\[  
\sqrt{\frac{a}{b}}  
\]
```

$$\sqrt{\frac{a}{b}}$$

```
\[  
\sqrt[n]{1+x+x^2+x^3+\ldots}  
\]
```

$$\sqrt[n]{1+x+x^2+x^3+\dots}$$

Some people prefer writing the square root "closing" it over its content. This method arguably makes it more clear what is in the scope of the root sign. This habit is not normally used while writing with the computer, but if you still want to change the output of the square root, LaTeX gives you this possibility. Just add the following code in the preamble of your document:

```

% New definition of square root:
% it renames \sqrt as \oldsqrt
\let\oldsqrt\sqrt
% it defines the new \sqrt in terms of the old one
\def\sqrt{\mathpalette\DHLhksqrt}
\def\DHLhksqrt#1#2{%
\setbox0=\hbox{#1\oldsqrt{#2\,}}\dimen0=\ht0
\advance\dimen0-0.2\ht0
\setbox2=\hbox{\vrule height\ht0 depth -\dimen0}%
{\box0\lower0.4pt\box2}}

```



Figure 54 The new style is on left, the old one on right

This TeX code first renames the `\sqrt` command as `\oldsqrt`, then redefines `\sqrt` in terms of the old one, adding something more. The new square root can be seen in the picture on the left, compared to the old one on the right. Unfortunately this code won't work if you want to use multiple roots: if you try to write $\sqrt[b]{a}$ as `\sqrt[b]{a}` after you used the code above, you'll just get a wrong output. In other words, you can redefine the square root this way only if you are not going to use multiple roots in the whole document.

An alternative piece of TeX code that does allow multiple roots is


```

\LetLtxMacro{\oldsqrt}{\sqrt} % makes all sqrts closed
\renewcommand{\sqrt}[1][ ]{%
  \def\DHLindex{#1}\mathpalette\DHLhksqrt}
\def\DHLhksqrt#1#2{%
  \setbox0=\hbox{${#1}\oldsqrt[\DHLindex]{#2},}$}\dimen0=\ht0
  \advance\dimen0-0.2\ht0
  \setbox2=\hbox{\vrule height\ht0 depth -\dimen0}%
  {\box0\lower0.71pt\box2}<!-->}

$\sqrt{a}{b} \quad \oldsqrt{a}{b}$

```

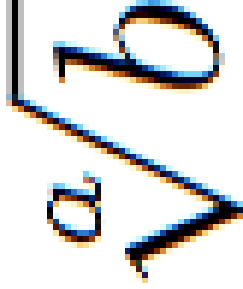
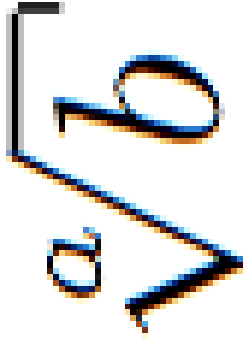


Figure 55

However this requires the `\usepackage{letltxmacro}` package

17.8. Sums and integrals

The `\sum` and `\int` commands insert the sum and integral symbols respectively, with limits specified using the caret (`^`) and underscore (`_`). The typical notation for sums is:

```
\[
\sum_{i=1}^{10} t_{-i}
\]
```

```
\sum_{i=1}^{10} t_i
```

The limits for the integrals follow the same notation. It's also important to represent the integration variables with an upright d , which in math mode is obtained through the `\mathrm{}` command, and with a small space separating it from the integrand, which is attained with the `\,` command.

```
\[
\int_0^\infty e^{-x}\, \mathrm{d}x
\]
```

$$\int_0^\infty e^{-x} dx$$

There are many other "big" commands which operate in a similar manner:

\sum	\prod	\coprod
\bigoplus	\bigotimes	\bigodot
\bigcup	\bigcap	\biguplus
\bigsqcup	\bigvee	\bigwedge
\int	\oint	$\int\!\!\int$

`\iiint`¹⁸ \iiint

`\iiint`¹⁹ \iiint

`\idotsint`²⁰ \int

For more integral symbols, including those not included by default in the Computer Modern font, try the `esint` package.

The `\substack` command²¹ allows the use of `\\` to write the limits over multiple lines:

18 requires the `amsmath` package

19 requires the `amsmath` package

20 requires the `amsmath` package

21 requires the `amsmath` package

```

\l
\sum_{\substack{
0 < i < m \\
0 < j < n \\
} < ! ----->}
P(i, j)
\l

```

$$\sum_{\substack{0 < i < m \\ 0 < j < n}} P(i, j)$$

If you want the limits of an integral to be specified above and below the symbol (like the sum), use the `\limits` command:

```
\[
\int\limits_a^b
\]
```

$$\int_a^b$$

However if you want this to apply to ALL integrals, it is preferable to specify the `intlimits` option when loading the `amsmath` package:

```
\usepackage[intlimits]{amsmath}
```

Subscripts and superscripts in other contexts as well as other parameters to `amsmath` package related to them are described in [Advanced Mathematics](#)²² chapter.

For bigger integrals, you may use personal declarations, or the `bigints` package²³.

17.9. Brackets, braces and delimiters

How to use braces in multi line equations is described in the [Advanced Mathematics](#)²⁴ chapter.

The use of delimiters such as brackets soon becomes important when dealing with anything but the most trivial equations. Without them, formulas can become ambiguous. Also, special types of mathematical structures, such as matrices, typically rely on delimiters to enclose them.

There are a variety of delimiters available for use in LaTeX:

²³ <http://hdl.handle.net/2268/6219>

²⁴ Chapter 18.2.3 on page 609


```

\l
(a), [ b ], {\ c \}, d, \ e \,
\l angle f \r angle, \lfloor g \rfloor,
\lceil h \rceil
\j

```

$(a), [b], \{c\}, |d|, \|e\|, \langle f \rangle, [g], [h]$

17.9.1. Automatic sizing

Very often mathematical features will differ in size, in which case the delimiters surrounding the expression should vary accordingly. This can be done automatically using the `\left` and `\right` commands. Any of the previous delimiters may be used in combination with these:

```
\[
\left(\frac{x^2}{y^3}\right)
\]
```

$$\left(\frac{x^2}{y^3}\right)$$

Curly braces are defined differently by using `\left\{` and `\right\}` ,

```
\[
\left\{\frac{x^2}{y^3}\right\}
\]
```

$$\left\{\frac{x^2}{y^3}\right\}$$

If a delimiter on only one side of an expression is required, then an invisible delimiter on the other side may be denoted using a period (`.`).

```
\left.\frac{x^3}{3}\right|_0^1
```

$$\left.\frac{x^3}{3}\right|_0^1$$

17.9.2. Manual sizing

In certain cases, the sizing produced by the `\left` and `\right` commands may not be desirable, or you may simply want finer control over the delimiter

sizes. In this case, the `\big`, `\Big`, `\bigg` and `\Bigg` modifier commands may be used:

```
\[  
  ( \big( \Big( \bigg( \Bigg(  
\]
```

$((((($

17.10. Matrices and arrays

A basic matrix may be created using the `matrix` environment²⁵: in common with other table-like structures, entries are specified by row, with columns separated using an ampersand (&) and a new row separated with a double backslash (`\\`)

```
\[
  \begin{matrix}
    a & b & c \\
    d & e & f \\
    g & h & i
  \end{matrix}
\]
```

²⁵ requires the `amsmath` package

a b c
d e f
g h i

To specify alignment of columns in the table, use starred version²⁶:

26 requires the `mathtools` package

```

\[
\begin{matrix}
-1 & 3 \\
2 & -4
\end{matrix}
=
\begin{matrix*}[r]
-1 & 3 \\
2 & -4
\end{matrix*}
\]

```

```

-1 & 3 & -1 & 3 \\
2 & -4 & = & 2 & -4

```

The alignment by default is `c` but it can be any column type valid in `array` environment.

However matrices are usually enclosed in delimiters of some kind, and while it is possible to use the `\left` and `\right` commands²⁷, there are various other predefined environments which automatically include delimiters:

Environment name	Surrounding delimiter	Notes
<code>pmatrix</code> ²⁸	<code>()</code>	centers columns by default
<code>pmatrix*</code> ²⁹	<code>()</code>	allows to specify alignment of columns in optional parameter
<code>bmatrix</code> ³⁰	<code>[]</code>	centers columns by default

27 Chapter 17.9.1 on page 534

28 requires the `amsmath` package

29 requires the `mathtools` package

30 requires the `amsmath` package

Environment name	Surrounding delimiter	Notes
<code>bmatrix*</code> ³¹	<code>[]</code>	allows to specify alignment of columns in optional parameter
<code>Bmatrix</code> ³²	<code>{}</code>	centers columns by default
<code>Bmatrix*</code> ³³	<code>{}</code>	allows to specify alignment of columns in optional parameter
<code>vmatrix</code> ³⁴	<code> </code>	centers columns by default

³¹ requires the `mathtools` package

³² requires the `amsmath` package

³³ requires the `mathtools` package

³⁴ requires the `amsmath` package

Environment name	Surrounding delimiter	Notes
<code>vmatrix*</code> ³⁵	<code> </code>	allows to specify alignment of columns in optional parameter
<code>Vmatrix</code> ³⁶	<code> </code>	centers columns by default
<code>Vmatrix*</code> ³⁷	<code> </code>	allows to specify alignment of columns in optional parameter

When writing down arbitrary sized matrices, it is common to use horizontal, vertical and diagonal triplets of dots (known as ellipses³⁸) to fill in certain

³⁵ requires the `mathtools` package

³⁶ requires the `amsmath` package

³⁷ requires the `mathtools` package

³⁸ <http://en.wikipedia.org/wiki/ellipsis>

columns and rows. These can be specified using the `\cdots`, `\vdots` and `\ddots` respectively:

```
\[
  A_{m,n} =
  \begin{pmatrix}
    a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\
    a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\
    \vdots & \vdots & \ddots & \vdots \\
    a_{m,1} & a_{m,2} & \cdots & a_{m,n}
  \end{pmatrix}
\]
```

$$A_{m,n} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix}$$

In some cases you may want to have finer control of the alignment within each column, or want to insert lines between columns or rows. This can be achieved using the `array` environment, which is essentially a math-mode version of the `tabular` environment³⁹, which requires that the columns be pre-specified:

```
\[
\begin{array}{cc}
1 & 2 \\
\hline
3 & 4
\end{array}
\]
```

$$\begin{array}{c|c} 1 & 2 \\ \hline 3 & 4 \end{array}$$

You may see that the AMS matrix class of environments doesn't leave enough space when used together with fractions resulting in output similar to this:

$$M = \begin{bmatrix} \frac{5}{6} & \frac{1}{6} & 0 \\ \frac{5}{6} & 0 & \frac{1}{6} \\ 0 & \frac{5}{6} & \frac{1}{6} \end{bmatrix}$$

To counteract this problem, add additional leading space with the optional parameter to the `\` command:

```
\[
M = \begin{bmatrix}
\frac{5}{6} & \frac{1}{6} & 0 & \frac{5}{6} & \frac{1}{6} & 0 \\
\frac{5}{6} & 0 & \frac{1}{6} & 0 & \frac{5}{6} & \frac{1}{6} \\
0 & \frac{5}{6} & \frac{1}{6} & 0 & \frac{5}{6} & \frac{1}{6}
\end{bmatrix}
\]
```

$$M = \begin{bmatrix} \frac{5}{6} & \frac{1}{6} & 0 \\ \frac{5}{6} & 0 & \frac{1}{6} \\ 0 & \frac{5}{6} & \frac{1}{6} \end{bmatrix}$$

If you need "border" or "indexes" on your matrix, plain TeX provides the macro `\bordermatrix`

```
\[
M = \bordermatrix{~ & x & y \cr
A & 1 & 0 \cr
B & 0 & 1 \cr}
\]
```

$$M = \begin{matrix} & x & y \\ \begin{matrix} A \\ B \end{matrix} & \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \end{matrix}$$

Figure 56

17.10.1. Matrices in running text

To insert a small matrix, and not increase leading in the line containing it, use `smallmatrix` environment:

```
A matrix in text must be set smaller:  
$\bigl(\begin{smallmatrix}  
a&b\\ c&d  
\end{smallmatrix}\biggr)$  
to not increase leading in a portion of text.
```

A matrix in text must be set smaller: $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ to not increase leading in a portion of text.

Figure 57

17.11. Adding text to equations

The math environment differs from the text environment in the representation of text. Here is an example of trying to represent text within the math environment:


```
\[
50 apples \times 100 apples = lots of apples^2
\]
```

50apples \times *100apples* = *lotsofapples*²

There are two noticeable problems: there are no spaces between words or numbers, and the letters are italicized and more spaced out than normal. Both issues are simply artifacts of the maths mode, in that it treats it as a mathematical expression: spaces are ignored (LaTeX spaces mathematics according to its own rules), and each character is a separate element (so are not positioned as closely as normal text).

There are a number of ways that text can be added properly. The typical way is to wrap the text with the `\text{...}` command⁴⁰ (a similar command is `\mbox{...}`), though this causes problems with subscripts, and has a less descriptive name). Let's see what happens when the above equation code is adapted:

```
\[
50 \text{apples} \times 100 \text{apples}
= \text{lots of apples}^2
\]
```

50apples × 100apples = lots of apples²

40 requires the `amsmath` package

The text looks better. However, there are no gaps between the numbers and the words. Unfortunately, you are required to explicitly add these. There are many ways to add spaces between maths elements, but for the sake of simplicity you may literally add the space character in the affected `\text(s)` itself (just before the text.)

```
\[ 50 \text{ apples} \times 100 \text{ apples}
= \text{lots of apples}^2
\]
```

50 apples \times 100 apples = lots of apples²

17.11.1. Formatted text

Using the `\text` is fine and gets the basic result. Yet, there is an alternative that offers a little more flexibility. You may recall the introduction of font formatting commands⁴¹, such as `\textrm`, `\textit`, `\textbf`, etc. These commands format the argument accordingly, e.g., `\textbf{bold text}` gives **bold text**. These commands are equally valid within a maths environment to include text. The added benefit here is that you can have better control over the font formatting, rather than the standard text achieved with `\text`.

```
\[
50 \textrm{ apples} \times 100
\textbf{ apples} = \textit{lots of apples}^2
\]
```

41 Chapter 43.9.1 on page 1093

50 apples \times 100 apples = lots of apples²

17.12. Formatting mathematics symbols

We can now format text; what about formatting mathematical expressions? There are a set of formatting commands very similar to the font formatting ones just used, except that they are specifically aimed at text in math mode (requires `amstext`)

LaTeX command	Sample	Description	Common use
<code>\mathnormal</code>	<code>\mathnormal{ABCDEFabcdef123456}</code>	the default math font	most mathematical notation
<code>\mathrm</code>	<code>\mathrm{. . . }ABCDEFabcdef123456</code>	this is the default or normal font, unitalicised	units of measurement, one word functions
<code>\mathit</code>	<code>\mathit{. . . }ABCDEFabcdef123456</code>	italicised font	
<code>\mathbf</code>	<code>\mathbf{. . . }ABCDEFabcdef123456</code>	bold font	vectors

LaTeX command	Sample	Description	Common use
<code>\mathsf{...}</code>	<code>\ABCDEFabcdef123456</code>	Sans-serif ⁴²	
<code>\mathtt{...}</code>	<code>\ABCDEFabcdef123456</code>	Monospace (fixed-width) font ⁴³	

42 <http://en.wikipedia.org/wiki/sans-serif>

43 <http://en.wikipedia.org/wiki/Monospace%20font>

LaTeX command	Sample	Description	Common use
<code>\mathfrak{}</code>	<code>\mathfrak{.123456}</code> . 1 2 3 4 5 6	Fraktur ⁴⁹	Almost canonical font for Lie algebras, with superscript used to denote New Testament papyri ⁵⁰ , ideals ⁵¹ in ring theory
<code>\mathbb{}</code>	<code>\mathbb{.123456}</code> . 1 2 3 4 5 6	Blackboard bold ⁵³	Used to denote special sets (e.g. real numbers)

48 **requires amsfonts or amssymb packages**

49 <http://en.wikipedia.org/wiki/Fraktur%20%28script%29>

50 <http://en.wikipedia.org/wiki/List%20of%20New%20Testament%20papyri>

51 <http://en.wikipedia.org/wiki/Ideal%20%28ring%20theory%29>

52 **requires amsfonts or amssymb packages**

53 <http://en.wikipedia.org/wiki/Blackboard%20bold>

LaTeX command	Sample	Description	Common use
<code>\mathscr{. . }</code> ⁵⁴		Script ⁵⁵	An alternative font for categories and sheaves.

The maths formatting commands can be wrapped around the entire equation, and not just on the textual elements: they only format letters, numbers, and uppercase Greek, and the rest of the maths syntax is ignored.

To bold lowercase Greek or other symbols use the `\boldsymbol` command⁵⁶; this will only work if there exists a bold version of the symbol in the current font. As a last resort there is the `\pmb` command⁵⁷ (poor mans bold): this prints multiple versions of the character slightly offset against each other

54 `require mathrsfs` package

55 <http://en.wikipedia.org/wiki/Script%20%28typesetfaces%29>

56 requires the `amsmath` package

57 requires the `amsmath` package

```
\[
\boldsymbol{\beta} = (\beta_1, \beta_2, \dots, \beta_n)
\]
```

$$\boldsymbol{\beta} = (\beta_1, \beta_2, \dots, \beta_n)$$

To change the size of the fonts in math mode, see Changing font size⁵⁸.

17.12.1. Accents

So what to do when you run out of symbols and fonts? Well the next step is to use accents:

a'	a''
\hat{a}	\bar{a}
\grave{a}	\acute{a}
\dot{a}	\ddot{a}
\not{a}	\mathring{a}
\overrightarrow{ABAB}	\overleftarrow{ABAB}
a'''	a''''
\overline{aaa}	\check{a}
\breve{a}	\vec{a}
$\dot{\dot{\dot{a}}}$ ⁵⁹	$\dot{\dot{\dot{\dot{a}}}}$ ⁶⁰
\widehat{AAA}	\widetilde{AAA}
\tilde{a}	

59 requires the `amsmath` package

60 requires the `amsmath` package

17.13. Plus and minus signs

Latex deals with the + and - signs in two possible ways. The most common is as a binary operator. When two maths elements appear either side of the sign, it is assumed to be a binary operator, and as such, allocates some space either side of the sign. The alternative way is a sign designation. This is when you state whether a mathematical quantity is either positive or negative. This is common for the latter, as in maths, such elements are assumed to be positive unless a - is prefixed to it. In this instance, you want the sign to appear close to the appropriate element to show their association. If you put a + or a - with nothing before it but you want it to be handled like a binary operator you can add an *invisible* character before the operator using `{ }`. This can be useful if you are writing multiple-line formulas, and a new line could start with a = or a +, for example, then you can fix some strange alignments adding the invisible character where necessary.

A plus-minus sign used for uncertainty is written as:

`\l`
`\pm`
`\j`

±

17.14. Controlling horizontal spacing

LaTeX is obviously pretty good at typesetting math—it was one of the chief aims of the core TeX system that LaTeX extends. However, it can't always be relied upon to accurately interpret formulas in the way you did. It has to make certain assumptions when there are ambiguous expressions. The result tends to be slightly incorrect horizontal spacing. In these events, the output is still satisfactory, yet any perfectionists will no doubt wish to *fine-tune*

their formulas to ensure spacing is correct. These are generally very subtle adjustments.

There are other occasions where LaTeX has done its job correctly, but you just want to add some space, maybe to add a comment of some kind. For example, in the following equation, it is preferable to ensure there is a decent amount of space between the math and the text.

```
\[
  f(n) = \left\{
    \begin{array}{l}
      n/2 & \text{\quad \text{if } \$n\$ is even} \\
      -(n+1)/2 & \text{\quad \text{if } \$n\$ is odd}
    \end{array}
  \right.
```

$$f(n) = \begin{cases} n/2 & \text{if } n \text{ is even} \\ -(n+1)/2 & \text{if } n \text{ is odd} \end{cases}$$

This code produces errors with MikTeX 2.9 and does not yield the results seen on the right. Use `\texttrm` instead of just `\text`.

(Note that this particular example can be expressed in more elegant code by the `cases` construct provided by the `amsmath` package described in Advanced Mathematics⁶¹ chapter.)

LaTeX has defined two commands that can be used anywhere in documents (not just maths) to insert some horizontal space. They are `\quad` and `\qquad`

A `\quad` is a space equal to the current font size. So, if you are using an 11pt font, then the space provided by `\quad` will also be 11pt (horizontally, of course.) The `\qquad` gives twice that amount. As you can see from the code

from the above example, `\quads` were used to add some separation between the maths and the text.

OK, so back to the fine tuning as mentioned at the beginning of the document. A good example would be displaying the simple equation for the indefinite integral of y with respect to x :

$$\int y dx$$

If you were to try this, you may write:

```
\[ \int y \mathrm{d}x \]
```

$$\int y dx$$

However, this doesn't give the correct result. LaTeX doesn't respect the white-space left in the code to signify that the y and the dx are independent entities. Instead, it lumps them altogether. A `\quad` would clearly be overkill in this situation—what is needed are some small spaces to be utilized in this type of instance, and that's what LaTeX provides:

Command	Description	Size
<code>\,</code>	small space	3/18 of a quad
<code>\:</code>	medium space	4/18 of a quad
<code>\;</code>	large space	5/18 of a quad
<code>\!</code>	negative space	-3/18 of a quad

NB you can use more than one command in a sequence to achieve a greater space if necessary.

So, to rectify the current problem:

`\[\int y \, \mathrm{d}x \]`

$\int y dx$

`\[\int y \! : \, \mathrm{d}x \]`

$\int y dx$

```
\[ \int y \; ; \; \mathrm{d}x \]
```

$\int y dx$

The negative space may seem like an odd thing to use, however, it wouldn't be there if it didn't have *some* use! Take the following example:

```

\[
\left(
\begin{array}{c}
n \\
r
\end{array}
\right) = \frac{n!}{r!(n-r)!}
\]

```

$$\binom{n}{r} = \frac{n!}{r!(n-r)!}$$

The matrix-like expression for representing binomial coefficients is too padded. There is too much space between the brackets and the actual contents within. This can easily be corrected by adding a few negative spaces after the left bracket and before the right bracket.

```

\left( \left[ \begin{array}{c} n \\ r \end{array} \right] = \frac{n!}{r!(n-r)!} \right)

```

$$\binom{n}{r} = \frac{n!}{r!(n-r)!}$$

In any case, adding some spaces manually should be avoided whenever possible: it makes the source code more complex and it's against the basic principles of a What You See is What You Mean approach. The best thing to do is to define some commands using all the spaces you want and then, when you use your command, you don't have to add any other space. Later, if you change your mind about the length of the horizontal space, you can easily change it

modifying only the command you defined before. Let us use an example: you want the d of a dx in an integral to be in roman font and a small space away from the rest. If you want to type an integral like $\int x \, dx$, you can define a command like this:

```
\newcommand{\dd}{\; \mathrm{d}}
```

in the preamble of your document. We have chosen `\dd` just because it reminds the "d" it replaces and it is fast to type. Doing so, the code for your integral becomes `\int x \, dd x`. Now, whenever you write an integral, you just have to use the `\dd` instead of the "d", and all your integrals will have the same style. If you change your mind, you just have to change the definition in the preamble, and all your integrals will be changed accordingly.

17.15. Advanced Mathematics: AMS Math package

The AMS (American Mathematical Society⁶²) mathematics package is a powerful package that creates a higher layer of abstraction over mathematical LaTeX language; if you use it it will make your life easier. Some commands `amsmath` introduces will make other plain LaTeX commands obsolete: in order to keep consistency in the final output you'd better use `amsmath` commands whenever possible. If you do so, you will get an elegant output without worrying about alignment and other details, keeping your source code readable. If you want to use it, you have to add this in the preamble:

```
\usepackage{amsmath}
```

62 <http://en.wikipedia.org/wiki/American%20Mathematical%20Society>

17.15.1. Introducing text and dots in formulas

`amsmath` defines also the `\dots` command, that is a generalization of the existing `\ldots`. You can use `\dots` in both text and math mode and LaTeX will replace it with three dots "... " but it will decide according to the context whether to put it on the bottom (like `\ldots`) or centered (like `\cdots`).

17.15.2. Dots

LaTeX gives you several commands to insert dots in your formulae. This can be particularly useful if you have to type big matrices omitting elements. First of all, here are the main dots-related commands LaTeX provides:


Code	Output	Comment
------	--------	---------

Code	Output	Comment
<code>\dots</code>	...	generic dots, to be used in text (outside formulae as well). It automatically manages whitespaces before and after itself according to the context, it's a higher level command.
<code>\ldots</code>	...	the output is similar to the previous one, but there is no automatic whitespace management; it works at a lower level.

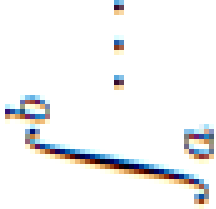

Code	Output	Comment
<code>\cdots</code>	\cdots	These dots are centered relative to the height of a letter. There is also the binary multiplication operator, <code>\cdot</code> , mentioned below.
<code>\vdots</code>	\vdots	vertical dots
<code>\ddots</code>	\ddots	diagonal dots
<code>\iddots</code>		inverse diagonal dots (requires the <code>mathdots</code> package)

Code	Output	Comment
<code>\hdotsfor{n}</code>	to be used in matrices, it creates a row of dots spanning n columns.

Instead of using `\ldots` and `\cdots`, you should use the semantically oriented commands. It makes it possible to adapt your document to different conventions on the fly, in case (for example) you have to submit it to a publisher who insists on following house tradition in this respect. The default treatment for the various kinds follows American Mathematical Society conventions.

Code	Output	Comment
<code>A_1, A_-2, \dotsc,</code>	 <p>Figure 58</p>	for "dots with commas"

Code	Output	Comment
A_1+\dotsb+A_N	$A_1 + \dots + A_N$ <p>Figure 59</p>	for "dots with binary operators/relations"
A_1 \dotsm A_N	$A_1 \dots A_N$ <p>Figure 60</p>	for "multiplication dots"

Code	Output	Comment
$\int_a^b \dots$	 <p style="text-align: center;">Figure 61</p>	for "dots with integrals"
$A_1 \dots A_N$	 <p style="text-align: center;">Figure 62</p>	for "other dots" (none of the above)

17.16. List of Mathematical Symbols

All the pre-defined mathematical symbols from the `\TeX` package are listed below. More symbols are available from extra packages.

Relation Symbols

Sym-bol	Script	Sym-bol	Script	Sym-bol	Script	Sym-bol	Script
$<$	<code>\leq</code>	\geq	<code>\geq</code>	\equiv	<code>\equiv</code>	\approx	<code>\approx</code>
\succ	<code>\succ</code>	\sim	<code>\sim</code>	\perp	<code>\perp</code>	\preceq	<code>\preceq</code>
\simeq	<code>\simeq</code>	\mid	<code>\mid</code>	\ll	<code>\ll</code>	\gg	<code>\gg</code>
\parallel	<code>\parallel</code>	\subset	<code>\subset</code>	\supset	<code>\supset</code>	\approx	<code>\approx</code>
\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>	\cong	<code>\cong</code>	\approx	<code>\approx</code>
\neq	<code>\neq</code>	\smile	<code>\smile</code>	\sqsubseteq	<code>\sqsubseteq</code>	\sqsupset	<code>\sqsupset</code>
\frown	<code>\frown</code>	\in	<code>\in</code>	\ni	<code>\ni</code>	\notin	<code>\notin</code>
\vdash	<code>\vdash</code>	\dashv	<code>\dashv</code>	$<$	<code><</code>	$>$	<code>></code>

Binary Operations

Sym-bol	Script	Sym-bol	Script	Sym-bol	Script	Sym-bol	Script
\pm	<code>\pm</code>	\cap	<code>\cap</code>	\diamond	<code>\diamond</code>	\diamond	<code>\diamond</code>
\mp	<code>\mp</code>	\cup	<code>\cup</code>	Δ	<code>\Delta</code>	\bigtriangleup	<code>\bigtriangleup</code>

\times	<code>\times</code>
\div	<code>\div</code>
$*$	<code>\ast</code>
\star	<code>\star</code>
\dagger	<code>\dagger</code>
\ddagger	<code>\ddagger</code>

\oplus	<code>\oplus</code>
\sqcap	<code>\sqcap</code>
\sqcup	<code>\sqcup</code>
\vee	<code>\vee</code>
\wedge	<code>\wedge</code>
\cdot	<code>\cdot</code>

∇	<code>\nabla</code>
\triangleleft	<code>\triangleleft</code>
\triangleright	<code>\triangleright</code>
\bigcirc	<code>\bigcirc</code>
\bullet	<code>\bullet</code>
\cdot	<code>\cdot</code>
\wr	<code>\wr</code>

\bigtriangledown	<code>\bigtriangledown</code>
\triangleleft	<code>\triangleleft</code>
\triangleright	<code>\triangleright</code>
\bigcirc	<code>\bigcirc</code>
\bullet	<code>\bullet</code>
\cdot	<code>\cdot</code>
\wr	<code>\wr</code>

Set and/or Logic Notation

Symbol	Script
\exists	<code>\exists</code>
\nexists	<code>\nexists</code>
\forall	<code>\forall</code>
\neg	<code>\neg</code>
\in	<code>\in</code>
\notin	<code>\notin</code>
\ni	<code>\ni</code>
\wedge	<code>\wedge</code>

Symbol	Script
\rightarrow	<code>\rightarrow</code>
\Rightarrow	<code>\implies</code>
\Rightarrow	<code>\Rightarrow</code> (preferred for implication)
\leftrightarrow	<code>\leftrightarrow</code>
\Leftrightarrow	<code>\iff</code>
\Leftrightarrow	<code>\Leftrightarrow</code> (preferred for equivalence (iff))
\top	<code>\top</code>
\bot	<code>\bot</code>

\vee	<code>\lor</code>	\emptyset and \emptyset	<code>\emptysetset</code> and <code>\varnothing</code>
--------	-------------------	-----------------------------	--

Delimiters

Symbol	Script	Symbol	Script	Symbol	Script	Symbol
				/	/	\
{	\{	}	\}	<	\langle	>
↑	\uparrow	↑	\Uparrow	[\lceil]
↓	\downarrow	↓	\Downarrow]	\rfloor]

Greek Letters

Symbol	Script	Symbol	Script
A and α	\Alpha and \alpha	N and ν	\Nu and \nu
B and β	\Beta and \beta	Ξ and ξ	\Xi and \xi
Γ and γ	\Gamma and \gamma	and o	\Omicron and \omicron
Δ and δ	\Delta and \delta	Π, π and ϖ	\Pi, \pi and \varpi

E, ε and ϵ	<code>\Epsilon,</code> <code>\epsilon</code> and <code>\varepsilon</code>		
Z and ζ	<code>\Zeta</code> and <code>\zeta</code>		
H and η	<code>\Eta</code> and <code>\eta</code>		
Θ , θ and ϑ	<code>\Theta</code> , <code>\theta</code> and <code>\vartheta</code>		
I and i	<code>\Iota</code> and <code>\iota</code>		
K and κ	<code>\Kappa</code> and <code>\kappa</code>		
Λ and λ	<code>\Lambda</code> and <code>\lambda</code>		
M and μ	<code>\Mu</code> and <code>\mu</code>		
		P, ρ and ρ	<code>\Rho</code> , <code>\rho</code> and <code>\varrho</code>
		Σ , σ and ς	<code>\Sigma</code> , <code>\sigma</code> and <code>\varsigma</code>
		T and τ	<code>\Tau</code> and <code>\tau</code>
		Υ and υ	<code>\Upsilon</code> and <code>\upsilon</code>
		Φ , ϕ , and φ	<code>\Phi</code> , <code>\phi</code> , and <code>\varphi</code>
		X and χ	<code>\Chi</code> and <code>\chi</code>
		Ψ and ψ	<code>\Psi</code> and <code>\psi</code>
		Ω and ω	<code>\Omega</code> and <code>\omega</code>

Other symbols

Sym- bol	Script	Sym- bol	Script	Sym- bol	Script
∂	<code>\partial</code>	ι	<code>\imath</code>	\Re	<code>\Re</code>
	<code>\partial</code>		<code>\imath</code>	∇	<code>\nabla</code>
				∇_a	<code>\nabla_a</code>

\bar{d}	<code>\eth</code>		\mathcal{S}	<code>\Im</code>	<code>\Box</code>
\hbar	<code>\hbaraf</code>	ℓ	ρ	<code>\wp</code>	<code>\infty</code>
					<code>\infty</code>

Trigonometric Functions

Sym-bol	Script	Sym-bol	Script	Sym-bol	Script	Sym-bol
\sin	<code>\sin</code>	\arcsin	<code>\arcsin</code>	\sinh	<code>\sinh</code>	\sec
\cos	<code>\cos</code>	\arccos	<code>\arccos</code>	\cosh	<code>\cosh</code>	\csc
\tan	<code>\tan</code>	\arctan	<code>\arctan</code>	\tanh	<code>\tanh</code>	
\cot	<code>\cot</code>		<code>\arccot</code>	\coth	<code>\coth</code>	

17.17. Summary

As you begin to see, typesetting math can be tricky at times. However, because LaTeX provides so much control, you can get professional quality mathematics typesetting with relatively little effort (once you've had a bit of practice, of course!). It would be possible to keep going and going with math topics

because it seems potentially limitless. However, with this tutorial, you should be able to get along sufficiently.

17.18. Notes

17.19. Further reading

- [meta:Help:Displaying a formula](#)⁶³: Wikimedia uses a subset of LaTeX commands.

⁶³ <http://en.meta.org/wiki/Help%3ADisplaying%20a%20formula>

17.20. External links

- LaTeX maths symbols⁶⁴
- detexify⁶⁵: applet for looking up LaTeX symbols by handwriting them
- [ftp://ftp.ams.org/pub/tex/doc/amsmath/amslidoc.pdf amsmath documentation]
- LaTeX - The Student Room⁶⁶
- The Comprehensive LaTeX Symbol List⁶⁷

pl:LaTeX/Matematyka⁶⁸

⁶⁴ <http://www.artofproblemsolving.com/wiki/index.php/LaTeX>:

Symbols

- ⁶⁵ <http://detexify.kirelabs.org>
- ⁶⁶ <http://www.thestudentroom.co.uk/wiki/LaTeX>
- ⁶⁷ <http://www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-letter.pdf>
- ⁶⁸ <http://pl.wikibooks.org/wiki/LaTeX%2FMatematyka>

18. Advanced Mathematics

This page outlines some more advanced uses of mathematics markup using LaTeX. In particular it makes heavy use of the AMS-LaTeX packages supplied by the American Mathematical Society¹.

¹ <http://en.wikipedia.org/wiki/American%20Mathematical%20Society>

18.1. Equation numbering

The `equation` environment automatically numbers your equation.

```
\begin{equation}
f(x) = (x+a)(x+b)
\end{equation}
```

$$f(x) = (x+a)(x+b) \quad (1)$$

You can also use the `\label` and `\ref` (or `\eqref` from the `amsmath` package) commands to label and reference equations, respectively. For equation number 1, `\ref` results in 1 and `\eqref` results in (1):

```
\begin{equation} \label{eq:someequation}
5^2 - 5 = 20
\end{equation}
```

this references the equation `\ref{eq:someequation}`.

$$5^2 - 5 = 20 \quad (1)$$

this references the equation 1.


```
\begin{equation} \label{eq:erl}
a = bq + r
\end{equation}
```

where `\eqref{eq:erl}` is true if `a` and `b` are integers with `$b \neq c$`.

$$a = bq + r \quad (1)$$

where (1) is true if a and b are integers with $b \neq c$.

Further information is provided in the labels and cross-referencing² chapter.

To have the enumeration follow from your section or subsection heading, you must use the `amsmath` package or use `AMS` class documents. Then enter

```
\numberwithin{equation}{section}
```

to the preamble to get enumeration at the section level or

```
\numberwithin{equation}{subsection}
```

to have the enumeration go to the subsection level.

```

\documentclass[12pt]{article}
\usepackage{amsmath}
\numberwithin{equation}{subsection}
\begin{document}
\section{First Section}

\subsection{A subsection}
\begin{equation}
L' = \{L\}\{\sqrt{1-\frac{v^2}{c^2}}\}<!-- --><!-- -->
\end{equation}
\end{document}

```

$$L' = L\sqrt{1 - \frac{v^2}{c^2}} \quad (1.1.1)$$

If the style you follow requires putting dots after ordinals (as it is required at least in Polish typography) the `\numberwithin{equation}{subsection}` command in preamble will result in the equation number in the above example to be rendered in this way: (1.1..1).

To remove the duplicate dot, add following command immediately after `\numberwithin{equation}{subsection}` :

```
\renewcommand{\theequation}{\thesection\arabic{equation}}
```

For numbering scheme using `\numberwithin{equation}{subsection}` use:

```
\renewcommand{\theequation}{\thesubsection\arabic{equation}}
```

in the preamble of the document.

Note: Though it may look like the `\renewcommand` works by itself, it won't reset the equation number with each new section. It must be used together with manual equation number resetting after each new section beginning or with the much cleaner `\numberwithin`.

18.1.1. Subordinate equation numbering

To number subordinate equations in a numbered equation environment, place the part of document containing them in a `subequations` environment:

```

\begin{subequations}
Maxwell's equations:
\begin{align}
\mathbf{B}' &= -\nabla \times \mathbf{E}, \\
\mathbf{E}' &= \nabla \times \mathbf{B} - 4\pi \mathbf{j},
\end{align}
\end{subequations}

```

Maxwell's equations:

$$\mathbf{B}' = -\nabla \times \mathbf{E}, \quad (1.1a)$$

$$\mathbf{E}' = \nabla \times \mathbf{B} - 4\pi \mathbf{j}, \quad (1.1b)$$

Referencing of subordinate equations can be done in two methods. Adding a label after the `\begin{subequations}` command, will reference to the main equation (1.1 the case presented). Adding a label after at the end of each line, before the `\` command, will reference to the sub-equation (1.1a or

1.1b the case presented). It is possible to add both labels in case both types of references are needed.

18.2. Vertically aligning displayed mathematics

An often encountered problem with displayed environments (`displaymath` and `equation`) is the lack of any ability to span multiple lines. While it is possible to define lines individually, these will not be aligned.

18.2.1. Above and below

The `\overset` and `\underset` commands³ typeset symbols above and below expressions. Without AmsTeX the same result of `\overset` can be

3 requires the `amsmath` package

obtained with `\stackrel{!}{=}`. This can be particularly useful for creating new binary relations:

```
\[
  A \overset{!}{=} B; A \stackrel{!}{=} B
\]
```

$A \overset{!}{=} B$; $A \stackrel{!}{=} B$

or to show usage of L'Hôpital's rule⁴:

⁴ http://en.wikipedia.org/wiki/L%27H%F4pital%27s_rule

```

\l
\lim_{x\to 0}{\frac{e^x-1}{2x}}<!-- -->
\overset{\left[\frac{0}{0}\right]}{\underset{\mathrm{H}}{<!-- -->}}{\frac{e^x-1}{2x}}<!-- -->
\lim_{x\to 0}{\frac{e^x}{2}}<!-- -->
\l

```

$$\lim_{x \rightarrow 0} \frac{e^x - 1}{2x} \underset{\text{H}}{\overset{\left[\frac{0}{0}\right]}{=}} \frac{e^x}{2} = \frac{1}{2}$$

It's convenient to define a new operator that will set the equal sign with H and provided fraction:

```

\newcom
mand{\Heg}[1]{\overset{\left[#1\right]}{\underset{\mathrm{H}}{=}}}

```

to reduce above example to:

```

\l

```



```

\lim_{x\to 0}{\frac{e^x-1}{x}}<!-- -->
\Heq{\frac{0}{0}}<!-- -->
\lim_{x\to 0}{\frac{e^x}{x^2}}<!-- -->={\frac{1}{2}}<!-- -->
\]

```

If the purpose is to make comments on particular parts of an equation, the `\overbrace` and `\underbrace` commands may be more useful, however they have a different syntax:

```

\[
z = \overbrace{
  \underbrace{x}_{\text{real}} +
  \underbrace{iy}_{\text{imaginary}}
}^{\text{complex number}}
\]

```

$$z = \underbrace{x}_{\text{real}} + \underbrace{iy}_{\text{imaginary}}$$

Sometimes the comments are longer than the formula being commented on, which can cause spacing problems. These can be removed using the `\mathclap` command⁵:

```
\[
y = a + f(\underbrace{b x}_{\ge 0 \text{ by assumption}}<!-- -->)
= a + f(\underbrace{b x}_{\mathclap{\ge 0 \text{ by assumption}}<!-- -->}<!-- -->)
\]
```

$$y = a + f(\underbrace{bx}_{\ge 0 \text{ by assumption}}) = a + f(\underbrace{bx}_{\ge 0 \text{ by assumption}})$$

Figure 63

5 requires the `mathtools` package

Alternatively, to use brackets instead of braces use `\underbracket` and `\overbracket` commands⁶:

```
\[
z = \overbracket[3pt]{
  \underbracket{x}_ {\text{real}}<!----> +
  \underbracket[0.5pt][7pt]{iy}_ {\text{imaginary}}<!---->}
  ^{\text{complex number}}<!---->}
\]
```

$$z = \overbrace{x + iy}^{\text{complex number}}$$

$\underbrace{\hspace{1.5cm}}_{\text{real}}$ $\underbrace{\hspace{1.5cm}}_{\text{imaginary}}$

Figure 64

⁶ requires the `mathtools` package

The optional arguments set the rule thickness and bracket height respectively:

```
\underbracket[rule thickness][bracket height]{argument}_{text below}
```

The `\xleftarrow` and `\xrightarrow` commands⁷ produce arrows which extend to the length of the text. Yet again, the syntax is different: the optional argument (using `[&]`) specifies the subscript, and the mandatory argument (using `{ & }`) specifies the superscript (this can be left empty).

```
\[
A \xleftarrow{\text{this way}}<!-- --> B
  \xrightarrow[\text{or that way}]{} C
\]
```

$$A \xleftarrow{\text{this way}} B \xrightarrow{\text{or that way}} C$$

⁷ requires the `amsmath` package

For more extensible arrows, you must use `mathtools` package:

```
\[
a \xleftarrow[under]{over} b\\
%
A \xLeftarrow[under]{over} B\\
%
B \xrightarrow[under]{over} C\\
%
C \xLeftrightarrow[under]{over} D\\
%
D \xhookleftarrow[under]{over} E\\
%
E \xhookrightarrow[under]{over} F\\
%
F \xmapsto[under]{over} G\\
\]
```

$$a \begin{array}{c} \xleftarrow{\text{over}} \\ \xrightarrow{\text{under}} \end{array} b$$

$$A \begin{array}{c} \xleftarrow{\text{over}} \\ \xrightarrow{\text{under}} \end{array} B$$

$$B \begin{array}{c} \xrightarrow{\text{over}} \\ \xrightarrow{\text{under}} \end{array} C$$

$$C \begin{array}{c} \xleftarrow{\text{over}} \\ \xrightarrow{\text{under}} \end{array} D$$

$$D \begin{array}{c} \xleftarrow{\text{over}} \\ \xrightarrow{\text{under}} \end{array} E$$

$$E \begin{array}{c} \xleftarrow{\text{over}} \\ \xrightarrow{\text{under}} \end{array} F$$

$$F \begin{array}{c} \xrightarrow{\text{over}} \\ \xrightarrow{\text{under}} \end{array} G$$

Figure 65

and for harpoons:

\[
H \xrightarrow[\text{under}]{\text{over}} I \\ %
I \xrightarrow[\text{under}]{\text{over}} J \\ %
J \xleftarrow[\text{under}]{\text{over}} K \\ %
K \xleftarrow[\text{under}]{\text{over}} L \\ %
L \xrightarrow[\text{under}]{\text{over}} M \\ %
M \xleftarrow[\text{under}]{\text{over}} N
\]

$$H \xrightarrow[\text{under}]{\text{over}} I$$
$$I \xrightarrow[\text{under}]{\text{over}} J$$
$$J \xrightarrow[\text{under}]{\text{over}} K$$
$$K \xrightarrow[\text{under}]{\text{over}} L$$
$$L \xrightarrow[\text{under}]{\text{over}} M$$
$$M \xrightarrow[\text{under}]{\text{over}} N$$

Figure 66

18.2.2. align and align*

The `align` and `align*` environments⁸ are used for arranging equations of multiple lines. As with matrices and tables, `\` specifies a line break, and `&` is used to indicate the point at which the lines should be aligned.

The `align*` environment is used like the `displaymath` or `equation*` environment:

```
\begin{align*}
f(x) &= (x+a)(x+b) \\
&= x^2 + (a+b)x + ab
\end{align*}
```

$$\begin{aligned} f(x) &= (x+a)(x+b) \\ &= x^2 + (a+b)x + ab \end{aligned}$$

8 requires the `amsmath` package

To force numbering on a specific line, use the `\tag{...}` command before the `linebreak`.

The `align` is similar, but automatically numbers each line like the equation environment. Individual lines may be referred to by placing a `\label{...}` before the `linebreak`. The `\nonumber` or `\notag` command can be used to suppress the number for a given line:

```
\begin{align}
f(x) &= x^4 + 7x^3 + 2x^2 \nonumber \\
&\quad {} + 10x + 12 \\
\end{align}
```

$$f(x) = x^4 + 7x^3 + 2x^2 + 10x + 12 \quad (3)$$



Notice that we've added some indenting on the second line. Also, we need to insert the double braces `{ }` before the `+` sign because otherwise latex won't create the correct spacing after the `+` sign. The reason for this is that without the braces, latex interprets the `+` sign as a unary operator, instead of the binary operator that it really is.

More complicated alignments are possible. The following example illustrates the alignment rule of `align*`:

```
\begin{align*}
a & b & c & d & e & f & g & h & i & j \\
\end{align*}
```

ab cd ef gh ij

18.2.3. Braces spanning multiple lines

Additional `&s` on a single line will specify multiple "equation columns", each of which is aligned. If you want a brace to continue across a new line, do the following:

```
\begin{align}
f(x) &= \pi \left\{ x^4 + 7x^3 + 2x^2 \right. \nonumber \\
&\quad \left. \right\} + 10x + 12 \left. \right\} \\
\end{align}
```

$$f(x) = \pi \left\{ x^4 + 7x^3 + 2x^2 \right. \\ \left. + 10x + 12 \right\} \quad (4)$$

In this construction, the sizes of the left and right braces are not automatically equal, in spite of the use of `\left\{` and `\right\}`. This is because each

line is typeset as a completely separate equation —notice the use of `\right·` and `\left·` so there are no unpaired `\left` and `\right` commands within a line. (These aren't needed if the formula is on one line.) You can control the size of the braces manually with the `\big`, `\Big`, `\bigg`, `\Bigg` commands.

Or you can replicate the height of the taller equation in the other using the `\vphantom` command:

```
\begin{align}
A &= \left(\int_t XXX \right.\nonumber\!
&\quad \left.\vphantom{\int_t} YYY \dots \right)
\end{align}
```

$$A = \left(\int_t XXX \right. \left. YYY \dots \right) \quad (5)$$

18.2.4. The cases environment

The cases environment⁹ allows the writing of piecewise functions:

```
\[
u(x) =
\begin{cases}
\exp{x} & \text{if } x \geq 0 \\
1 & \text{if } x < 0
\end{cases}
\]
```

$$u(x) = \begin{cases} \exp x & \text{if } x \geq 0 \\ 1 & \text{if } x < 0 \end{cases}$$

⁹ requires the `amsmath` package

Just like before, you don't have to take care of definition or alignment of columns, LaTeX will do it for you.

Unfortunately, it sets the internal math environment to text style, leading to such result:

$$a = \left\{ \int x dx \right. \\ \left. b^2 \right.$$

To force display style for equations inside this construct, use `dcases environment`¹⁰:

¹⁰ requires the `mathtools` package

```

\[
a = \begin{dcases}
\int x \backslash, dx \\
b^2
\end{dcases}
\]

```

$$a = \left\{ \int x dx \right. \\ \left. b^2 \right.$$

Often the second column consists mostly of normal text, to set it in normal roman font of the document use `dcases*` environment¹¹:

¹¹ requires the `mathtools` package


```

\[
f(x) = \begin{dcases*}
x & \text{when } \mathbf{\$x\$} \text{ is even} \\
-x & \text{when } \mathbf{\$x\$} \text{ is odd}
\end{dcases*}
\]

```

$$f(x) = \begin{cases} x & \text{when } x \text{ is even} \\ -x & \text{when } x \text{ is odd} \end{cases}$$

18.2.5. Other environments

Although `align` and `align*` are the most useful, there are several other environments which may also be of interest:

Environment name	Description	Notes
------------------	-------------	-------

Environment name	Description	Notes
<code>eqnarray</code> and <code>eqnarray*</code>	Similar to <code>align</code> and <code>align*</code>	Not recommended since spacing is inconsistent
<code>multline</code> and <code>multline*</code> ¹²	First line left aligned, last line right aligned	Equation number aligned vertically with first line and not centered as with other environments.
<code>gather</code> and <code>gather*</code> ¹³	Consecutive equations without alignment	

¹² requires the `amsmath` package

¹³ requires the `amsmath` package

Environment name	Description	Notes
<code>flalign</code> and <code>flalign*</code> ¹⁴	Similar to <code>align</code> , but left aligns first equation column, and right aligns last column	
<code>alignat</code> and <code>alignat*</code> ¹⁵	Takes an argument specifying number of columns. Allows to control explicitly the horizontal space between equations	You can calculate the number of columns by counting & characters in a line, adding 1 and dividing the result by 2

There are also few environments that don't form a math environment by themselves and can be used as building blocks for more elaborate structures:

¹⁴ requires the `amsmath` package

¹⁵ requires the `amsmath` package

Math environment name	Description
gathered ¹⁶	Allows to gather few equations to be set under each other and assigned a single equation number
split ¹⁷	Similar to align*, but used inside another displayed mathematics environment
aligned ¹⁸	Similar to align, to be used inside another mathematics environment.

16 requires the `amsmath` package

17 requires the `amsmath` package

18 requires the `amsmath` package

Math environment name	Description
alignedat ¹⁹	Similar to alignat, and just as it, takes an additional argument specifying number of columns of equations to set.

For example:

```

\begin{equation}
\left. \begin{aligned}
& B' &= & -\partial \times E, \\
& E' &= & \partial \times B - 4\pi j,
\end{aligned} \right\}
\end{equation}

```

¹⁹ requires the `amsmath` package

$$\left. \begin{aligned} \mathbf{B}' &= -\partial \times \mathbf{E}, \\ \mathbf{E}' &= \partial \times \mathbf{B} - 4\pi \mathbf{j}, \end{aligned} \right\} \text{Maxwell's equations} \quad (1.1)$$

```
\begin{alignat}{2}
\sigma_1 &= x + y & \quad \sigma_2 &= \frac{x}{y} \\
\sigma_1' &= \frac{\partial x}{\partial x} + \frac{\partial y}{\partial y} & \quad \sigma_2' &= \frac{\partial^2 x}{\partial x \partial y}
\end{alignat}
```

$$\sigma_1 = x + y \quad \sigma_2 = \frac{x}{y} \quad (1)$$

$$\sigma_1' = \frac{\partial x + y}{\partial x} \quad \sigma_2' = \frac{\partial^2 x}{\partial x \partial y} \quad (2)$$

18.3. Indented Equations

In order to indent an equation, you can set `fleqn` in the document class and then specify a certain value for `\mathindent` variable:

```
\documentclass[a4paper,fleqn]{report}
\usepackage{amsmath}
\setlength{\mathindent}{1cm}
\begin{document}
\noindent Euler's formula is given below:
\begin{equation*}
e^{ix} = \cos{x} + i \sin{x}
\end{equation*}
\noindent This is a very important formula.
\end{document}
```

Euler's formula is given below:

$$e^{ix} = \cos x + i \sin x$$

This is a very important formula.

Figure 67

18.4. Page breaks in math environments

To suggest LaTeX a page break inside one of `amsmath` environments you may use the `\displaybreak` command before line break. Just like with

`\pagebreak`, `\displaybreak` can take an optional argument between 0 and 4 denoting the level of desirability of a page break in specific break. While 0 means "it is permissible to break here", 4 forces a break. No argument means the same as 4.

Alternatively, you may enable automatic page breaks in math environments with `\allowdisplaybreaks`. It can too have an optional argument denoting permissiveness of page breaks in equations. Similarly, 1 means "allow page breaks but avoid them" and 4 means "break whenever you want". You can prohibit a page break after a given line using `\|*`.

LaTeX will insert a page break into a long equation if it has additional text added using `\intertext{ }` without any additional commands though.

Specific usage may look like this:

```

\begin{align*}
&\vdots\\
&=12+7 \int_0^2 \\
&\left(
\quad -\frac{1}{4}\left(e^{-4t_1}+e^{4t_1-8}\right)
\right. \\
&\quad \left.\right) \, dt_1 \displaybreak[3] \\
&= 12-\frac{7}{4}\int_0^2 e^{-4t_1}+e^{4t_1-8} \, dt_1 \\
&\vdots \quad \frac{\circ}{\circ} \\
\end{align*}

```

$$\begin{aligned} & \vdots \\ & = 12 + 7 \int_0^2 \left(-\frac{1}{4} (e^{-4t_1} + e^{4t_1-8}) \right) dt_1 \end{aligned}$$

1

$$= 12 - \frac{7}{4} \int_0^2 e^{-4t_1} + e^{4t_1-8} dt_1$$

\vdots

Figure 68

18.5. Boxed Equations

For a single equation or alignment building block, with the tag outside the box, use `\boxed{}`:

```
\begin{equation}
\boxed{x^2+y^2 = z^2}
\end{equation}
```

$$x^2 + y^2 = z^2$$

(1)

Figure 69

If you want the entire line or several equations to be boxed, use a `minipage` inside an `\fbox{}`:

```

\fbbox{
\addtolength{\linewidth}{-2\fbboxsep}%
\addtolength{\linewidth}{-2\fbboxrule}%
\begin{minipage}{\linewidth}
\begin{equation}
x^2+y^2=z^2
\end{equation}
\end{minipage}
}

```

$$x^2 + y^2 = z^2$$

(1)

Figure 70

There is also the `\boxed{ }` which is able to box across alignment marks

```
\begin{align*}
\Aboxed{ f(x) &= \int h(x)\, dx } \\
&= g(x) \\
\end{align*}
```

$$f(x) = \int h(x) dx$$

$$= g(x)$$

Figure 71

18.6. Custom operators

Although many common operators²⁰ are available in LaTeX, sometimes you will need to write your own, for example to typeset the argmax ²¹ operator. The `\operatorname` and `\operatorname*` commands²² display a custom operators, the `*` version sets the underscored option underneath like the `\lim` operator:

```
\[
\operatorname{arg}\,max\}_a f(a)
= \operatorname*{arg}\,max\}_b f(b)
\]
```

20 Chapter 17.4 on page 510

21 <http://en.wikipedia.org/wiki/Arg%20max>

22 requires the `amsmath` package

$$\operatorname{argmax}_a f(a) = \operatorname{argmax}_b f(b)$$

However if the operator is frequently used, it is preferable to keep within the LaTeX ideal of markup to define a new operator. The `\DeclareMathOperator` and `\DeclareMathOperator*` commands²³ are specified in the header of the document:

```
\DeclareMathOperator*{\argmax}{arg\,max}
```

This defines a new command which may be referred to in the body:

```
\[  
  \argmax_c f(c)  
\]
```

23 requires the `amsmath` package

$\arg \max_c f(c)$

18.7. Advanced formatting

18.7.1. Limits

There are defaults for placement of subscripts and superscripts. For example, limits for the `\lim` operator are usually placed below the symbol, like this:

```
\begin{equation}
\lim_{a \to \infty} \tfrac{1}{a}
\end{equation}
```

$$\lim_{a \rightarrow \infty} \frac{1}{a}$$

To override this behavior use the `\nolimits` operator:

```
\begin{equation}
\lim\nolimits_{a \to \infty} \tfrac{1}{a}
\end{equation}
```

$$\lim_{a \rightarrow \infty} \frac{1}{a}$$

A `\lim` in running text (inside `§ . . . §`) will have its limits placed on the side, so that additional leading won't be required. To override this behavior use `\limits` command.

Similarly one can put subscripts under a symbol that usually have them on the side:

```
\begin{equation}
\int_a^b x^2
\end{equation}
```

$$\int_a^b x^2$$

Limits below and under:

```
\begin{equation}
\int\limits_a^b x^2
\end{equation}
```

$$\int_a^b x^2$$

To change default placement in all instances of summation-type symbol to the side add `nosumlimits` option to `amsmath` package. To change placement for integral symbols add `intlimits` to options and `nonamlimits` to change the default for named operators like `det`, `min`, `lim`...

18.7.2. Subscripts and superscripts

While you can place symbols in sub- or superscript in summation style symbols with the above introduced `\nolimits`:

```
\begin{equation}
\sum\nolimits' C_n
\end{equation}
```

$$\Sigma' C_n$$

It's impossible to mix them with typical usage of such symbols:

```
\begin{equation}
\sum_{n=1}\nolimits' C_n
\end{equation}
```

$$\sum'_{n=1} C_n$$

To add both prime and a limit to a symbol, one have to use `\sideset` command:

```
\begin{equation}
\sideset{}{\prime}\sum_{n=1}C_n
\end{equation}
```

$$\sum_{n=1}^{\prime} C_n$$

It's very flexible, for example, to put letters in each corner of the symbol use this command:

```
\begin{equation}  
  \sideset{_{a}^{b}}{_{c}^{d}}\sum  
\end{equation}
```

$${}^b\sum_a^d {}_c$$

If you wish to place them on the corners of an arbitrary symbol, you should use `\fourIdx` from the `fourIdx` package.

18.7.3. Multiline subscripts

To produce multiline subscript use `\substack` command:

```
\begin{equation}
\prod_{\substack{1\leq i \leq n \\ 1\leq j \leq m} <!-->}
M_{i,j}
\end{equation}
```

$$\prod_{\substack{1\leq i\leq n \\ 1\leq j\leq m}} M_{i,j}$$

18.8. Text in aligned math display

To add small interjections in math environments use `\intertext` command:

```
\begin{minipage}{3in}
\begin{align*}
\intertext{If}
A &= \sigma_1 + \sigma_2 \\
B &= \rho_1 + \rho_2 \\
\intertext{then}
C(x) &= e^{\{Ax^2 + \pi\}} + B
\end{align*}
\end{minipage}
```

If

$$A = \sigma_1 + \sigma_2$$

$$B = \rho_1 + \rho_2$$

then

$$C(x) = e^{Ax^2 + \pi} + B$$

Figure 72

Note that usage of this command doesn't change alignment, as would stopping and restarting the `align` environment.

Also in this example, the command `\shortintertext{}` from the `mathtools` package could have been used instead of intertext to reduce the amount of vertical whitespace added between the lines.

18.9. Changing font size

Probably a rare event, but there may be a time when you would prefer to have some control of the size. For example, using text-mode maths, by default a simple fraction will look like this: $\frac{a}{b}$ where as you may prefer to have it displayed larger, like when in display mode, but still keeping it in-line, like this: $\frac{a}{b}$.

A simple approach is to utilize the predefined sizes for maths elements:

Size command	Description
<code>\displaystyle</code>	Size for equations in display mode
<code>\textstyle</code>	Size for equations in text mode
<code>\scriptstyle</code>	Size for first sub/superscripts
<code>\scriptscriptstyle</code>	Size for subsequent sub/superscripts

A classic example to see this in use is typesetting continued fractions (though it's better to use the `\cfrac` command²⁴ described in the Mathematics²⁵ chapter over the method provided below). The following code provides an example.

```
\begin{equation}
x = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + a_4}}} <!----><
\end{equation}
```

$$x = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + a_4}}}$$

As you can see, as the fractions continue, they get smaller (although they will not get any smaller as in this example, they have reached the `\scriptstyle`

²⁴ requires the `amsmath` package

²⁵ Chapter 17.6.1 on page 520

limit). If you wanted to keep the size consistent, you could declare each fraction to use the display style instead, e.g.:

```
\begin{equation}
  x = a_0 + \frac{1}{\displaystyle a_1}
        + \frac{1}{\displaystyle a_2}
        + \frac{1}{\displaystyle a_3 + a_4}<!--><!-->
\end{equation}
```

$$x = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + a_4}}}$$

Another approach is to use the `\DeclareMathSizes` command to select your preferred sizes. You can only define sizes for `\displaystyle`,

`\textstyle`, etc. One potential downside is that this command sets the global maths sizes, as it can only be used in the document preamble.

However, it's fairly easy to use: `\DeclareMathSizes{ds}{ts}{ss}{ss}` where *ds* is the *display size*, *ts* is the *text size*, etc. The values you input are assumed to be point (pt) size.

NB the changes only take place if the value in the first argument matches the current document text size. It is therefore common to see a set of declarations in the preamble, in the event of the main font being changed. E.g.,

```
\DeclareMathSizes{10}{18}{12}{8}   % For size 10 text
\DeclareMathSizes{11}{19}{13}{9}   % For size 11 text
\DeclareMathSizes{12}{20}{14}{10}  % For size 12 text
```

18.10. Forcing `\displaystyle` for all math in a document

Put

```
\everymath{\displaystyle}
```

before `\begin{document}` to force all math to `\displaystyle`.

18.11. Adjusting vertical whitespace around displayed math

There are four parameters which control the vertical whitespace around displayed math:

```
\abovedisplayskip=12pt  
\belowdisplayskip=12pt  
\abovedisplayshortskip=0pt  
\belowdisplayshortskip=7pt
```

Short skips are used if the preceding line ends, horizontally, before the formula. These parameters must be set after `\begin{document}`.

18.12. Notes

19. Labels and Cross-referencing

19.1. Introduction

Another good point of LaTeX is that you can easily reference almost anything that is numbered (sections, figures, formulas), and LaTeX will take care of numbering, updating it whenever necessary. The commands to be used do not depend on what you are referencing, and they are:

```
\label{marker}
```

you give the object you want to reference a *marker*, you can see it like a name.

`\ref{marker}`

you can reference the object you have *marked* before. This prints the number that was assigned to the object.

`\pageref{marker}`

It will print the number of the page where the object is.

LaTeX will calculate the right numbering for the objects in the document; the *marker* you have used to label the object will not be shown anywhere in the document. Then LaTeX will replace the string "`\ref{marker}`" with the right number that was assigned to the object. If you reference a *marker* that does not exist, the compilation of the document will be successful but LaTeX will return a warning:

LaTeX Warning: There were undefined references.

and it will replace `\ref{unknown-marker}` with `"?"` (so it will be easy to find in the document).

As you may have noticed reading how it works, it is a two-step process: first the compiler has to store the labels with the right number to be used for referencing, then it has to replace the `\ref` with the right number. That is why, when you use references, you have to compile your document twice to see the proper output. If you compile it only once, LaTeX will use the older information it collected in previous compilations (that might be outdated), but the compiler will inform you printing on the screen at the end of the compilation:

```
LaTeX Warning: Label(s) may have changed. Rerun
to get cross-references right.
```

Using the command `\pageref{}` you can help the reader to find the referenced object by providing also the page number where it can be found. You could write something like:

```
See figure~\ref{fig:test} on page~\pageref{fig:test}.
```

Since you can use exactly the same commands to reference almost anything, you might get a bit confused after you have introduced a lot of references. It is common practice among LaTeX users to add a few letters to the label to describe *what* you are referencing. Here is an example:

chap:	chapter
sec:	section
fig:	figure
tab:	table
eq:	equation
lst:	code listing

Following this convention, the label of a figure will look like `\label{fig:my_figure}`, etc. You are not obligated to use these prefixes. You can use any string as argument of `\label{...}`, but these prefixes become increasingly useful as your document grows in size.

Another suggestion: try to avoid using numbers within labels. You are better off describing *what* the object is about. This way, if you change the order of the objects, you will not have to rename all your labels and their references.

If you want to be able to see the markers you are using in the output document as well, you can use the `showkeys` package; this can be very useful while developing your document. For more information see the `Packages`¹ section.

19.2. Examples

Here are some practical examples, but you will notice that they are all the same because they all use the same commands.

19.2.1. Sections

1 Greetings

```
\section{Greetings}  
\label{sec:greetings}
```

Hello!

2 Referencing

```
\section{Referencing}
```

I greeted in section 1.

I greeted in section~\ref{sec:greetings} **Figure 73**

You could place the label anywhere in the section; however, in order to avoid confusion, it is better to place it immediately after the beginning of the section. Note how the marker starts with *sec*; as suggested before. The label is then referenced in a different section. The tilde (~) indicates a non-breaking space².

19.2.2. Pictures

You can reference a picture by inserting it in the figure floating environment.

² <http://en.wikipedia.org/wiki/non-breaking%20space>


```
\begin{figure}
\centering
\includegraphics[width=0.5\text
\caption{Close-up of a gull}
\label{gull}
\end{figure}
Figure~\ref{gull} shows a photograp
```



Figure 1: Close-up of a gull

Figure 1 shows a photograph of a gull.

Figure 74

When a label is declared within a float environment, the `\ref{...}` will return the respective fig/table number, but it must occur **after** the caption. When declared outside, it will give the section number. To be completely safe, the label for any picture or table can go within the `\caption{}` command, as in

```
\caption{Close-up of a gull\label{gull}}
```

See the **Floats, Figures and Captions**³ section for more about the `figure` and related environments.

Fixing wrong labels

The command `\label` must appear after (or inside) `\caption`. Otherwise, it will pick up the current section or list number instead of what you intended.

```
\begin{figure}  
  \centering  
  \includegraphics[width=0.5\textwidth]{gull}  
  \caption{Close-up of a gull} \label{fig:gull}  
\end{figure}
```

Issues with links to tables and figures handled by hyperref

In case you use the package `hyperref` to create a PDF, the links to tables or figures will point to the caption of the table or figure, which is always below the table or figure itself⁴. Therefore the table or figure will not be visible, if it is above the pointer and one has to scroll up in order to see it. If you want the link point to the top of the image you can give the option `hypcap` to the `caption` package:

```
\usepackage[hypcap]{caption}
```

19.2.3. Formulas

Here is an example showing how to reference formulas:

⁴ <http://www.ctan.org/tex-archive/macros/latex/contrib/hyperref/README>

$$x^2 - 5x + 6 = 0 \tag{1}$$

```
\begin{equation} \label{eq:solve}
x^2 - 5x + 6 = 0
\end{equation}
```

$$x_1 = \frac{5 + \sqrt{25 - 4 \times 6}}{2} = 3 \tag{2}$$

$$x_2 = \frac{5 - \sqrt{25 - 4 \times 6}}{2} = 2 \tag{3}$$

```
\begin{equation}
x_1 = \frac{5 + \sqrt{25 - 4 \times 6}}{2} = 3
\end{equation}
```

Figure 75

```
\begin{equation}
x_2 = \frac{5 - \sqrt{25 - 4 \times 6}}{2} = 2
\end{equation}
```

and so we have solved `\ref{eq:solve}`

As you can see, the label is placed soon after the beginning of the math mode. In order to reference a formula, you have to use an environment that adds numbers. Most of the times you will be using the `equation` environment;

that is the best choice for one-line formulas, whether you are using `amsmath` or not. Note also the *eq:* prefix in the label.

eqref

The `amsmath` package adds a new command for referencing formulas; it is `\eqref{}`. It works exactly like `\ref{}`, but it adds brackets so that, instead of printing a plain number as 5, it will print (5). This can be useful to help the reader distinguish between formulas and other things, without the need to repeat the word "formula" before any reference. Its output can be changed as desired; for more information see the `amsmath` documentation.

tag

The `\tag{label}` command allows replacing the *labelnumber* by the arbitrary text string *label*.

numberwithin

The `amsmath` package adds the `\numberwithin{countera}{counterb}` command which replaces the simple `countera` by a more sophisticated `counterb.countera`. For example `\numberwithin{equation}{section}` in the preamble will prepend the section number to all equation numbers.

cases

The `cases` package adds the `\numcases` and the `\subnumcases` commands, which produce multi-case equations with a separate equation number and a separate equation number plus a letter, respectively, for each case.

19.3. The `varioref` package

The `varioref` package introduces a new command called `\vref{}`. This command is used exactly like the basic `\ref`, but it has a different output according to the context. If the object to be referenced is in the same page, it works just like `\ref`; if the object is far away it will print something like "5 on page 25", i.e. it adds the page number automatically. If the object is close, it can use more refined sentences like "on the next page" or "on the facing page" automatically, according to the context and the document class.

This command has to be used very carefully. It outputs more than one word, so it may happen its output falls on two different pages. In this case, the algorithm can get confused and cause a loop. Let's make an example. You label an object on page 23 and the `\vref` output happens to stay between page 23 and 24. If it were on page 23, it would print like the basic `\ref`, if it were on page 24, it would print "on the previous page", but it is on both, and this may cause some strange errors at compiling time that are very hard to be fixed. You could think that this happens very rarely; unfortunately, if you write a long document it

is not uncommon to have hundreds of references, so situations like these are likely to happen. One way to avoid problems during development is to use the standard `ref` all the time, and convert it to `vref` when the document is close to its final version, and then making adjustments to fix possible problems.

19.4. The `hyperref` package and `\autoref` { }

The `hyperref`⁵ package introduces another useful command; `\autoref`{ }. This command creates a reference with additional text corresponding to the target's type, all of which will be a hyperlink. For example, the command `\autoref{sec:intro}` would create a hyperlink to the `\label{sec:intro}` command, wherever it is. Assuming that this label is pointing to a section, the hyperlink would contain the text "section 3.4", or similar (the full list of default names can be found here⁶). Note that, while there's an

5 <http://en.wikibooks.org/wiki/LaTeX%2FPackages%2FHyperref>

6 <http://www.tug.org/applications/hyperref/manual.html#>

`\autoref*`

command that produces an unlinked prefix (useful if the label is on the same page as the reference), no alternative

`\Autoref`

command is defined to produce capitalized versions (useful, for instance, when starting sentences); but since the capitalization or autoref names was chosen by the package author, you can customize the prefixed text by redefining `\typeautorefname` to the prefix you want, as in:

```
\def\sectionautorefname{Section}
```

This renaming trick can, of course, be used for other purposes as well.

- If you would like a hyperlink reference, but do not want the predefined text that `\autoref{}` provides, you can do this with a command such as `\hyperref[sec:intro]{Appendix~\ref*{sec:intro}}`. Note that you can disable the creation of hyperlinks in `hyperref`, and just use these commands for automatic text.

- Keep in mind that the `\label` **must** be placed inside an environment with a counter, such as a table or a figure. Otherwise, not only the number will refer to the current section, as mentioned above⁷, but the name will refer to the previous environment with a counter. For example, if you put a label after closing a figure, the label will still say "figure n", on which n is the current section number.

19.5. The `hyperref` package and `\nameref` { }

The `hyperref` package also automatically includes the `nameref` package, and a similarly named command. It is similar to `\autoref`{ }, but inserts text corresponding to the section name, for example.

Input:

```
\section{MyFirstSection} \label{sec:marker}  
\section{MySecondSection}  
In section~\nameref{sec:marker} we defined...
```

Output:

In section MyFirstSection we defined...

19.6. The hyperref package and `\phantomsection`

When you define a `\label` outside a figure, a table, or other floating objects, the label points to the current section. In some case, this behavior is not what you'd like and you'd prefer the generated link to point to the line where the `\label` is defined. This can be achieved with the command `\phantomsection` as in this example:

%The link location will be placed on the line below.

```
\phantomsection  
\label{the_label}
```

19.7. References

20. Theorems

With "theorem¹" we can mean any kind of labelled enunciation that we want to look separated from the rest of the text and with sequential numbers next to it. This approach is commonly used for theorems in mathematics, but can be used for anything. LaTeX provides a command that will let you easily define any theorem-like enunciation.

¹ <http://en.wikipedia.org/wiki/Theorem>

20.1. Basic theorems

First of all, make sure you have the `amsthm` package enabled:

```
\usepackage{amsthm}
```

The easiest is the following:

```
\newtheorem{name}{Printed output}
```

put it in the preamble. The first argument is the name you will use to reference it, the second argument is the output LaTeX will print whenever you use it.

For example:

```
\newtheorem{mydef}{Definition}
```

will define the `mydef` environment; if you use it like this:

```
\begin{mydef}
Here is a new definition
\end{mydef}
```

It will look like this:

Definition 3 *Here is a new definition*

with line breaks separating it from the rest of the text.

20.2. Theorem counters

Often the counters are determined by section, for example "Theorem 2.3" refers to the 3rd theorem in the 2nd section of a document. In this case, specify the theorem as follows:

```
\newtheorem{name}{Printed output}[numberby]
```


where *numberby* is the name of the section level² (section/subsection/etc.) at which the numbering is to take place.

By default, each theorem uses its own counter. However it is common for similar types of theorems (e.g. Theorems, Lemmas and Corollaries) to share a counter. In this case, define subsequent theorems as:

```
\newtheorem{name}[counter]{Printed output}
```

where *counter* is the name of the counter to be used. Usually this will be the name of the master theorem.

You can also create a theorem environment that is not numbered by using the `newtheorem*` command³. For instance,

```
\newtheorem*{mydef}{Definition}
```

2 Chapter 3.1.4 on page 85

3 Requires the `amsthm` package

defines the `mydef` environment, which will generate definitions without numbering. This requires `amsthm` package.

20.3. Proofs

The `proof` environment⁴ can be used for adding the proof of a theorem. The basic usage is:

```
\begin{proof}
Here is my proof
\end{proof}
```

It just adds *Proof* in italics at the beginning of the text given as argument and a white square (Q.E.D⁵ symbol, also known as a tombstone⁶) at the end of

⁴ Requires the `amsthm` package

⁵ <http://en.wikipedia.org/wiki/Q.E.D>

⁶ <http://en.wikipedia.org/wiki/Tombstone%20%28typography%29>

it. If you are writing in another language than English, just use babel⁷ with the right argument and the word *Proof* printed in the output will be translated accordingly; anyway, in the source the name of the environment remains proof.

If you would like to manually name the proof, include the name in square brackets:

```
\begin{proof}[Proof of important theorem]
Here is my important proof
\end{proof}
```

If the last line of the proof is displayed math then the Q.E.D. symbol will appear on a subsequent empty line. To put the Q.E.D. symbol at the end of the last line, use the `\qedhere` command:

```
\begin{proof}
Here is my proof:
```

```
\[
a^2 + b^2 = c^2 \qedhere
\]
\end{proof}
```

The method above does not work with the deprecated environment `eqnarray*`. Use `align*` instead.

To use a custom Q.E.D. symbol, redefine the `\qedsymbol` command. To hide the Q.E.D. symbol altogether, redefine it to be blank:

```
\renewcommand{\qedsymbol}{} }
```

20.4. Theorem styles

It adds the possibility to change the output of the environments defined by `\newtheorem` using the `\theoremstyle` command⁸ in the header:

```
\theoremstyle{stylename}
```

the argument is the style you want to use. All subsequently defined theorems will use this style. Here is a list of the possible pre-defined styles:

stylename	Description
<code>plain</code>	Used for theorems, lemmas, propositions, etc. (default)
<code>definition</code>	Used for definitions and examples
<code>remark</code>	Used for remarks and notes

8 Requires the `amsthm` package

20.4.1. Custom styles

To define your own style, use the `\newtheoremstyle` command⁹:

```
\newtheoremstyle{stylename}% name of the style to be used
  {spaceabove}% measure of space to leave above the theorem. E.g.:
  3pt
  {spacebelow}% measure of space to leave below the theorem. E.g.:
  3pt
  {bodyfont}% name of font to use in the body of the theorem
  {indent}% measure of space to indent
  {headfont}% name of head font
  {headpunctuation}% punctuation between head and body
  {headspace}% space after theorem head; " " = normal interword space
  {headspec}% Manually specify head
```

(Any arguments that are left blank will assume their default value). Here is an example *headspec*:

```
\thmname{#1}\thmnumber{ #2}:\thmnote{ #3}
```

⁹ Requires the `amsthm` package

which would look something like:

Definition 2: Topology

for the following:

```
\begin{definition}[Topology]...
```

(The note argument, which in this case is Topology, is always optional, but will not appear by default unless you specify it as above in the head spec).

20.5. Conflicts

The theorem environment conflicts with other environments, for example *wrappfigure*. A work around is to redefine theorem, for example the following way:

```
% Fix latex
\def\smallskip{\vskip\smallskipamount}
\def\medskip{\vskip\medskipamount}
\def\bigskip{\vskip\bigskipamount}

% Hand made theorem
\newcounter{thm}[section]
\renewcommand{\thethm}{\thesection.\arabic{thm}}
\def\claim#1{\par\medskip\noindent\refstepcounter{thm}\hbox{\bf
\arabic{chapter}.\arabic{section}.\arabic{thm}. #1.}
\it\ %ignorespaces
}
\def\endclaim{
\par\medskip}
\newenvironment{thm}{\claim}{\endclaim}
```


In this case theorem looks like:

```
\begin{thm}{Claim}\label{lyt-prob}
Let it be.
Then you know.
\end{thm}
```

20.6. Notes

20.7. External links

- [<ftp://ftp.ams.org/pub/tex/doc/amscs/amsthdoc.pdf>
documentation]

amsthm

21. Algorithms and Pseudocode

LaTeX has a variety of packages that can help to format algorithms, code, and "pseudocode"¹. These packages provide stylistic enhancements over a uniform style (i.e., typewriter fonts) so that constructs such as loops or conditionals are visually separated from other text.

¹ <http://en.wikipedia.org/wiki/pseudocode>

22. Typesetting Algorithms

22.1. Typesetting using the `algorithmic` package

The `algorithmic` package uses a different set of commands than the `algorithmicx` package. Basic commands are:

```
\State <text>
\If{<condition>} <text> \EndIf
\For{<condition>} <text> \EndFor
\ForAll{<condition>} <text> \EndFor
```

```
\While{<condition>} <text> \EndWhile  
  
\Repeat <text> \Until{<condition>}  
  
\Require <text>  
  
\Return <text>  
  
\Comment{<text>}
```

Complete documentation is listed at <http://www.tug.org/texlive/Contents/live/text/dist/doc/latex/algorithms/algorithms.pdf>¹. Most commands are similar to the `algorithmicx` equivalents, but with different capitalization.

¹ <http://www.tug.org/texlive/Contents/live/texmf-dist/doc/latex/algorithms/algorithms.pdf>

22.2. Typesetting using the `algorithmicx` package

The `algorithmicx` package provides a number of popular constructs for algorithm designs. Put `\usepackage{algpseudocode}` in the preamble to use the algorithmic environment to write algorithm pseudocode (`\begin{algorithmic}... \end{algorithmic}`). You might want to use the algorithm environment (`\usepackage{algorithm}`) to wrap your algorithmic code in an algorithm environment (`\begin{algorithm}... \end{algorithm}`) to produce a floating environment with numbered algorithms.

The command `\begin{algorithmic}` can be given the optional argument of a positive integer, which if given will cause line numbering to occur at multiples of that integer. E.g. `\begin{algorithmic}[5]` will enter the algorithmic environment and number every fifth line.

Below is an example of typesetting a basic algorithm using the `algorithmicx` package (remember to add the

`\usepackage{algpseudocode}` statement to your document preamble):

```
\begin{algorithmic}
\If {$i \geq \maxval$}
  \State  $i$  gets 0
\Else
  \If {$i+k \leq \maxval$}
    \State  $i$  gets  $i+k$ 
  \EndIf
\EndIf
\end{algorithmic}
```

The LaTeX source can be written to a format familiar to programmers so that it is easy to read. This will not, however, affect the final layout in the document.

```
if  $i \geq \mathit{maxval}$  then  
     $i \leftarrow 0$   
else  
    if  $i + k \leq \mathit{maxval}$  then  
         $i \leftarrow i + k$   
    end if  
end if
```

Figure 76

There are several constructs provided by `algorithmix` detailed below.

22.2.1. If-statements

There are three forms of this construct


```
\If{<condition>} <text> \EndIf
\nIf{<condition>} <text> \Else <text> \EndIf
\nIf{<condition>} <text> \ElseIf{<condition>} <text> \Else <text>
\nEndIf
```

The third form accepts as many `\ElseIf{ }` clauses as required. Note that it is `\ElsIf` and not `\ElseIf`.

22.2.2. For-loops

There are two forms

```
\For{<condition>} <text> \EndFor
\nForAll{<condition>} <text> \EndFor
```

A traditional "for" loop. The method of iteration is usually described in the first argument,

e.g.

```
\For{$i = 1 \to 10$}  
  \State $i \gets i + 1$  
\EndFor
```

22.2.3. While-loops

```
\While{<condition>} <text> \EndWhile
```

```
\While{$i \leq 10$}  
  \State i=i+1;  
\EndWhile
```

22.2.4. Repeat until condition

```
\Repeat <text> \Until{<condition>}
```

22.2.5. Infinite loops

```
\Loop <text> \EndLoop
```

22.2.6. Precondition

```
\Require <text>
```

22.2.7. Postcondition

```
\Ensure <text>
```

22.2.8. Functions

```
\Function{<name>}{<params>}{<body> \EndFunction
```

22.2.9. Returning variables

```
\Return <text>
```

This command will usually be used in conjunction with a `\State` command as follows:

```
\Function{Increment}{${a}$}  
  \State $a \gets a+1$  
  \State \Return $a$  
\EndFunction
```

22.2.10. Comments

```
\Comment{<text>}
```

Note to users who switched from the old `algorithmic` package: comments may be placed everywhere in the source; there are no limitations as in the old `algorithmic` package.

22.2.11. Renaming things: algorithm to procedure, require/ensure to input/output

```
\floatname{algorithm}{Procedure}
\newcommand{\algorithmicrequire}{\textbf{Input:}}
\newcommand{\algorithmicensure}{\textbf{Output:}}
```

22.2.12. Custom algorithmic blocks

The `algorithmicx` package allows you to define your own environments. To define blocks beginning with a starting command and ending with an ending command, use

```
\algblock [<block>] {<start>} {<end>}
```

This defines two commands `\<start>` and `\<end>` which have no parameters. The text displayed by them is `\textbf{<start>}` and `\textbf{<end>}`.

With `\algblockdefx` you can give the text to be output by the starting and ending command and the number of parameters for these commands. In the text the n -th parameter is referenced by `#n`.

```
\algblockdefx[<block>]{<start>}{<end>}
[<startparamcount>][<default value>]{<start text>}
[<endparamcount>][<default value>]{<end text>}
```

Example:

```
\algblock[Name]{Start}{End}
\algblockdefx[NAME]{START}{END}%
[2][Unknown]{Start #1(#2)}%
{Ending}
\algblockdefx[NAME]{}{OTHEREND}%
[1]{Until (#1)}
\begin{algorithmic}
\Start
\Start
\START[One]{x}
\END
\START{0}
```

```
\OTHEREND{\texttt{True}}  
  
\End  
\Start  
\End  
  
\End  
\end{algorithmic}
```

More advanced customization and other constructions are described in the `algorithmicx` manual: <http://mirror.ctan.org/macros/latex/contrib/algorithmicx/algorithmicx.pdf>

22.3. The `algorithm` environment

It is often useful for the algorithm produced by `algorithmic` to be "floated" to the optimal point in the document to avoid it being split across pages. The `algorithm` environment provides this and a few other useful features. Include it by adding the

`\usepackage{algorithm}` to your document's preamble. It is entered into by

```
\begin{algorithm}
\caption{<your caption for this algorithm>}
\label{<your label for references later in your document>}
\begin{algorithmic}
<algorithmic environment>
\end{algorithmic}
\end{algorithm}
```

22.3.1. Algorithm numbering

The default numbering system for the `algorithm` package is to number algorithms sequentially. This is often not desirable, particularly in large documents where numbering according to chapter is more appropriate. The numbering of algorithms can be influenced by providing the name of the document component within which numbering should be recommenced. The

legal values for this option are: part, chapter, section, subsection, subsection or nothing (default). For example:

```
\usepackage[chapter]{algorithm}
```

22.3.2. List of algorithms

When you use figures or tables, you can add a list of them close to the table of contents; the `algorithm` package provides a similar command. Just put

```
\listofalgorithms
```

anywhere in the document, and LaTeX will print a list of the "algorithm" environments in the document with the corresponding page and the caption.

22.3.3. An example from the manual

This is an example taken from the manual (official manual, p.14²)

```
\begin{algorithm}
environment
\caption{Calculate  $y = x^n$ }
\label{alg1}
commands later in the document
\begin{algorithmic}
environment
\Require  $n \geq 0 \vee x \neq 0$ 
\Ensure  $y = x^n$ 
\State  $y \leftarrow 1$ 
\If{ $n < 0$ }
\State  $x \leftarrow 1 / x$ 
\State  $n \leftarrow -n$ 
\Else
\State  $x \leftarrow x$ 
\State  $n \leftarrow n$ 
\EndIf
\end{algorithmic}
% enter the algorithm
% give the algorithm a caption
% and a label for \ref{}
% enter the algorithmic
```

```

\While{$N \neq 0$}
  \If{$N$ is even}
    \State $X \Leftarrow X \times X$
    \State $N \Leftarrow N / 2$
  \Else[$N$ is odd]
    \State $Y \Leftarrow Y \times X$
    \State $N \Leftarrow N - 1$
  \EndIf
\EndWhile
\end{algorithmic}
\end{algorithm}

```

More information about all possible commands available at the project page

http://developer.berlios.de/docman/?group_id=3442

The official manual is located at

<http://mirrors.ctan.org/macros/latex/contrib/algorithm/algorithmms.pdf>

22.4. Typesetting using the program package

The `program` package provides macros for typesetting algorithms. Each line is set in math mode, so all the indentation and spacing is done automatically. The notation `|variable_name|` can be used within normal text, maths expressions or programs to indicate a variable name. Use `\origbar` to get a normal | symbol in a program. The commands `\A`, `\B`, `\P`, `\Q`, `\R`, `\S`, `\T` and `\Z` typeset the corresponding bold letter with the next object as a subscript (eg `\S1` typesets $\{\mathbf{S}_1\}$ etc). Primes work normally, eg `\S''`.

Below is an example of typesetting a basic algorithm using the `program` package (remember to add the `\usepackage{program}` statement to your document preamble):

```
\begin{program}
\mbox{A fast exponentiation procedure:}
\BEGIN %
\FOR i:=1 \TO 10 \STEP 1 \DO
```

```

lexpt(2,i); \\ |newline() \OD %
\rcoment{This text will be set flush to the right margin}
\WHERE
\PROC lexpt(x,n) \BODY
  z:=1;
  \DO \IF n=0 \THEN \EXIT \FI;
  \DO \IF |odd|(n) \THEN \EXIT \FI;
\COMMENT{This is a comment statement};
  n:=n/2; x:=x*x \OD;
  \{ n>0 \};
  n:=n-1; z:=z*x \OD;
  |print|(z) \ENDPROC
\END
\end{program}

```

A fast exponentiation procedure:

```
begin
  for i := 1 to 10 step 1 do
    expt(2, i);
    newline() od
  where
    proc expt(x, n) ≡
      z := 1;
      do if n = 0 then exit fi;
          do if odd(n) then exit fi;
              comment: This is a comment statement;
                  n := n/2; x := x * x od;
              {n > 0};
                  n := n - 1; z := z * x od;
          print(z).
      end
```

This text will be set flush to the right margin

Figure 77

The commands `\ (` and `\)` are redefined to typeset an algorithm in a minipage, so an algorithm can appear as a single box in a formula. For example, to state that a particular action system is equivalent to a WHILE loop you can write:

```

\[
\< \ACTIONS A:
    A \EQ \IF \B{} \THEN \S{}; \CALL A
    \ELSE \CALL Z \FI \QE
\ENDACTIONS \)
\EQT
\< \WHILE \B{} \DO \S{} \OD \)
\]

\begin{program}
\IF x = 1 \AR y:=y+1
\BAR x = 2 \AR y:=y^2
\utdots
\BAR x = n \AR y:=\displaystyle\sum_{i=1}^n y_i \FI

\DO 2 \origar x \AND x>0 \AR x:= x/2
\BAR \NOT 2 \origar x \AR x:= \modbar{x+3} \OD
\end{program}

```

Dijkstra conditionals and loops:

Loops with multiple exits:

```

\begin{program}
\DO \IF \B1 \THEN \EXIT \FI;
  \S1;
  \IF \B2 \THEN \EXIT(2) \FI \OD;
  \IF \B1 \THEN \EXIT \FI \OD
\end{program}

```

A Reverse Engineering Example.

Here's the original program:

```

\begin{program}
\VAR \seq{m := 0, p := 0, |last| := `` ''};
\ACTIONS |prog|:
|prog| \ACTIONEQ %
  \seq{|line| := `` '', m := 0, i := 1};
  \CALL |inhere| \ENDACTION
1 \ACTIONEQ %
  i := i+1;
  \IF (i=(n+1)) \THEN \CALL |alldone| \FI ;
  m := 1;
  \IF |item|[i] \neg |last|
    \THEN |write|(|line|); |line| := `` ''; m := 0;

```



```

\CALL |inhere| \FI ;

\CALL |more| \ENDACTION
|inhere| \ACTIONEQ %
  p := |number|[i]; |line| := |item|[i];
  |line| := |line| \concat ' ' \concat p;
\CALL |more| \ENDACTION

|more| \ACTIONEQ %
\IF (m=1) \THEN p := |number|[i];
  |line| := |line| \concat ' ' \concat p \FI ;
  |last| := |item|[i];
\CALL 1 \ENDACTION

|alldone| \ACTIONEQ |write|(|line|); \CALL z \ENDACTION \ENDACTIONS
\END
\end{program}

```

And here's the transformed and corrected version:

```

\begin{program}
\seq{|line| := ' ', i := 1};
\WHILE i \neg n+1 \DO
  |line| := |item|[i] \concat ' ' \concat |number|[i];
  i := i+1;
\WHILE i \neg n+1 \AND |item|[i] = |item|[i-1] \DO
  |line| := |line| \concat ' ' \concat |number|[i];

```

```
i := i+1 \OD ;  
  |write(|line|) \OD  
\end{program}
```

The package also provides a macro for typesetting a set like this: `\set{x
\in N | x > 0}`.

Lines can be numbered by setting `\NumberProgramstrue` and numbering turned off with `\NumberProgramsfalse`

Package page³

Package documentation⁴

3 <http://www.ctan.org/pkg/program>

4 [http://mirror.ctan.org/macros/latex/contrib/program/
program-doc.pdf](http://mirror.ctan.org/macros/latex/contrib/program/program-doc.pdf)

22.5. Source Code Formatting using the Listings package

(See the Listings package reference page⁵ for more information.)

A complete reference manual can be found at <http://tug.ctan.org/tex-archive/macros/latex/contrib/listings/listings.pdf>

This is a basic example for some Pascal code:

```
\documentclass{article}
\usepackage{listings}
\begin{document}
\lstset{language=Pascal} % Set your language (you can change
the language for each code-block optionally)

\begin{lstlisting}[frame=single] % Start your
code-block
```

⁵ [http://en.wikibooks.org/wiki/LaTeX%2FPackages%2FListings%](http://en.wikibooks.org/wiki/LaTeX%2FPackages%2FListings%2)

```
for i:=maxint to 0 do
begin
  { do nothing }
end;
Write('Case insensitive ');
Write('Pascal keywords. ');
\end{lstlisting}
\end{document}
```

```
begin
{ do nothing }
end;
Write(Case insensitive );
Write(Pascal keywords.);
```

Figure 78

23. References

- THE OFFICIAL MANUAL FOR THE ALGORITHMS PACKAGE, ROGÉRIO BRITO (2009), [HTTP://MIRRORS.CTAN.ORG/MACROS/LATEX/CONTRIB/ALGORITHMS/ALG](http://mirrors.ctan.org/macros/latex/contrib/algorithms/ALGORITHMS/ALG)

1 <http://mirrors.ctan.org/macros/latex/contrib/algorithms/algorithms.pdf>

24. Linguistics

There are a number of LaTeX packages available for writing linguistics papers. Various packages have been created for enumerated examples, syntactic trees, OT tableaux, feature matrices, IPA fonts, and many other applications. Some packages such as the

`tipa`

package are effectively standard within the field, while others will vary by author preference.

Some recommended packages¹:

- Glosses: `gb4e`;
- IPA symbols: `tipa`;
- OT Tableaux: `OTtblx`;
- Syntactic trees: `qtree + tree-dvips` (for drawing arrows);
- Alternatively, `xyling` is very powerful but not as user friendly as `qtree`;
- The `xy2` package itself has a steep learning curve, but allows a lot of control; for simplest trees use the `xymatrix` feature and arrows;
- `tikz-qtree3` has the same syntax as `qtree`, but uses `PGF/TikZ`, which allows more options for drawing arrows, etc.
- Dependency trees and bubble parses:

¹ <http://jones.ling.indiana.edu/~mdickinson/08/latex/slides.pdf>
pdf LaTeX for Linguists presentation

² <http://ctan.org/tex-archive/macros/generic/diagrams/xy-pic/xy>

³ <http://ctan.org/pkg/tikz-qtree>

- The TikZ-dependency⁴ package provides a high-level, convenient interface to draw dependency graphs. It is based on PGF/TikZ but does not require prior knowledge of TikZ in order to be used productively.
- Attribute-Value Matrices (AVMs): `avm`⁵
- John Frampton's `expex`: `expex`⁶

24.1. Enumerated examples

There are several commonly used packages for creating the kinds of numbered examples that are used in linguistics publications.

4 <http://sourceforge.net/projects/tikz-dependency/>

5 <http://nlp.stanford.edu/~manning/tex/avm.sty>

6 <http://www.math.neu.edu/ling/tex/>

24.1.1.1. gb4e

The

gb4e

package⁷ is called with:

```
\usepackage{gb4e}
```

Examples for this package are placed within the

exe

environment, and each example is introduced with the `\ex` command.

```
\begin{exe}  
  \ex This is an example.  
\end{exe}
```

⁷ <http://ctan.mines-albi.fr/help/Catalogue/entries/gb4e.html> / The gb4e package on CTAN

produces:

- (1) This is an example.

Figure 79

Multiple examples can be included within the environment, and each will have its own number.

```
\begin{exe}  
  \ex This is the first example.  
  \ex This is the second example.  
  \ex This is the third.  
\end{exe}
```

produces:

- (1) This is the first example.
- (2) This is the second example.
- (3) This is the third.

Figure 80

To create nested lists of examples, the

`xlist`

environment is used.

```
\begin{exe}
  \ex \begin{xlist}
    \ex This is a sub-example.
  \ex This is a second sub-example.
  \ex \begin{xlist}
    \ex This is a sub-sub-example.
    \ex This is a second sub-sub-example.
  \end{xlist}
\end{exe}
```

```
\end{xlist}  
\end{exe}
```

produces:

- (1) a. This is a sub-example.
- b. This is a second sub-example.
- c. i. This is a sub-sub-example.
- ii. This is a second sub-sub-example.

Figure 81

For notating acceptability judgments, the `\ex` command can take an optional argument. When including a judgment marker, the corresponding sentence must be surrounded by braces.

```
\begin{exe}  
  \ex This sentence is grammatical English.
```

```
\ex[*] {This sentence English in ungrammatical is.}
\end{exe}
```

produces:

(1) This sentence is grammatical English.

(2) * This sentence English in ungrammatical is.

Figure 82

Referencing examples in text works as it does in normal LaTeX documents. See the labeling and cross-referencing⁸ section for more details.

```
\begin{exe}
  \ex\label{ex1} Gozilla destroyed the city.
  \ex\label{ex2} Godzilla roared.
\end{exe}
```

Sentence `(\ref{ex1})` contains two arguments, but `(\ref{ex2})` contains only one.

Further details can be found in the full documentation available [here](#)⁹.

24.1.2. **lingmacros**

The

`lingmacros`

package¹⁰ created by Emma Pease is an alternate method for example numbering. This package uses two main commands `\enumsentence` and `\eenumsentence`. The former is used for singleton examples,

⁹ <http://ctan.mackichan.com/macros/latex/contrib/gb4e/gb4e-doc.pdf>

¹⁰ <http://ctan.org/tex-archive/macros/latex209/contrib/trees/tree-dvi.ps> The `lingmacros` package on CTAN


```
\enumsentence{This is an example.}
```

- (1) This is an example.

Figure 83

while the latter command is used for nested examples.

```
\enumsentence{\item This is the first example.  
              \item This is the second example.  
              \item This is the third.  
              }
```

- (1) This is the first example.
- (2) This is the second example.
- (3) This is the third.

Figure 84

Multiply nested examples make use of the normal LaTeX list environments¹¹.

```
\enumsentence{\item This is a sub-example.  
  \item This is a second sub-example.  
  \item \begin{enumerate}  
    \item This is sub-sub-example.  
    \item This is a second sub-sub-example.  
  \end{enumerate}  
}
```

produces:

- (1) a. This is a sub-example.
- b. This is a second sub-example.
- c. i. This is a sub-sub-example.
- ii. This is a second sub-sub-example.

Figure 85

Full documentation can be found [here](#)¹².

24.2. Syntactic trees

Often, linguists will have to illustrate the syntactic structure of a sentence. One device for doing this are syntactic trees. Unfortunately, trees look very different in different grammar formalisms, and different LaTeX packages are suited for different formalisms.

24.2.1. Constituent trees

While there are several packages for drawing syntactic trees available for LaTeX, this article focuses on the `qtree` and `xyling` packages.

12 <http://mirrors.ibiblio.org/pub/mirrors/CTAN/macros/latex209/contrib/trees/tree-dvips/lingmacros-manual.pdf>

qtrees

Drawing trees with `qtrees` is relatively straightforward. First, the `qtrees` package has to be included in the document's preamble:

```
\usepackage{qtrees}
```

A new tree is started using the `\Tree` command, each (sub-)tree is indicated by brackets `[]`. The root of a (sub-)tree is always preceded by a `.`, leaf nodes are simply expressed by their labels.

For example, the following code

```
\Tree [ .S [ .NP LaTeX ] [ .VP [ .V is ] [ .NP fun ] ] ]
```

produces this syntactic tree as output:

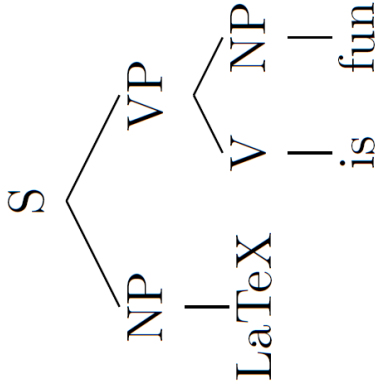


Figure 86

Note that the spaces before the closing brackets, as well as the space between `\Tree` and the first opening bracket are **mandatory**.

By default, `qtree` centers syntactic trees on the page. This behaviour can be turned off by either specifying the behaviour when loading the package

```
\usepackage[nocenter]{qtree} % do not center trees
```

or via the command

```
\qtreetocenterfalse % do not center trees from here on
```

anywhere in the document. The effect of the latter can be undone by using the command

```
\qtreetocentertrue % center trees from here on
```

tikz-qtree

Using the same syntax as `qtree`, `tikz-qtree` is another easy-to-use alternative for drawing syntactic trees.

For simple trees, `tikz-qtrees` is completely interchangeable with `qtrees`. However, some of `qtrees`'s advanced features are implemented in a different way, or not at all. On the other hand, `tikz-qtrees` provides other features such as controlling the direction of the tree's growth (top to bottom, left to right etc.) or different styles for edges.

To use the `tikz-qtrees` package for drawing trees, put the following into the document's preamble:

```
\usepackage{tikz}
\usepackage{tikz-qtrees}
```

The syntax of `tikz-qtrees` and result when drawing a simple tree is the same as for `qtrees`.

```
\Tree [.S [.NP LaTeX] [.VP [.V is] [.NP fun]] ] ]
```

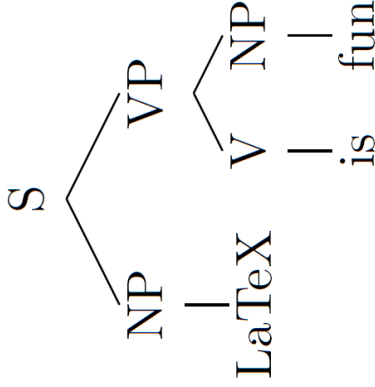


Figure 87

Note that, other than for `qtree`, trees are not centered by default. To center them, put them into a centered environment:

```
\begin{center}
```



```
\Tree [.S [.NP LaTeX ] [.VP [V is ] [.NP fun ] ] ] ]
\end{center}
```

For setting the style of trees, `tikz-qtree` provides the `\tikzset` command. For example, to make a tree grow from left to right instead of from top to bottom, use the following code:

```
\tikzset{grow'=right} % make trees grow from left to right
\tikzset{every tree node/.style={anchor=base west}} % align nodes of
the tree to the left (west)
\Tree [.S [.NP LaTeX ] [.VP [V is ] [.NP fun ] ] ] ]
```

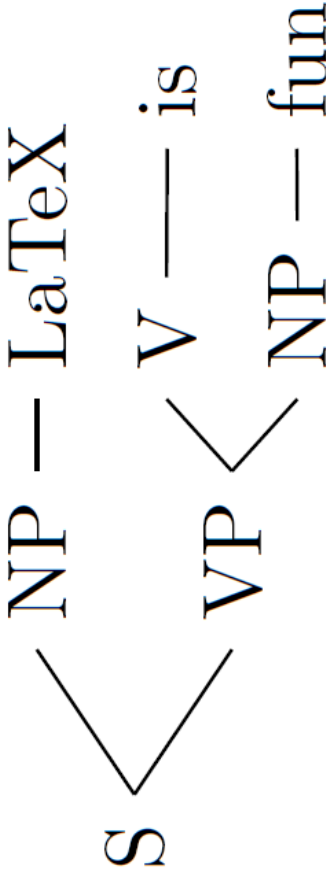


Figure 88

The above code changes the default orientation for **all** trees that are defined after `\tikzset` commands. To only change the direction of a single tree, it has to be put into a `\tikzpicture` environment:

```

\begin{tikzpicture} % all changes only affect trees within this
environment
\tikzset{grow'=right} % make trees grow from left to right
\tikzset{every tree node/.style={anchor=base west}} % align nodes of
the tree to the left (west)
\Tree [.S [.NP LaTeX ] [.VP [.V is ] [.NP fun ] ] ]
\end{tikzpicture}

```

24.2.2. Dependency Trees

Dependency trees can take multiple visual forms. Commonly, they quite resemble phrase structure trees. Alternatively, they can be captured by brackets drawn above running text.

Two-dimensional Dependency Trees

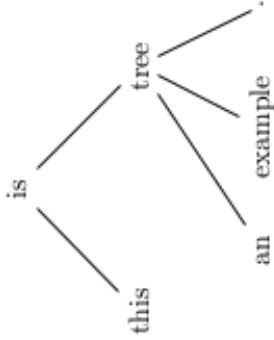
These can be either achieved using the fairly universal drawing package TikZ, like so:

```
% In the preamble:
\usepackage{tikz}

% In the document:
\begin{tikzpicture}
  \node (is-root) {is}
    [sibling distance=3cm]
    child { node {this} }
    child {
      node { tree }
      [sibling distance=1.5cm]
      child { node {an} }
      child { node {example} }
      child { node {.} }
      child[missing]
    }
  ;
\path (is-root) + (0,-2.5\tikzleveldistance)
```

```
node {\textit{This is an example tree.}};  
\end{tikzpicture}
```

which gives you the following drawing:



This is an example tree.

Figure 89 A dependency tree created using TikZ

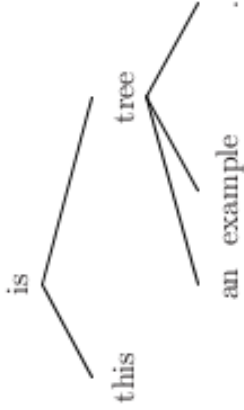
TikZ has the advantage that it allows for generating PDF directly from the LaTeX source, without need for any detour of compiling to DVI using `latex`, and then converting to PDF probably via PS using tools such as `dvi2ps` and `ps2pdf`. Latter is the case of another package based on the package **xy**, namely **xyling**.

The code for a similar tree using **xyling** might look like:

```
% In the preamble:
\usepackage{xytree}

% In the document:
\Tree{
  & \K{is}\B{dl}\B{drr} \& \& \K{this} \&\& \K{tree}\B{dll}\B{dl}\B{dr} \& \& \K{an} & \K{example} \&\& \K{.} }
\medskip
\textit{This is an example tree.}
```

which gives you a drawing like this:



This is an example tree.

Figure 90 A dependency tree created using `xyling`

Dependency Trees as Brackets above Text

One way to typeset dependency brackets above running text is using the package `xytree`. It gives you fairly good control of how the brackets are typeset but requires compiling the LaTeX code to DVI (and perhaps converting to PDF using the tools `dvips` and `ps2pdf` later).

An example code:

```
% In the preamble:
\usepackage{xytree}

% In the document:
\xytext{
  \xybarnode{Peter} &~~~&
  \xybarnode{and}
  \xybarconnect (UL,U){-2}"_{\small conj}"
  \xybarconnect (UR,U){2}"^{\small conj}"
  &~~~&
  \xybarnode{Mary} &~~~&
  \xybarnode{bought}
  \xybarconnect [8] (UL,U){-4}"_{\small subj}"
```



```

\xybarconnect[13]{6}^{\small punct} "
\xybarconnect[8](UR,U){4} "\small obj"
&~~~&
\xybarnode{a} &~~~&
\xybarnode{car}
\xybarconnect(UL,U){-2} "\small det"
&~~~&
\xybarnode{.}
}

```

results in:

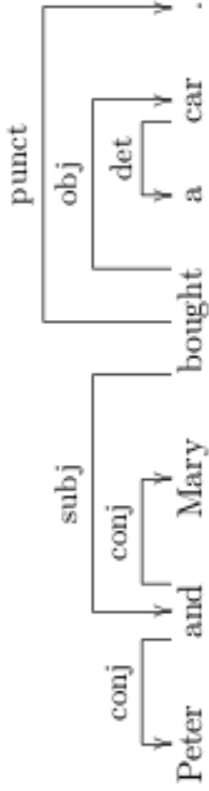


Figure 91 A dependency tree above running text created using `xytree`

Dependency Trees using TikZ-dependency

The package provides high level commands to design and style dependency graphs. To draw a graph, you only need to create a `depxtext` environment, write the text of the sentence within the `depxtext` environment and use `depedge` commands to draw the edges. Global and local optional parameters can be used to style and fine tune the looks of the graph, as shown in the following example:

```
% In the preamble:
\usepackage{tikz-dependency}

% In the document:
\begin{dependency}[theme = simple]
  \begin{depxtext}[column sep=1em]
    A \& hearing \& is \& scheduled \& on \& the \& issue \& today
  \end{depxtext}
  \deproot{3}{ROOT}
  \depedge{2}{1}{ATT}
  \depedge[edge start x offset=-6pt]{2}{5}{ATT}
\end{dependency}
```

```

\depedge{3}{2}{SBJ}
\depedge{3}{9}{PU}
\depedge{3}{4}{VC}
\depedge{4}{8}{TMP}
\depedge{5}{7}{PC}
\depedge[arc angle=50]{7}{6}{ATT}
\end{dependency}

```

This code snippet would produce the following result:

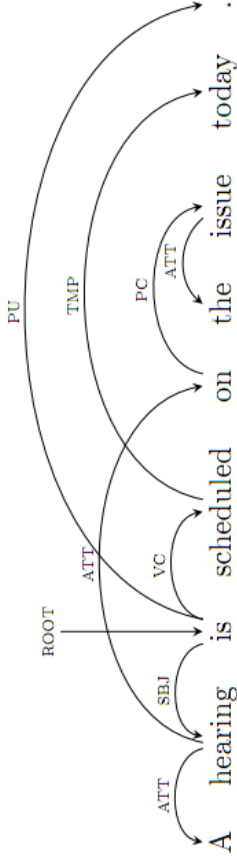


Figure 92 A dependency tree drawn with TikZ-dependency.

24.3. Glosses

24.3.1. With gb4e

Below, it is explained how to a glossed example in your document using package

```
gb4e
```

```
.
```

To create a glossed example, use the normal

```
exe
```

environment. But after the `\ex` tag, introduce the example and its gloss using `\gll` and the translation after it with `\trans` tag.

```
\begin{exe}  
\ex  
\gll Кор ест сметану\\  
cat.NOM eat.3.SG.PRS sour-cream.ACC\\
```

```
\trans 'The cat eats sour cream'  
\end{exe}
```

The code will produce the following output:

```
(1) KoT      ect      cMeTaHy  
    Cat.NOM eat.3.SG.PRS sour-cream.ACC  
'The cat eats sour cream'
```

Figure 93

Vertically aligned glosses are separated by spaces, so if it's necessary to include a space in part the gloss, simply enclose the connected parts inside braces.

```
\begin{exe}  
\ex
```

```
\gll Pekka pel\ "astyi karhusta.\\  
  Pekka {became afraid} bear.EIA\<\  
\trans `Pekka became afraid because of the/a bear.'  
\end{exe}
```

24.3.2. With **lingmacros**

The

`lingmacros`

package uses the `\shortex` command to introduce glossed examples inside the `\enumsentence` and `\enumsentence` commands. This command takes four arguments and builds off the normal tabular¹³ environment. Its first argument specifies the number of columns in the gloss. The second and third arguments give the text and its gloss respectively, and items within each column are divided by the usual `&` tabular separator. The fourth argument is the translation.

```
\enumsentence{\shortex{3}
  {Pekka & pel\"astyi & karhu-sta.}
  {Pekka & became afraid & bear.ELA}
  {'Pekka became afraid because of the/a bear.'}
}
```

24.4. IPA characters

The

`tipa`

package is the standard LaTeX package for International Phonetic Alphabet symbols.

```
\usepackage{tipa}
```

There are two methods for getting IPA symbols into a document. The first way is to use the

IPA

environment.

```
\begin{IPA}  
text in IPA format here  
\end{IPA}
```

This method is useful for long stretches of text that need to be in IPA. Alternatively, there is the `\textipa` command that will format the text in its argument into IPA. This command is similar to other font typesetting commands¹⁴.

```
\textipa{text in IPA format here}
```

14 <http://en.wikibooks.org/wiki/LaTeX%2FFormatting%23Fonts>

24.4.1. Basic symbols

The IPA format works by translating ASCII characters into corresponding IPA symbols. Lower case letters are rendered as usual,

```
\textipa{abcdefghijklmnopqrstuvwxyz}
```

abcdefghijklmnopqrstuvwxyz

Figure 94

however capital letters are rendered differently.

```
\textipa{ABCDEFGHIJKLMNPOQRSTUVWXYZ}
```

produces:

αβδεϕγ hijκλμ ηζ ς Γ ρ θ σ υ π ρ χ υ ζ

Figure 95

Punctuation marks that are normally used in LaTeX are also rendered faithfully in the

IPA

environment.

```
\textipa{! * + = ? . , / [ ] ( ) ` ' | || }
```

produces:

! * + = ? . , / [] () ` ' | ||

Figure 96

Numerals and @ also have variants in the

tipa

environment.

```
\textipa{1234567890 @}
```

produces:

ɪ ʌ ʒ ɥ ɐ ɒ ɔ ɛ ɞ ɟ

Figure 97

In addition, there are a number of special macros for representing symbols that don't have other associations, some of which are listed here. For a complete list see the official TIPA Manual¹⁵.

The `\;` macro preceding a capital letter produces a small caps version of the letter.

¹⁵ <http://uit.no/getfile.php?PageId=874&FileId=165> TIPA manual

```
\textipa{\;A \;B \;E \;G \;H \;I \;L \;R \;Y}
```

produces:

A B E G H I L R Y

Figure 98

The `\:` macro produces retroflex symbols.

```
\textipa{\:d \:l \:n \:r \:s \:t \:z}
```

gets you:

d l n r s t z,

Figure 99

The `\!` macro produces implosive symbols and the bilabial click.

```
\textipa{\!b \!d \!g \!j \!G \!o}
```

gets you:

b d g j G O

Figure 100

25. References

26. External links

- LaTeX for Linguists¹
- The qtree package for drawing syntactic trees.²
- The gb4e package page on CTAN.³

1 <http://www.essex.ac.uk/linguistics/external/clmt/latex4ling/>

2 <http://www.ling.upenn.edu/advice/latex/qtrees/>

3 <http://www.ctan.org/tex-archive/macros/latex/contrib/gb4e/>

Part IV.

Special Documents and Pages

27. Indexing

Especially useful in printed books, an index is an alphabetical list of words and expressions with the pages of the book upon which they are to be found. LaTeX supports the creation of indices with its package `makeidx`, and its support program `makeindex`, called on some systems `makeidx`.

27.1. Using `makeidx`

To enable the indexing feature of LaTeX, the `makeidx` package must be loaded in the preamble with:

```
\usepackage{makeidx}
```

and the special indexing commands must be enabled by putting the

```
\makeindex
```

command into the input file preamble. This should be done within the preamble, since it tells LaTeX to create the files needed for indexing. To tell LaTeX what to index, use

```
\index{key}
```

where *key* is the index entry and does not appear in the final layout. You enter the index commands at the points in the text that you want to be referenced in the index, likely near the reason for the *key*. For example, the text

To solve various problems in physics, it can be advantageous to express any arbitrary piecewise-smooth function as a Fourier Series composed of multiples of multiples of sine and cosine functions.

can be re-written as

To solve various problems in physics, it can be advantageous to express any arbitrary piecewise-smooth function as a Fourier Series

```
\index{Fourier Series}
```

composed of multiples of sine and cosine functions.

to create an entry called 'Fourier Series' with a reference to the target page. Multiple uses of *\index* with the same *key* on different pages will add those target pages to the same index entry.

To show the index within the document, merely use the command

```
\printindex
```

It is common to place it at the end of the document. The default index format is two columns.

The `showidx` package that comes with LaTeX prints out all index entries in the left margin of the text. This is quite useful for proofreading a document and verifying the index.

27.1.1.1. Compiling indices

When the input file is processed with LaTeX, each `\index` command writes an appropriate index entry, together with the current page number, to a special file. The file has the same name as the LaTeX input file, but a different extension (`.idx`). This `.idx` file can then be processed with the `makeindex` program. Type in the command line:

```
makeindex filename
```

Note that *filename* is without extension: the program will look for *filename.idx* and use that. You can optionally pass *filename.idx* directly to the program as an argument. The `makeindex` program generates a sorted index with the same base file name, but this time with the extension `.ind`. If now the LaTeX input file is processed again, this sorted index gets included into the document at the point where LaTeX finds `\printindex`.

The index created by latex with the default options may not look as nice or as suitable as you would like it. To improve the looks of the index `makeindex` comes with a set of style files, usually located somewhere in the `tex` directory structure, usually below the `makeindex` subdirectory. To tell `makeindex` to use a specific style file, run it with the command line option:

```
makeindex -s
```

Many technical documents use terms or acronyms unknown to the general population. It's common practice to add glossaries to make those works more understandable.

The `glossaries` package was created to assist users in creating glossaries. It supports multiple glossaries, acronyms and symbols.

It replaces the `glossary` package and can be used instead of the `nomencl` package.

27.2. Using glossaries

To enable the use of `glossaries` package, you have to load the package:

```
\usepackage{glossaries}
```

if you will be using `xindy` (highly recommended) rather than `makeindex` you need to specify `xindy` option:

```
\usepackage[xindy]{glossaries}
```

For the glossary to show up in Table of Contents you need to additionally add `toc` option:

```
\usepackage[toc]{glossaries}
```

See also `#Custom_TOC-Entry1` at the bottom of this page.

The glossary index won't be generated until you place the following command in document preamble:

¹ Chapter ?? on page ??

`\makeglossaries`

Note that the links in generated glossary won't be “clickable” unless you load this package *after* the `hyperref` package.

Windows users will need to install Perl for `makeglossaries` to work.

27.3. Defining glossary entries

To use an entry from glossary you first need to define it. There are few ways to define an entry depending on what you define and how it is going to be used.

Note that a defined entry *won't* be included in the printed glossary *unless* it is used in the document. This enables you to create a glossary of general terms and just `\include` it in all your documents.

27.4. Defining terms

To define a term in glossary you use `\newglossaryentry` macro:

```
\newglossaryentry{ is a unique label used to identify an entry in  
glossary, <settings> are comma separated pairs used to define an  
entry.
```

For example, to define a computer entry:

The above example defines an entry that has the same label and entry name.

This is not always the case as the next entry will show:

When you define terms, you need to remember that they will be sorted by or .

While is a bit more LaTeX aware, it does it by omitting latex macros () thus incorrectly sorting the above example as .

won't fare much better, because it doesn't understand TeX macros, it will interpret the word exactly as it was defined, putting it inside symbol class, before words beginning with .

Therefore it's needed to extend our example and specify how to sort

the word:

You can also specify plural forms, if they are not formed by adding "s" (we will learn how to use them in next section):

Defined entries can also be symbols:

You can also define both a name and a symbol:

Note that not all glossary styles show defined symbols.

To define a new acronym you use the macro:

Sometimes the mundane things are the most painful. However, it doesn't have to be that way because of evolved, user-friendly templates. Thankfully, LaTeX allows for very quick letter writing, with little hassle.

27.5. The letter class

To write letters use the standard document class *letter*.

You can write multiple letters in one LaTeX file - start each one with `\begin{letter}` and end with `\end{letter}`. You can leave *recipient* blank. Each letter consists of four parts:

1. opening (like `\opening{Dear Sir or Madam,}` or `\opening{Dear Kate,}`)
2. main body - written as usual in LaTeX
3. closing (like `\closing{Yours sincerely,}`)

LaTeX will leave some space after closing for your hand-written signature; then it will put your name and surname, if you have declared them.

4. additional elements: post scripta, carbon copy and list of enclosures

If you want your name, address and telephone number to appear in the letter, you have to declare them first signature, address and telephone.

The output letter will look like this:

11 White Street
New Canaan, CT 06840
Dorwick Old Home
April 24, 2007

Henry
100
32 Andrew Road
Newtown, CT 06470

Dear Sir or Madam:

I am writing to you on behalf of the Wikimedia project (<http://www.wikimedia.org/>), and to thank you for your contribution to the free encyclopedia, Wikipedia. We are pleased to have your name listed in our copyright notices.

...

That said, allow me to estimate that your material will be used to the noble end of providing a free volume of knowledge to the world in a manner that is not only free of charge, but also free of copyright. We will not be able to identify you in our standard form and post it back to you. We shall greatly appreciate it.

Thank you for your time and consideration.

I look forward to your reply.

Yours faithfully,

Jim Borgas

P.S. You can find the full text of GFDL license at <http://www.gnu.org/copyleft/ll.html>, and Copyright permission form.

Figure 101 A
sample letter.

Here is the example's code:

```
\documentclass{letter}
```

```
\usepackage{hyperref}
\signature{Joe Bloggs}
\address{21 Bridge Street \ Smallville \ Dunwich DU3 4WE}
\begin{document}

\begin{letter}{Director \ Doe \& Co \ 35 Anthony Road
\ Newport \ Ipswich IP3 5RT}
\opening{Dear Sir or Madam:}
```

I am writing to you on behalf of the Wikipedia project
(<http://www.wikipedia.org/>),
an endeavour to build a fully-fledged multilingual encyclopaedia in
an entirely
open manner, to ask for permission to use your copyrighted material.

```
% The \ldots command produces dots in a way that will not upset
% the typesetting of the document.
\ldots
```

That said, allow me to reiterate that your material will be used to
the noble end of
providing a free collection of knowledge for everyone; naturally
enough, only if you
agree. If that is the case, could you kindly fill in the attached
form and post it

back to me? We shall greatly appreciate it.
Thank you for your time and consideration.
I look forward to your reply.

```
\closing{Yours Faithfully,}

\ps{P.S. You can find the full text of GFDL license at
\url{http://www.gnu.org/copyleft/fdl.html}.}
\encl{Copyright permission form}

\end{letter}

\end{document}
```

To move the closing and signature parts to the left, insert the following before `\begin{document}`:

```
\longindentation=0pt
```

The amount of space to the left can be adjusted by increasing the `0pt`.

27.6. Envelopes

27.6.1. Using the `envlab` package

The `envlab` package provides customization to the `\makeLabels` command, allowing the user to print on any of an assortment of labels or envelope sizes. For example, beginning your LaTeX file the following way produces a document which includes the letter and a business-size (#10) envelope on the following page.

```
\documentclass{letter}
\usepackage[businessenvelope]{envlab}
\makeLabels
```

Refer to the `envlab` user guide² for more information about this capable package.

² <http://mirror.hmc.edu/ctan/macros/latex/contrib/envlab/elguide.pdf>

27.6.2. Using the `geometry` package

Here is a relatively simple envelope which uses the `geometry` package which is used because it vastly simplifies the task of rearranging things on the page (and the page itself).

```
% envelope.tex
\documentclass{letter}
\usepackage[left=1in,to
p=0.15in,papersize={4.125in,9.5in},landscape,twoside=false]{geometry}
\setlength\parskip{0pt}
\pagestyle{empty}

\begin{document}

FROM-NAME

FROM-STREET ADDRESS

FROM-CITY, STATE, \ ZIP

\vspace{1.0in}\large
```

```
\setlength\parindent{3.6in}

TO-NAME

TO-STREET ADDRESS

TO-CITY, STATE, \ ZIP

\end{document}
```

Figure 102

A sample envelope to be printed in landscape mode.

27.6.3. Printing

The above will certainly take care of the spacing but the actual printing is between you and your printer. One user reports that printing envelopes created with `enlab` is relatively painless. If you use the `geometry` package, you may find the following commands useful for printing the envelope.

```
$ pdflatex envelope.tex
$ pdf2ps envelope.pdf
$ lpr -o landscape envelope.ps
```

Alternatively, you can use the `latex dvi` output driver.

In the first line, `dvips` command converts the `.dvi` file produced by `latex` into a `.ps` (PostScript) file. In the second line, the PostScript file is sent to the printer.

```
$ latex envelope.tex && dvips -t unknown -T 9.5in,4.125in
envelope.dvi
$ lpr -o landscape envelope.ps
```

It is reported that `pdflatex` creates the right page size but not `dvips` despite what it says in the `geometry` manual. It will never work though unless your printer settings are adjusted to the correct page style. These settings depend on the printer filter you are using and in CUPS might be available on the `lpr` command line if you are masochistic.

27.7. Windowed envelopes

An alternative to separately printing addresses on envelopes is to use the letter class from the KOMA package. It supports additional features like folding marks and the correct address placement for windowed envelopes. Using the `scrltr2` document class from the KOMA package the example letter code is:

```
% koma_env.tex
\documentclass[a4paper]{scrltr2}
\usepackage{lmodern}
\usepackage[utf8]{inputenc}
```

```
\usepackage[T1]{fontenc}
\usepackage[english]{babel}
\usepackage{url}

\setkomavar{fromname}{Joe Bloggs}
\setkomavar{fromaddress}{21 Bridge Street \\\ Smallville \\\ Dunwich
DU3 4WE}
\setkomavar{fromphone}{0123 45679}

\begin{document}

\begin{letter}{Director \\\ Doe \& Co \\\ 35 Anthony Road
\\\ Newport \\\ Ipswich IP3 5RT}

\KOMAOptions{fromphone=true,fromfax=false}
\setkomavar{subject}{Wikipedia}
\setkomavar{customer}{2342}
\opening{Dear Sir or Madam,}
```

I am writing to you on behalf of the Wikipedia project (<http://www.wikipedia.org/>), an endeavour to build a fully-fledged multilingual encyclopaedia in an entirely open manner, to ask for permission to use your copyrighted material.

```
\ldots
```

That said, allow me to reiterate that your material will be used to the noble end of providing a free collection of knowledge for everyone; naturally enough, only if you agree. If that is the case, could you kindly fill in the attached form and post it back to me? We shall greatly appreciate it.

Thank you for your time and consideration.

I look forward to your reply.

```
\closing{Yours Faithfully,}  
\ps{P.S. You can find the full text of GFDL license at  
\url{http://www.gnu.org/copyleft/fdl.html}.}  
\encl{Copyright permission form}
```

```
\end{letter}
```

```
\end{document}
```

The output is generated via

```
$ pdflatex koma_env
```

```
Jan Hage
Koma-Envs
Schnitzler
November 13, 2012
Dresden, Germany

An Ruck! 11.11.12 (Tue, Nov 13, 2012) 20:05
EBC/IC
Dresden, Germany
Dresden, Germany
November 13, 2012

Content on
this
website

Dear Sir or Madam,

I am writing to you on behalf of the Wikimedia project (http://www.wikimedia.org/),
an endeavor to build a free digital encyclopedia, to an existing work
of yours, to ask for permission to use your copyrighted material.

We are currently working on a project to build a free encyclopedia
providing a free overview of knowledge for everyone, across all languages
and in all free encyclopædias. We call it the Wikimedia Free and open text book project.

Thank you for your time and consideration.

I am looking forward to your reply.

Yours faithfully,

Jan Hage

PS: You can find the full text of DMCA license at http://www.gnu.org/copyleft/
and Copyright permission form
```

Figure 103 A
sample letter with
folding marks ready
for standardized
windowed envelopes.

Folding the print of the resulting file `koma_env.pdf` according to the folding marks it can be placed into standardized windowed envelopes DIN C6/5, DL, C4, C5 or C6.

In addition to the default, the KOMA-package includes predefined format definitions for different standardized Swiss and Japanese letter formats.

27.8. Reference: `letter.cls` commands

command	description
<code>\name{}</code>	
<code>\signature{}</code>	
<code>\address{}</code>	
<code>\location{}</code>	
<code>\telephone{}</code>	
<code>\makelabels</code>	

command	description
<code>\stopbreaks</code>	
<code>\startbreaks</code>	
<code>\opening{}</code>	
<code>\closing{}</code>	
<code>\cc{}</code>	Start a parbox introduced with <code>\ccname:</code>
<code>\encl{}</code>	Start a parbox introduced with <code>\enclname:</code>
<code>\ps</code>	Begins a new paragraph, normally at the close of the letter
<code>\stopletter</code>	(empty)
<code>\returnaddress</code>	(empty)
<code>\startlabels</code>	
<code>\mlabel{ }{ }</code>	
<code>\descriptionlabel{ }</code>	
<code>\ccname</code>	"cc"

command	description
<code>\enclname</code>	"encl"
<code>\pagename</code>	"Page"
<code>\headtoname</code>	"To"
<code>\today</code>	Long form date

environment	Description
<code>letter{ }</code>	See main article
<code>description</code>	
<code>verse</code>	
<code>quotation</code>	
<code>quote</code>	

27.9. Sources

- KOMA-Script - The Guide³
- pl:LaTeX/Pisanie listów⁴ fr:LaTeX/Lettre⁵

3 <http://www.ctan.org/tex-archive/macros/latex/contrib/koma-script/scrguien.pdf>

4 <http://pl.wikibooks.org/wiki/LaTeX%2FPisanie%20list%F3w>

5 <http://fr.wikibooks.org/wiki/LaTeX%2FLettre>

28. Presentations

LaTeX can be used for creating presentations. There are several packages for the task, including the Beamer package.

28.1. The Beamer package

The beamer package is provided with most LaTeX distributions, but is also available from CTAN¹. If you use MikTeX, all you have to do is to include the beamer package and let LaTeX download all wanted packages automatically. The documentation² explains the features in great detail. You can also have a look at the PracTex article **Beamer by example**³

28.1.1. Introductory example

The beamer package is loaded by calling the `beamer` class:

-
- 1 `http://www.ctan.org/tex-archive/macros/latex/contrib/beamer/`
 - 2 `http://www.ctan.org/tex-archive/macros/latex/contrib/beamer/doc/beameruserguide.pdf`
 - 3 Andrew Mertz and William Slough *Beamer by Example*

```
\documentclass{beamer}
```

The usual header information may then be specified. Note that if you are compiling with XeTeX then you should use

```
\documentclass[xetex,mathserif,serif]{beamer}
```

Inside the usual document environment, multiple frame environments specify the content to be put on each slide. The `frametitle` command specifies the title for each slide (See image):

```
\begin{document}
\begin{frame}
\frametitle{This is the first slide}
%Content goes here
\end{frame}
\begin{frame}
\frametitle{This is the second slide}
\framesubtitle{A bit more information about this}
%More content goes here
```



```
\end{frame}  
% etc  
\end{document}
```

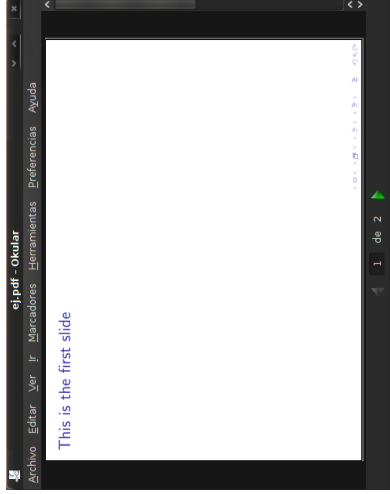


Figure 104

Usual environments (`itemize`, `enumerate`, `equation`, `etc.`) may be used as usual.

Inside of frames, you can use environments like `block`, `theorem`, `proof`, ... Also, `\maketitle` is possible to create the frontpage, if `title` and `author` is set.

Trick : Instead of using `\begin{frame}... \end{frame}`, you can also use `\frame{...}`.

For the actual talk, if you can compile it with pdfLaTeX then you could use Adobe Reader with its fullscreen mode. If you want to navigate in your presentation, you can use the almost invisible links in the bottom right corner without leaving the fullscreen mode.

28.1.2. Document Structure

Title page and information

You give information about authors, titles and dates in the preamble

```
\title[Crisis](optional, only for long titles)
{The Economics of Financial Crisis}
\subtitle[Evidence from India]
\author[Author, Anders] % (optional, for multiple authors)
{F.~Author\inst{1} \and S.~Anders\inst{2}}
\institute[Universitäten Hier und Dort] % (optional)
{
  \inst{1}%
  Institut für Informatik\\
  Universität Hier
  \and
  \inst{2}%
  Institut für theoretische Philosophie\\
  Universität Dort
}
\date[KPT 2004] % (optional)
```

```
{Konferenz über Präsentationstechniken, 2004}  
\subject{Informatik}
```

In the document, you add the title page :

```
\frame{\titlepage}
```

Table of Contents

You can print the table of contents and highlight the current section/subsection by typing :

```
\begin{frame}  
\frametitle{Table of Contents}  
\tableofcontents[currentsection]  
\end{frame}
```

You can automatically print the table of contents at the beginning of each section by adding in the preamble the following line.

```
\AtBeginSection []
{
  \begin{frame}
  \frametitle{Table of Contents}
  \tableofcontents[currentsection]
  \end{frame}
}
```

You can do the same for subsections :

```
\AtBeginSubsection []
{
  \begin{frame}
  \frametitle{Table of Contents}
  \tableofcontents[currentsection, currentsubsection]
  \end{frame}
}
```

References (Beamer)

```
\begin{frame}[allowframebreaks]
\frametitle{presentation}>Weiterf, hrende Literatur}
\begin{thebibliography}{10}
\beamertemplatebookbibitems
\bbitem{Autor1990}
A.~Autor.
\newblock {\em Einf, h rung in das Pr s entationswesen}.
\newblock Klein-Verlag, 1990.
\beamertemplatearticlebibitems
\bbitem{Jemand2000}
S.~Jemand.
\newblock On this and that.
\newblock {\em Journal of This and That}, 2(1):50--100, 2000.
\end{thebibliography}
\end{frame}
```

28.1.3. Style

Themes

The first solution is to use a built-in theme such as `Warsaw`, `Berlin`, etc. The second solution is to specify colors, inner themes and outer themes.

The Built-in solution

To the preamble you can add the following line:

```
\usetheme {Warsaw}
```

to use the "Warsaw" theme. `Beamer` has several themes, many of which are named after cities (e.g. `Barcelona`, `Madrid`, `Berlin`, etc.). Color themes, typically with animal names, can be specified with

```
\usecolortheme{beaver}
```

This Theme Matrix⁴ contains the various theme and color combinations included with beamer. For more customizing options, have a look to the official documentation included in your distribution of beamer, particularly the part *Change the way it looks*.

The list of all themes : Antibes, Bergen, Berkeley, Berlin, Boadilla, Copenhagen, Darmstadt, Dresden, Frankfurt, Goettingen, Hannover, Ilmenau, JuanLesPins, Luebeck, Madrid, Malmoe, Marburg, Montpellier, PaloAlto, Pittsburgh, Rochester, Singapore, Szeged, Warsaw, boxes, default

The *do it yourself* solution

4 <http://www.hartwork.org/beamer-theme-matrix/>

First you can specify the *outertheme*. The *outertheme* defines the head and the footline of each slide.

```
\useoutertheme{infolines}
```

Here is a list of all available outer themes

- infolines
- miniframes
- shadow
- sidebar
- smoothbars
- smoothtree
- split
- tree

Then you can add the *innertheme* :

```
\useinnertheme{rectangles}
```

Here is a list of all available inner themes :

- rectangles
- circles
- inmargin
- rounded

You can define the color of every element :

```
\setbeamercolor{alerted text}{fg=orange}
\setbeamercolor{background canvas}{bg=white}
\setbeamercolor{block body alerted}{bg=normal text.bg!90!black}
\setbeamercolor{block body}{bg=normal text.bg!90!black}
\setbeamercolor{block body example}{bg=normal text.bg!90!black}
\setbeamercolor{block title alerted}{use={normal text,alerted
text},fg=alerted text.fg!75!normal text.fg,bg=normal
text.bg!75!black}
\setbeamercolor{block title}{bg=blue}
\setbeamercolor{block title example}{use={normal text,example
text},fg=example text.fg!75!normal text.fg,bg=normal
text.bg!75!black}
\setbeamercolor{fine separation line}{}
\setbeamercolor{frametitle}{fg=brown}
```

```
\setbeamercolor{item projected}{fg=black}
\setbeamercolor{normal text}{bg=black,fg=yellow}
\setbeamercolor{palette sidebar primary}{use=normal text,fg=normal
text.fg}
\setbeamercolor{palette sidebar
quaternary}{use=structure,fg=structure.fg}
\setbeamercolor{palette sidebar
secondary}{use=structure,fg=structure.fg}
\setbeamercolor{palette sidebar tertiary}{use=normal text,fg=normal
text.fg}
\setbeamercolor{section in sidebar}{fg=brown}
\setbeamercolor{section in sidebar shaded}{fg=grey}
\setbeamercolor{separation line}{}
\setbeamercolor{sidebar}{bg=red}
\setbeamercolor{sidebar}{parent=palette primary}
\setbeamercolor{structure}{bg=black,fg=green}
\setbeamercolor{subsection in sidebar}{fg=brown}
\setbeamercolor{subsection in sidebar shaded}{fg=grey}
\setbeamercolor{title}{fg=brown}
\setbeamercolor{titlelike}{fg=brown}
```

Remember that you can define your own colors :

```
\definecolor{chocolate}{RGB}{33,33,33}
```

You can also define the style of blocks :

```
\setbeamertemplate{blocks}[rounded][shadow=true]  
\setbeamertemplate{background canvas}[vertical  
shading][bottom=white,top=structure.fg!25]  
\setbeamertemplate{sidebar canvas left}[horizontal  
shading][left=white!40!black,right=black]
```

You can also suppress the navigation bar:

```
\beamertemplatenavigationsymbolsempy
```

Fonts

You may also change the fonts for particular elements. If you wanted the title of the presentation as rendered by `\frame{\titlepage}` to occur in a serif font instead of the default sanserif, you would use:

```
\setbeamerfont{title}{family=\rm}
```

You could take this a step further if you are using OpenType fonts with Xe(La)TeX and specify a serif font with increased size and oldstyle proportional alternate number glyphs:

```
\setbeamerfont{title}{family=\rm\addfontfeatures{Scale=1.18,  
Numbers={Lining, Proportional}}}
```

Math Fonts

The default settings for `beamer` use a different set of math fonts than one

would expect from creating a simple math article. One quick fix for this is to use at the beginning of the file the option `mathserif`

```
\documentclass[mathserif]{beamer}
```

Others have proposed to use the command

```
\usefonttheme[onlymath]{serif}
```

but it is not clear if this works for absolutely every math character.

28.1.4. Frames Options

The *plain* option. Sometimes you need to include a large figure or a large table and you don't want to have the bottom and the top of the slides. In that case, use the plain option :

```
\frame[plain]{
```

```
...  
}
```

If you want to include lots of text on a slide, use the *shrink* option.

```
\frame[shrink]{  
...  
}
```

Before using any verbatim environment (like listings), you should pass the option containsverbatim to the \frame command.

```
\frame[containsverbatim]{  
\frametitle{Source code}  
  
\begin{lstlisting}[caption=First C example]  
int main()  
{  
    printf("Hello World!");  
    return 0;  
}
```

```
\end{lstlisting}  
}
```

28.1.5. Text animations

You can simply use the `\pause` statement :

```
\begin{frame}  
\frametitle{Some background}  
We start our discussion with some concepts.  
\pause  
The first concept we introduce originates with Erd\H os.  
\end{frame}
```

For text animations, for example in the `itemize` environment, you can write:

```
\begin{itemize}  
\item This one is always shown  
\item<1-> The first time  
\item<2-> The second time
```



```

\item<1-> Also the first time
\only<1-> This one is shown at the first time, but it will hide
soon.
\end{itemize}

\begin{frame}
\frametitle{'Hidden higher-order concepts?'}
\begin{itemize}[<+>]
\item The truths of arithmetic which are independent of PA in
some
higher-order}},
or infinitary, concepts'???)
\item 'Truths in the language of arithmetic which \ldots
\item That suggests stronger version of Isaacson's thesis
\end{itemize}
\end{frame}

```

28.1.6. Handout mode

In beamer class, the default mode is *presentation* which makes the slides. However, you can work in a different mode that is called *handout* by setting this option when calling the class:

```
\documentclass[12pt,handout]{beamer}
```

This mode is useful to see each slide only one time with all its stuff on it, making the `itemize<+>` to be there all at once (for instance, printable version). Nevertheless, this makes an issue when working with the only command, because its purpose is to have *only* some text or figures at a time and not all of them together.

If you want to solve this, you can add a statement to precise the behavior it must have when dealing with only commands in handout mode. Suppose you have a code like this

```
\only<1>{\includegraphics{pic1.eps}}  
\only<2>{\includegraphics{pic2.eps}}
```

These pictures being completely different, you want them both in the handout, but they cannot be both on the same slide since they are large. The solution is to add the handout statement to have the following:

```
\only<1|handout:1>{\includegraphics{pic1.eps}}  
\only<2|handout:2>{\includegraphics{pic2.eps}}
```

This will ensure the handout will make a slide for each picture.

Now imagine you still have your two pictures with the only statements, but the second one show the first one plus some other graphs and you don't need the first one to appear in the handout. You can thus precise the handout mode not to include some only commands by:

```
\only<1|handout:0>{\includegraphics{pic1.eps}}  
\only<2>{\includegraphics{pic2.eps}}
```

The command can also be used to hide frames, e.g.

```
\begin{frame}<handout:0>
```

or even, if you have written a frame that you don't want anymore but maybe you will need it later, you can write

```
\begin{frame}<0 | handout:0>
```

and this will hide your slide in both modes. (The order matters. Don't put `handout:0|beamer:0` or it won't work.)

A last word about the handout mode is about the notes. Actually, the full syntax for a frame is

```
\begin{frame}  
...  
\end{frame}  
\note{...}
```

```
\note{...}  
...
```

and you can write your notes about a frame in the field *note* (many of them if needed). Using this, you can add an option to the class calling, either

```
\documentclass[12pt,handout,notes=only]{beamer}
```

or

```
\documentclass[12pt,handout,notes=show]{beamer}
```

The first one is useful when you make a presentation to have only the notes you need, while the second one could be given to those who have followed your presentation or those who missed it, for them to have both the slides with what you said.

Note that the 'handout' option in the `\documentclass` line suppress all the animations.

Important: the *notes=only* mode is **literally** doing only the notes. This means there will be no output file but the DVI. Thus it requires you to have run the compilation in another mode before. If you use separate files for a better distinction between the modes, you may need to copy the .aux file from the handout compilation with the slides (w/o the notes).

28.1.7. Columns and Blocks

There are two handy environments for structuring a slide: "blocks", which divide the slide (horizontally) into headed sections, and "columns" which divides a slide (vertically) into columns.

Columns

Example

```

\begin{frame}
  \begin{columns}[c] % the "c" option specifies center vertical
    alignment
    \column{.5\textwidth} % column designated by a command
    Contents of the first column
    \column{.5\textwidth}
    Contents split \\ into two lines
  \end{columns}
\end{frame}

\begin{frame}
  \begin{columns}[t] % contents are top vertically aligned
  \begin{column}[T]{5cm} % each column can also be its own
environment
    Contents of first column \\ split into two lines
  \end{column}
  \begin{column}[T]{5cm} % alternative top-align that's better for
graphics
    \includegraphics[height=3cm]{graphic.png}
  \end{column}
  \end{columns}
\end{frame}

```

Blocks

Enclosing text in the *block* environment creates a distinct, headed block of text. This allows to visually distinguish parts of a slide easily. There are three basic types of block. Their formatting depends on the theme being used.

Simple

```
\begin{frame}

\begin{block}{This is a Block}
  This is important information
\end{block}

\begin{alertblock}{This is an Alert block}
This is an important alert
\end{alertblock}

\begin{exampleblock}{This is an Example block}
This is an example
\end{exampleblock}
```



```
\end{frame}
```

28.1.8. PDF options

You can specify the default options of your PDF.

```
\hypersetup{pdfstartview={FitH}} % By default the pdf fit the width  
of the screen.
```

28.2. The powerdot package

The powerdot package is available from CTAN⁵. The documentation⁶ explains the features in great detail.

The powerdot package is loaded by calling the `powerdot` class:

```
\documentclass{powerdot}
```

The usual header information may then be specified.

Inside the usual document environment, multiple `slide` environments specify the content to be put on each slide.

```
\begin{document}
```

5 <http://www.ctan.org/tex-archive/macros/latex/contrib/powerdot/>

6 <http://www.bitlib.net/mirror/ctan/macros/latex/contrib/powerdot/docs/powerdot.pdf>

```
\begin{slide}{This is the first slide}
  %Content goes here
\end{slide}
\begin{slide}{This is the second slide}
  %More content goes here
\end{slide}
% etc
\end{document}
```

28.3. References

28.4. Links

- [Wikipedia:Beamer \(LaTeX\)](#)⁷
- [Beamer user guide](#)⁸ (pdf) from CTAN

⁷ <http://en.wikipedia.org/wiki/Beamer%20%28LaTeX%29>

⁸ <http://www.ctan.org/tex-archive/macros/latex/contrib/beamer/doc/beameruserguide.pdf>

- The powerdot class⁹ (pdf) from CTAN
- A tutorial for creating presentations using beamer¹⁰

fr:LaTeX/Faire des présentations¹¹

9 <http://www.ctan.org/get/macros/latex/contrib/powerdot/doc/powerdot.pdf>

10 <http://www.math-linux.com/spip.php?article77>

11 <http://fr.wikibooks.org/wiki/LaTeX%2FFaire%20des%20pr%20E9sentations>

29. Teacher's Corner

29.1. Intro

LaTeX has specific features for teachers. We present the **exam class**¹ which is useful for designing exams and exercises with solutions. Interested people

¹ `examdoc` `{http://www-math.mit.edu/~psh/exam/examdoc.pdf}` Using the exam document class

could also have a look at the **probsoln** package² or the **mathexm** document class³.

29.2. The exam class

We present the **exam** class. The exam class is well suited to design exams with solutions. You just have to specify in the preamble if you want the solutions to be printed or not. You can also count the number of points.

-
- 2 **Probsoln** `{http://www.tex.ac.uk/tex-archive/macros/latex/contrib/probsoln/probsoln.pdf}` creating problem sheets optionally with solutions
 - 3 <http://mat140.bham.ac.uk/~richard/programming/tex/exams/msexdoc.pdf>

29.2.1. Preamble

In the preamble you can specify the following lines :

```
\documentclass[a4paper,11pt]{exam}
\printanswers % If you want to print answers
% \noprintanswers % If you don't want to print answers
\addpoints % if you want to count the points
% \noaddpoints % if you don't want to count the points
% Specifies the way question are displayed:
\qformat{\textbf{Question\thequestion}\quad(\thepoints)\hfill}
\usepackage{color} % defines a new color
\definecolor{SolutionColor}{rgb}{0.8,0.9,1} % light blue
\shadedolutions % defines the style of the solution environment
% \framedsolutions % defines the style of the solution environment
% Defines the title of the solution environment:
\r
```

```
newcommand{\solutiontitle}{\noindent\textbf{Solution:}}\par\noindent}
```

You can replace the 3 first lines with the following :

```
\documentclass[a4paper,11pt,answers,addpoints]{exam}
```


29.2.2. Document

- The exam is included in the **questions** environment.
- The command **\question** introduces a new question.
- The number of points is specified in squared brackets.
- The solution is given in the **solution** environment. It appears only if **\print-answers** or **answers** as an option of the **\documentclass** are specified in the preamble.

Here is an example :

```
\begin{questions} % Begins the questions environment
\question[2] What is the solution? % Introduces a new question which
is worth 2 points
\begin{solution}
Here is the solution
\end{solution}
\question[5] What is your opinion?
\begin{solution}
This is my opinion
```

```
\end{solution}  
\end{questions}
```

It is also possible to add stuff only if answers are printed using the **\ifprintanswers** command.

```
\ifprintanswers  
Only if answers are printed  
\else  
Only if answers are not printed  
\fi
```

29.2.3. Introduction

The macro **\numquestions** gives the total number of questions. The macro **\numpoints** gives the total number of points.

```
\begin{minipage}{.8\textwidth}
```

This exam includes \numquestions\ questions. The total number of

```
points is \numpoints.  
\end{minipage}
```

The backslash after `\numquestion` prevents the macro from gobbling the following whitespace as it normally would.

29.3. References

Part V.

Advanced Topics

30. General Guidelines

During this guide we have seen what it is possible to do and how this can be achieved, but the question is: I want to write a proper text with LaTeX, what to do then? Where should I start from? This is a short step-by-step guide about how to start a document properly, keeping a good high-level structure. This way it will be very easy to make modifications even when the document is almost finished. These are all just suggestions, but you might take inspiration from that to create your own document.

30.1. Project structure

Create a clear structure of the whole project this way:

1. create a directory only for the project. We'll refer to that in the following parts as the *root directory*
2. create two other directories inside the root, one for LaTeX documents, the other one for images. Since you'll have to write their name quite often, choose short names. A suggestion would be simply *tex* and *img*.
3. create your document (we'll call it `document.tex`, but you can use the name you prefer) and your own package (for example *mystyle.sty*); this second file will help you to keep the code cleaner.

If you followed all those steps, these files should be in your root directory, using "/" for each directory:

```
./document.tex
./mystyle.sty
```

```
./tex/  
./img/
```

nothing else.

30.2. Getting LaTeX to process multiple files

As your work grows, your LaTeX file can become unwieldy and confusing, especially if you are writing a long article with substantial, discrete sections, or a full-length book. In such cases it is good practice to split your work into several files. For example, if you are writing a book, it makes a lot of sense to write each chapter in a separate `.tex` file. LaTeX makes this very easy thanks to two commands:

```
\input{filename}
```


and

```
\include{filename}
```

The differences between these files will be explained below but what they have in common is that they process the contents of `filename.tex` before continuing with the rest of the base file. When the compiler processes your base file and reaches the command `\input` or `\include`, it reads `filename.tex` and processes its content in accordance with the formatting commands specified in the base file. This way you can put all the formatting options in your base file and then `\input` or `\include` the files which contain the actual content of your work. This means that the important part of your working process, i.e. writing, is kept largely separate from formatting choices (which is one of the main reasons why LaTeX is so good for serious writing!). You will thus be dealing solely with text and very basic commands such as `\section`, `\emph` etc. Your document will be uncluttered and much easier to work with.

The second method of including a file, `\include{filename}`, differs from the first in some important ways. You cannot nest `\include` statements within a file added via `\include`; `\input`, on the other hand, allows you to call files which themselves call other files, ad infinitum (well, nearly!). You can, however, `\include` a file which contains one or more `\input` commands. Please resist the temptation to nest files in this way simply because the system can do it: you will end up with just another kind of complexity!

A further important difference is that using `\include` will force a page break (which makes it ideal for a book's chapters), whereas the `\input` command does not (which in turn makes it ideal for, say, a long article with discrete sections, which of course are not normally set on a new page).

Working on discrete parts of your documents has consequences for how the base file is compiled; these will be dealt with below.

30.2.1. Using different paths

When the LaTeX compiler finds a reference to an external file in the base file, it will look for it in the same directory. However, you can in principle refer to any file on your system, using both absolute and relative paths.

An *absolute* path is a full path- and filename with every element specified. So, `filename.tex` might have the full path,

```
\input{/home/user/texfiles/filename.tex}
```

If you had created the directory `myfiles` for your writing project, in your `texfiles` directory, its full path would be,

```
\input{/home/user/texfiles/myfiles/filename.tex}
```

Obviously, using absolute paths is inefficient if you are referring to a file in the current directory. If, however, you need to include a file which is always kept

at a specific place in your system, you may refer to it with an absolute path, for example,

```
\input{/home/user/documents/useful/foo.tex}
```

In practice, an absolute file path is generally used when one has to refer to a file which is quite some way away in the file system (or perhaps even on a different server!). One word of warning: do not leave empty spaces in the filenames, they can cause ambiguous behaviour. Either leave no spaces or use underscores `_` instead.

You may, however, need to make your source portable (to another computer or to a different location of your harddisk), in which case *relative* paths should be used if you wish to avoid unnecessary re-writing of path names. Or, a relative path may simply be a more efficient and elegant way of referring to a file. A relative path is one which is defined in relation to the current directory, in our case the one which contains the base file. LaTeX uses the standard UNIX notation: with a simple dot `.` you refer to the current directory, and by two

dots .. you refer to the previous directory, that is the one above the current directory in the file system tree. The slash / is used to separate the different components of a pathname: directories and filenames. So by ./ you refer to the current directory, by ../ you refer to the previous directory, by ../../ you refer to a directory which is two steps upwards in the filesystem tree. Writing

```
\input{./filename.tex}
```

will have *exactly* the same effect as writing

```
\input{filename.tex}
```

but if you found it more convenient to put all your files in a sub-directory of your current directory, called `myfiles`, you would refer to that file by specifying

```
\input{./myfiles/filename.tex}
```

Indeed, in our example of the absolute path above, you could refer to that file relatively, too:

```
\input{../../documents/useful/foo.tex}
```

Of course, all commonly used file systems – Linux, Mac OS X and Windows – also feature the UNIX `./`, `../` facility outlined above. Do note, however, that LaTeX uses forward slashes / even on Microsoft Windows platforms, which use backslashes \ in pathnames. LaTeX implementations for Windows systems perform this conversion for you, which ensures that your document will be valid across all installations.

This flexibility, inherent in the way in which LaTeX is integrated with modern file systems, lets you input files in a way which suits your particular set-up.

30.2.2. Compiling the base file

When you compile your document, page references and the like will change according to your use of the `\input` and `\include` commands. Normally LaTeX users only run the compiler on parts of the document to check that an individual chapter is syntactically correct and looks as the writer intended. A full run is generally only performed for producing a full draft or the final version. In such cases, it is invariably necessary to run LaTeX twice or more to resolve all the page numbers, references, etc. (especially if you are using bibliographic software such as BibTeX, too).

The simplest way to check that one or more of the various components of your work is syntactically robust, is to comment out the command with a percentage sign, for example:

```
\documentclass{article}
\begin{document}
% \input{Section_1}
% \input{Section_2}
```

```
%\input{Section_3}  
\input{Section_4}  
%\input{Section_5}  
\end{document}
```

This code will process your base file with the `article` conventions but only the material in the file `Section_4.tex` will be processed. If that was, say, the last thing you needed to check before sending off to that major journal, you would then simply remove all the percentage signs and re-run LaTeX, repeating the compiling process as necessary to resolve all references, page numbers and so on.

30.2.3. Using `\includeonly`

Using this command provides more complex, and hence more useful possibilities. If you include the following command in your preamble, i.e. before `\begin{document}`,


```
\includeonly{filename1,filename2,...}
```

only the files specified between the curly braces will be included. Note that you can have one or more files as the argument to this command: separate them with a comma, no spaces.

This requires that there are `\include` commands in the document which specify these files. The filename should be written without the `.tex` file extension:

```
\documentclass{book}
\includeonly{Chapter_1,Chapter_4}
\begin{document}
\include{Chapter_1}   %Note that we are using Chapter_1 and not
Chapter_1.tex
\include{Chapter_2}
\include{Chapter_3}
\include{Chapter_4}
\end{document}
```

This code would process the base file but only include the content of the author's first and fourth chapters (`Chapter_1.tex` and `Chapter_4.tex`). Importantly, this alternative retains as much of the `.aux` information as possible from the previous run, so messes up your cross-references much less than the makeshift suggestion above.

30.2.4. Subfiles package

A disadvantage of using `\input` and `\include` is that only the "root" document can be compiled and not the "child" documents individually. The `package subfiles1` resolves this problem.

In the "root" document the package must be loaded as:

```
\usepackage{subfiles}
```

¹ <http://ctan.org/tex-archive/macros/latex/contrib/subfiles>

Instead of using `\input` and `\include`, "child" documents must be loaded as follows:

```
\subfile{filename}
```

The "child" documents must start with the following statements:

```
\documentclass[rootdocument.tex]{subfiles}  
\begin{document}
```

and end with:

```
\end{document}
```

In summary, root document (`main.tex`) looks like:

```
\documentclass{book}  
\usepackage{subfiles}  
\begin{document}
```

```
% my document content
\subfile{chapter1}
% more of my document content
\end{document}
```

and chapter 1 (chapter1.tex) looks like:

```
\documentclass[main.tex]{subfiles}
\begin{document}
%% my chapter 1 content
%%
%% more of my chapter 1 content
\end{document}
```

Some linux distributions, don't have subfiles package in their latex distributions. You can download subfiles.zip² to generate subfiles.cls and subfiles.sty files:

2 <http://tezcatal.fciencias.unam.mx/tex-archive/macros/latex/contrib/subfiles.zip>

```
wget http://mirror.ctan.org/macros/latex/contrib/subfiles.zip
unzip subfiles.zip
cd subfiles
latex subfiles.dtx
latex subfiles.ins
```

30.2.5. Inserting PDF files

If you need to insert an existing, possibly multi-page, PDF file into your LaTeX document, whether or not the included PDF was compiled with LaTeX or another tool, consider using the `pdfpages` package³. In the preamble, include the package:

```
\usepackage[final]{pdfpages}
```

³ <http://www.ctan.org/tex-archive/macros/latex/contrib/pdfpages/>

This package also allows you to specify which pages you wish to include: for example, to insert pages 3 to 6 from some file `insertme.pdf`, use:

```
\includepdf[pages=3-6]{insertme.pdf}
```

To insert the whole of `insertme.pdf`:

```
\includepdf[pages=-]{insertme.pdf}
```

For full functionality, compile the output with `pdflatex`.

30.3. The file `mystyle.sty`

Instead of putting all the packages you need at the beginning of your document as you could, the best way is to load all the packages you need inside another dummy package called *mystyle* you will create just for your document. The good point of doing this is that you will just have to add one single

`\usepackage` in your document, keeping your code much cleaner. Moreover, all the info about your style will be within one file, so when you will start another document you'll just have to copy that file and include it properly, so you'll have exactly the same style you have used.

Creating your own style is very simple: create a file called `mystyle.sty` (you could name it as you wish, but it has to end with ".sty"). Write at the beginning:

```
\ProvidesPackage{mystyle}
```

Then add all the packages you want with the standard command `\usepackage{...}` as you would do normally, change the value of all the variables you want, etc. It will work like the code you put here would be copied and pasted within your document.

For a list of several packages you can use, see the List of Packages⁴ section.

30.4. The main document document.tex

Then create a file called `document.tex`; this will be the main file, the one you will compile, even if you shouldn't need to edit it very often because you will be working on other files. It should look like this (it's the sample code for a *report*, but you might easily change it for an *article* or whatever else):

```
\documentclass[12pt,a4paper]{report}
\usepackage{graphicx}
\usepackage{ifpdf}
\ifpdf
    % put here packages only for the PDF:
    \DeclareGraphicsExtensions{.pdf,.png,.jpg,.mps}
    \usepackage{hyperref}
\else
    % put here packages only for the DVI:
\fi
% put all the other packages here:
\usepackage{mystyle}
```



```
\begin{document }

\input{./tex/title.tex}
% \maketitle
\tableofcontents
\listoffigures
\listoftables

\input{./tex/intro.tex}
\input{./tex/main_part.tex}
\input{./tex/conclusions.tex}

\appendix
\input{./tex/myappendix.tex}

% Bibliography:
\clearpage
\addcontentsline{toc}{chapter}{Bibliography}
\input{./tex/mybibliography.tex}

\end{document }
```

Here a lot of code expressed in previous sections has been used. At the beginning there is the header discussed in the Tips & Tricks⁵ section, so you will be able to compile in both DVI and PDF. Then you import the only package you need, that is your *mystyle.sty* (note that in the code it has to be imported without the extension), then your document starts. Then it inserts the title: we don't like the output of `\maketitle` so we created our own, the code for it will be in a file called `title.tex` in the folder called `tex` we created before. How to write it is explained in the Title Creation⁶ section. Then tables of contents, figure and tables are inserted. If you don't want them, just comment out those lines. Then the main part of the document is inserted. As you can see, there is no text in `document.tex`: everything is in other files in the `tex` directory so that you can easily edit them. We are separating our text from the structural code, so we are improving the "What You See is What You Mean" nature of LaTeX. Then we can see the appendix and finally

5 Chapter 38 on page 1009

6 Chapter 7.3.3 on page 259

the Bibliography. It is in a separated file and it is manually added to the table of contents using a tip suggested in the Tips & Tricks⁷.

Once you created your `document.tex` you won't need to edit it anymore, unless you want to add other files in the `tex` directory, but this is not going to happen very often. Now you can write your document separating it in as many files as you want and adding many pictures without getting confused: thanks to the rigid structure you gave to the project, you will be able to keep track of all your edits clearly.

A suggestion: do not call your files like "chapter_01.tex" or "figure_03.png", i.e. try to avoid using numbers in file-names: if the numbering LaTeX gives them automatically is different from the one you gave (and this will likely happen) you will get really confused. When naming a file, stop for a second, think about a short name that can fully explain what is inside the file without being ambiguous, it will let you save a lot of time as soon as the document gets larger.

30.5. Writing your document

While writing, whenever you have to take a decision about formatting, define your own command for it and add it to your `mystyle.sty`:`let LaTeX` work for you. If you do so, it will be very easy to change it if you change your mind. Here is an example: if you are writing a book about Mathematics and you have to use vectors, you have to decide how they will look. There are several different standards, used in many books. If a is a vector, some people like to add an arrow over it (\vec{a}), other people write it underlined (\underline{a}); another common version is to write it bold (**a**). Let us assume you want to write your vectors with an arrow over them; then add the following line in your `mystyle.sty`.

```
\newcommand{\myvec}[1]{\vec{#1}}
```

and write your vectors inside the new `\myvec{...}` command. You can call it as you wish, but you'd better choose a short name because you will probably write it very often. Then, if you change your mind and you want your vectors to look differently you just have to change the definition of your

`\myvec{...}`}. Use this approach whenever you can: this will save you a lot of time.

30.6. External Links

- Subfiles package documentation⁸
- pdfpages package documentation⁹

⁸ <http://tug.ctan.org/tex-archive/macros/latex/contrib/subfiles/subfiles.pdf>

⁹ <http://mirror.ctan.org/macros/latex/contrib/pdfpages/pdfpages.pdf>

31. Customizing LaTeX

Documents produced with the commands you have learned up to this point will look acceptable to a large audience. While they are not fancy-looking, they obey all the established rules of good typesetting, which will make them easy to read and pleasant to look at. However, there are situations where LaTeX does not provide a command or environment that matches your needs, or the output produced by some existing command may not meet your requirements.

In this chapter, I will try to give some hints on how to teach LaTeX new tricks and how to make it produce output that looks different from what is provided by default.

31.1. New commands

To add your own commands, use the

```
\newcommand{name}[num]{definition}
```

command. Basically, the command requires two arguments: the *name* of the command you want to create, and the *definition* of the command. The *num* argument in square brackets is optional and specifies the number of arguments the new command takes (up to 9 are possible). If missing it defaults to 0, i.e. no argument allowed.

The following two examples should help you to get the idea. The first example defines a new command called `\wba1` that will print "The Wikibook about LaTeX". Such a command could come in handy if you had to write the title of this book over and over again.

This is “The Wikibook about LaTeX” ... “The Wikibook about

```
\newcommand{\wbal}{The Wikibook about LaTeX}
This is "\wbal" \ldots{} "\wbal"
```

The next example illustrates how to define a new command that takes one argument. The #1 tag gets replaced by the argument you specify. If you wanted to use more than one argument, use #2 and so on, these arguments are added in an extra set of brackets.

- This is the Wikibook about LaTeX supported by Wikimedia
- This is the Wikibook about LaTeX supported by #1
- This is the Wikibook about LaTeX supported by #1 #2
- This is the Wikibook about LaTeX supported by John Doe

```

\newcommand{\wbalsup}[1] {
  This is the Wikibook about LaTeX
  supported by #1}
\newcommand{\wbalTwo}[2] {
  This is the Wikibook about LaTeX
  supported by #1 #2}
% in the document body:
\begin{itemize}
\item \wbalsup{Wikimedia}
\item \wbalsup{lots of users!}
\item \wbalTwo{John}{Doe}
\end{itemize}

```

Note: use `\wbalTwo`, not `\wbal2` (error on compiling)

LaTeX will not allow you to create a new command that would overwrite an existing one. But there is a special command in case you explicitly want this: `\renewcommand`. It uses the same syntax as the `\newcommand` command.

In certain cases you might also want to use the `\providecommand` command. It works like `\newcommand`, but if the command is already defined, LaTeX will silently ignore the new command.

With LaTeX2e, it is also possible to add a default parameter to a command with the following syntax:

```
\newcommand{name}[num][default]{definition}
```

If the default parameter of `\newcommand` is present, then the first of the number of arguments specified by `num` is optional with a default value of `default`; if absent, then all of the arguments are required.

- This is the Wikibook about LaTeX supported by Wikimedia

```

\newcommand{\wbalTwo}[2][Wikimedia]{
  This is the Wikibook about LaTeX
  supported by {#1} and {#2}!!}
% in the document body:
\begin{itemize}
\item \wbalTwo{John Doe}
\item \wbalTwo[lots of users]{John Doe}
\end{itemize}

```

- This is the Wikibook about LaTeX supported by lots of users and John Doe!

NOTE: when the command is used with an explicit first parameter it is given enclosed with brackets ("lots of users").

31.2. New Environments

Just as with the `\newcommand` command, there is a command to create your own environments. The `\newenvironment` command uses the following syntax:

```
\newenvironment{name}[num]{before}{after}
```

Again `\newenvironment` can have an optional argument. The material specified in the *before* argument is processed before the text in the environment gets processed. The material in the *after* argument gets processed when the `\end{name}` command is encountered.

The *num* argument is used the same way as in the `\newcommand` command. LaTeX makes sure that you do not define an environment that already exists. If you ever want to change an existing command, you can use the `\renewenvironment` command. It uses the same syntax as the `\newenvironment` command.

The example below illustrates the usage of the `\newenvironment` command:

```
\newenvironment{king}
{\rule{1ex}{1ex}\hspace{\stretch{1}}Figure 105
{\hspace{\stretch{1}}\rule{1ex}{1ex}}

\begin{king}
My humble subjects \ldots
\end{king}
```



31.2.1. Extra space

When creating a new environment you may easily get bitten by extra spaces creeping in, which can potentially have fatal effects. For example when you want to create a title environment which suppresses its own indentation as well

as the one on the following paragraph. The `\ignorespaces` command in the begin block of the environment will make it ignore any space after executing the begin block. The end block is a bit more tricky as special processing occurs at the end of an environment. With the `\ignorespacesafterend` LaTeX will issue an `\ignorespaces` after the special ‘end’ processing has occurred.

See the space
to the left.

```
\newenvironment{simple}%  
{\noindent}%  
{\par\noindent}
```

Same
here.

```
\begin{simple}
```

See the space to the left.

```
\end{simple}
```

Same here.

No space
to the left.

```
\newenvironment{correct}%  
{\noindent\ignorespaces}%  
{\par\noindent%  
\ignorespacesafterend}
```

Same
here.

```
\begin{correct}  
No space\to the left.  
\end{correct}  
Same\here.
```

Also, if you're still having problems with extra space being appended at the end of your environment when using the `\input` for external source, make sure there is no space between the beginning, sourcing, and end of the environment, such as:

```
\begin{correct}\input{somefile.tex}\end{correct}
```

31.3. Command-line LaTeX

If you work on a Unix-like OS, you might be using Makefiles or any kind of script to build your LaTeX projects. In that connection it might be interesting to produce different versions of the same document by calling LaTeX with command-line parameters. If you add the following structure to your document:

```
\usepackage{ifthen}
\ifthenelse{\equal{\blackandwhite}{true}}{
% "black and white" mode; do something..
}{
% "color" mode; do something different..
}
```

Now you can call LaTeX like this:


```
latex '\newcommand{\blackandwhite}{true}\input{test.tex}'
```

First the command `\blackandwhite` gets defined and then the actual file is read with input. By setting `\blackandwhite` to false the color version of the document would be produced.

31.4. Creating your own package

If you define a lot of new environments and commands, the preamble of your document will get quite long. In this situation, it is a good idea to create a LaTeX package containing all your command and environment definitions. You can then use the `\usepackage` command to make the package available in your document. Writing a package basically consists of copying the contents of your document preamble into a separate file with a name ending in `.sty`. It is very simple, just follow the steps:

1. create a simple text file called *mypack.sty* (or any other name you like) and open it with any text editor
2. at the very beginning of the text document just write

```
\ProvidesPackage {mypack}
```

- note: it has to have the same name of the file without the extension. It tells LaTeX the name of the package and will allow it to issue a sensible error message when you try to include a package twice.
3. write whatever you want in it using all the LaTeX commands you know. Normally you should define new commands or import other packages.
 4. import your new package with the known command

```
\usepackage {mypack}
```

5. or

```
\RequirePackage {mypack}
```

the file *mypack.sty* and the LaTeX source you are compiling must be in the same directory. It will be as though all you have written within your package is within the document itself.

Alternatively, it is possible to place the package within `~/texmf/tex/latex/mypack/mypack.sty` where `~` is your home directory (On Windows this is often `C:\Users\username`) and where *mypack* is the name of your package. Running `texhash` or equivalent will allow you to use your package as detailed above, but without it needing to be in the same directory as your document.

31.5. Creating your own style

It is also possible to create your own style file. The process is similar to the creation of your own package, you can call your own style file in the preamble of any document by the command:

```
\documentclass{mystyle}
```

The name of the style file is then *mystyle.cls* and can be opened with any text editor. At the beginning of this file, the following line has to be provided:

```
\ProvidesClass{mystyle}
```

again, within the style files other files or packages are imported by the `\quirepackage` command.

32. Packages

Add-on features for LaTeX are known as packages. Dozens of these are pre-installed with LaTeX and can be used in your documents immediately. They should all be stored in subdirectories of `texmf/tex/latex` named after each package. To find out what other packages are available and what they do, you should use the CTAN search page¹ which includes a link to Graham Williams' comprehensive package catalogue. A package is a file or collection of files containing extra LaTeX commands and programming which add new styling features or modify those already existing. Installed package files all end with `.sty` (there may be ancillary files as well). When you try to typeset

¹ <http://tug.ctan.org/search.html>

a document which requires a package which is not installed on your system, LaTeX will warn you with an error message that it is missing, and you can then download the package and install it using the instructions in the installing extra packages² section. You can also download updates to packages you already have (both the ones that were installed along with your version of LaTeX as well as ones you added). There is no limit to the number of packages you can have installed on your computer (apart from disk space!), but there is probably a physical limit to the number that can be used inside any one LaTeX document at the same time, although it depends on how big each package is. In practice there is no problem in having even a couple of dozen packages active.

2 <http://en.wikibooks.org/wiki/LaTeX%2FPackages%2FInstalling%20Extra%20Packages>

32.1. Using an existing package

To use a package already installed on your system, insert a `\usepackage` command in your document preamble with the package name in curly braces:

```
\usepackage{package_name}
```

For example, to use the `color` package, which lets you typeset in colors, you would type:

```
\documentclass[11pt, a4paper, oneside]{report}  
  
\usepackage{color}  
  
\begin{document}  
...  
\end{document}
```

You can include several package names in one `\usepackage` command by separating the names with commas, like this:


```
\usepackage{package1,package2,package3}
```

and you can have more than one `\usepackage` command. Some packages allow optional settings in square brackets. If you use these, you must give the package its own separate `\usepackage` command, like geometry shown below:

```
\documentclass[11pt,a4paper,oneside]{report}

\usepackage{pslatex,palatino,avant,graphicx,color}
\usepackage[margin=2cm]{geometry}

\begin{document}
\title{\color{red}Practical Typesetting}
\author{\color{blue}Name \ \ Work}
\date{\color{green}December 2005}
\maketitle

\end{document}
```

Many packages can have additional formatting specifications in optional arguments in square brackets, in the same way as `geometry` does. Read the

documentation for the package concerned to find out what can be done. You can pass several options together separated by a comma:

```
\usepackage[option1,option2,option3]{'package_name'}
```

32.2. Package documentation

To find out what commands a package provides (and thus how to use it), you need to read the documentation. In the `texmf/doc` subdirectory of your installation there should be directories full of `.dvi` files, one for every package installed. This location is distribution-specific, but is *typically* found in:

Distribution	Path
MiKTeX	C:\Program Files\MiKTeX 2.7\doc\latex
teTeX	/usr/share/texmf-tetex/doc

Generally, *most* of the packages are in the `latex` subdirectory, although other packages (such as BibTeX and font packages) are found in other subdirectories in `doc`. The documentation directories have the same name of the package (e.g. `amsmath`), which generally have one or more relevant documents in a variety of formats (`dvi`, `txt`, `pdf`, etc.). The documents generally have the same name as the package, but there are exceptions (for example, the documentation for `amsmath` is found at `latex/amsmath/amsmath.dvi`). If your installation procedure has not installed the documentation, the DVI files can all be downloaded from CTAN. Before using a package, you should read the documentation carefully, especially the subsection usually called "User Interface", which describes the commands the package makes available. You cannot just guess and hope it will work: you have to read it and find out.

On most Linux machines, you can usually automatically open any installed package documentation with the `texdoc` command:

```
texdoc <package-name>
```

32.3. Packages list

Here is a (not complete) list of useful packages that can be used for a wide range of different kind of documents. Each package has a short description next to it and, when available, there is a link to a section describing such package in detail. All of them (unless stated) should be included in your LaTeX distribution as *package_name.sty*. For more information, refer to the documentation of the single packages, as described in the previous section. The list is in alphabetical order.

amsmath	it contains the advanced math extensions for LaTeX. The complete documentation should be in your LaTeX distribution; the file is called <i>amstdoc</i> , and can be <i>dvi</i> or <i>pdf</i> . For more information, see the chapter about Mathematics ³
amssymb	it adds new symbols in to be used in math mode.
amsthm	it introduces the <code>proof</code> environment and the <code>theoremstyle</code> command. For more information see the Theorems ⁴ section.

3 Chapter 17 on page 501

4 Chapter 20 on page 667

array

it extends the possibility of LaTeX to handle tables, fixing some bugs and adding new features. Using it, you can create very complicated and customized tables. For more information, see the Tables⁵ section.

babel	<p>it provides the internationalization of LaTeX. It has to be loaded in any document, and you have to give as an option the main language you are going to use in the document. For more information see the Internationalization⁶ section.</p>
bm	<p>allows use of bold greek letters in math mode using the <code>\bm{...}</code> command. This supersedes the <code>amsbsy</code> package.</p>

boxedminipage	it introduces the <code>boxedminipage</code> environment, that works exactly like <code>minipage</code> but adds a frame around it
caption	allows customization of appearance and placement of captions for figures, tables, etc.

cancel

provides commands for striking out mathematical expressions. The syntax is

```
\cancel{x}
```

or

```
\cancelto{0}{x}
```

change page

to easily change the margins of pages. The syntax is

```
\change page {text height} {text width} %  
{even side margin} {odd side margin} %  
{column sep} {top margin} %  
{head height} {head sep} %  
{foot skip}
```

All the arguments can be both positive and negative numbers; they will be added (keeping the sign) to the relative variable.

cite

assists in citation management

color	it adds support for colored text. For more information, see the relevant section ⁷
easylist	adds support for arbitrarily-deep nested lists (useful for outlines).
esint	adds additional integral symbols, for integrals over squares, clockwise integrals over sets, etc.
eucal	other mathematical symbols
fancyhdr	to change header and footer of any page of the document. It is described in the Page Layout ⁸ section

7 <http://en.wikibooks.org/wiki/LaTeX%2FPackages%2FColor>

fontenc	to choose the font encoding of the output text. You might need it if you are writing documents in a language other than English. Check in the Internationalization ⁹ section.
geometry	for easy management of document margins and the document page size
glossaries	for creation of glossaries and list of acronyms. For more information, see relevant chapter ¹⁰ .
graphicx	to manage external pictures

9 Chapter 36 on page 953

10 Chapter 27.1.1 on page 757

hyperref	it gives LaTeX the possibility to manage links within the document or to any URL when you compile in PDF. For more information, see the relevant section ¹¹
indentfirst	once loaded, the beginning of any chapter/section is indented by the usual paragraph indentation.
inputenc	to choose the encoding of the input text. You might need it if you are writing documents in a language other than English. Check in the Internationalization ¹² section.
latexsym	other mathematical symbols

¹¹ <http://en.wikibooks.org/wiki/LaTeX%2FPackages%2FHyperref>

listings	to insert programming code within the document. Many languages are supported and the output can be customized. For more information, see the relevant section ¹³
mathrsfs	other mathematical symbols
natbib	gives additional citation options and styles
pdfpages	This package simplifies the insertion of external multi-page PDF or PS documents.

13 <http://en.wikibooks.org/wiki/LaTeX%2FPackages%2FListings>

<p>rotating</p>	<p>It lets you rotate any kind of object. It is particularly useful for rotating tables. For more information, see the relevant section¹⁴</p>
<p>setspace</p>	<p>Lets you change line spacing, e.g. provides the <code>\doublespacing</code> command for making double spaced documents. For more information, see the relevant section¹⁵</p>

14 <http://en.wikibooks.org/wiki/LaTeX%2FPackages%2FRotating>

showkeys

it is very useful while writing any document. If you want to reference an image or a formula, you have to give it a name using `\label{...}` and then you can recall it using `\ref{...}`. When you compile the document these will be replaced only with numbers, and you can't know which label you had used unless you take a look at the source. If you have loaded the `showkeys` package, you will see the label just next or above the relevant number in the compiled version. An example of a reference to a section is

in section [II.I. Moreover](#)

<p>showidx</p>	<p>it prints out all index entries in the left margin of the text. This is quite useful for proofreading a document and verifying the index. For more information, see the Indexing¹⁶ section.</p>
<p>subfiles</p>	<p>the "root" and "child" document can be compiled at the same time without making changes to the "child" document. For more information, see the Subfile package¹⁷ section.</p>

16 Chapter 27 on page 753

17 http://en.wikibooks.org/wiki/LaTeX%2FMultiple_files%23Subfiles_package

subfig

it allows to define multiple floats (figures, tables) within one environment giving individual captions and labels in the form 1a, 1b.

syntaxonly

if you add the following code in your preamble:

```
\usepackage{syntaxonly}
\syntaxonly
```

LaTeX skims through your document only checking for proper syntax and usage of the commands, but doesn't produce any (DVI or PDF) output. As LaTeX runs faster in this mode you may save yourself valuable time. If you want to get the output, you can simply comment out the second line.

textcomp	<p>provides extra symbols, e.g. arrows like <code>\textrightarrow</code>, various currencies (<code>\texteuro</code>,...), things like <code>\textcelsius</code> and many other</p>
theorem	<p>you can change the style of newly defined theorems. For more information see the Theorems¹⁸ section.</p>
todonotes	<p>lets you insert notes of stuff to do with the syntax <code>\todo{Add details.}</code></p>

siunitx¹⁹

helps you typeset of SI-units correctly. For example `\SI{12}{\mega\hertz}`. Automatically handles the correct spacing between the number and the unit. Note that even non-SI-units are set, like dB, rad, ...

ulem

it allows to underline text (either with straight or wavy line). Few examples of usage are added to the Formatting²⁰ chapter.

19 <http://ctan.org/tex-archive/macros/latex/contrib/siunitx>

20 <http://en.wikibooks.org/wiki/LaTeX%2FFormatting>

url

it defines the `\url{...}` command. URLs often contain special character such as `_` and `&`, in order to write them you should *escape* them inserting a backslash, but if you write them as an argument of `\url{...}`, you don't need to escape any special character and it will take care of proper formatting for you. If you are using the `hyperref`, you don't need to load `url` because it already provides the `\url{...}` command.

verbatim	it improves the <code>verbatim</code> environment, fixing some bugs. Moreover, it provides the <code>comment</code> environment, that lets you add multiple-line comments or easily comment out big parts of the code.
wrapfig	to insert images surrounded by text. It was discussed in section Floats, Figures and Captions ²¹
xypic	is used to create trees, graphs, (commutative) diagrams, and similar things.

32.4. Creating packages

See

- [LaTeX/Customizing LaTeX#Creating your own package](#)²²

32.5. External resources

The best way to look for LaTeX packages is the already mentioned CTAN: Search²³. Additional resources form The TeX Catalogue Online²⁴:

- [Alphabetic catalogue](#)²⁵

²² [Chapter 31.4 on page 854](#)

²³ <http://tug.ctan.org/search.html>

²⁴ <http://www.ctan.org/tex-archive/help/Catalogue/catalogue.html>

²⁵ <http://www.ctan.org/tex-archive/help/Catalogue/alpha.html>

- With brief descriptions²⁶
- Topical catalogue²⁷ with packages sorted systematically
- Hierarchical²⁸ mirroring the CTAN folder hierarchy

26 <http://www.ctan.org/tex-archive/help/Catalogue/brief.html>

27 <http://www.ctan.org/tex-archive/help/Catalogue/bytopic.html>

28 <http://www.ctan.org/tex-archive/help/Catalogue/hier.html>

33. Advanced Topics

Here are some topics that are not really necessary to write a proper document, but could help you making your life easier and giving you some details to modify.

__FORCETOC__

33.1. Adding your own counters

In LaTeX it is fairly easy to create new counters and even counters that reset automatically when another counter is increased (think subsection in a section for example). With the command

```
\newcounter{NameOfTheNewCounter}
```

you create a new counter that is automatically set to zero. If you want the counter to be reset to zero every time another counter is increased, use:

```
\newcounter{NameOfTheNewCounter} [NameOfTheOtherCounter]
```

To increase the counter, either use

```
\stepcounter{NameOfTheNewCounter}
```

or

```
\refstepcounter{NameOfTheNewCounter} % used for labels and cross  
referencing
```

or

```
\addtocounter{NameOfTheNewCounter}{number}
```

**here the number can also be negative. For automatic resetting you need to use
\stepcounter.**

To set the counter value explicitly, use

```
\setcounter{NameOfTheNewCounter}{number}
```

There is several way to get access to a counter.

- `\theNameOfTheNewCounter` will print the formatted string related to
the counter.

- `\value{NameOfTheNewCounter}` will return the counter value which can be used by other counter, or for calculus. It is not a formatted string, so it cannot be used in text.
- `\arabic{NameOfTheNewCounter}` will print the formatted counter using arabic numbers.

Instead of `\arabic` you could also use `\alph`, `\Alph`, `\roman`, or `\Roman`. Note that `\arabic{NameOfTheNewCounter}` may be used as a value too, but not the others.

Here follows an example where we want to use parts and sections, but no chapters in the book class :

```
\renewcommand{\thesection}{\thepart .\arabic{section}}
\part{My Part}
\section{My Section}
\subsection{My Subsection}
```

Here is an example for recreating something similar to a section and subsection counter that already exist in LaTeX:

```
\newcounter{mysection}
\newcounter{mysubsection}[mysection]
\addtocounter{mysection}{2} % set them to some other numbers than 0
\addtocounter{mysubsection}{10} % same
%
\arabic{mysection}.\arabic{mysubsection}
bla bla

\stepcounter{mysection}
\arabic{mysection}.\arabic{mysubsection}
bla bla

\stepcounter{mysubsection}
\arabic{mysection}.\arabic{mysubsection}
bla bla

\addtocounter{mysubsection}{25}
\arabic{mysection}.\arabic{mysubsection}
bla bla and more bla bla
```

33.2. Boxes

LaTeX builds up its pages by pushing around boxes. At first, each letter is a little box, which is then glued to other letters to form words. These are again glued to other words, but with special glue, which is elastic so that a series of words can be squeezed or stretched as to exactly fill a line on the page.

Admittedly, this is a very simplistic description of what really happens, but the point is that TeX operates with glue and boxes. Letters are not the only things that can be boxes. One can put virtually everything into a box, including other boxes. Each box will then be handled by LaTeX as if it were a single letter.

The past chapters have already dealt with some boxes, although they weren't described as such. The tabular environment and the `\includegraphics`, for example, both produce a box. This means that one can easily arrange two tables or images side by side. You just have to make sure that their combined width is not larger than the `\textwidth`.

You can also pack a paragraph of your choice into a box with either the

```
\parbox[pos]{width}{text}
```

command or the

```
\begin{minipage}[pos]{width} text \end{minipage}
```

environment. The *pos* parameter can take one of the letters *c*, *t* or *b* to control the vertical alignment of the box, relative to the baseline of the surrounding text. *width* takes a length argument specifying the width of the box. The main difference between a `minipage` and a `\parbox` is that you cannot use all commands and environments inside a `parbox`, while almost anything is possible in a `minipage`.

While `\parbox` packs up a whole paragraph doing line breaking and everything, there is also a class of boxing commands that operates only on horizontally aligned material. We already know one of them; it's called `\mbox`. It simply packs up a series of boxes into another one, and can be used to prevent LaTeX from breaking two words. As you can put boxes inside boxes, these horizontal box packers give you ultimate flexibility.


```
\makebox[width][pos]{text}
```

width defines the width of the resulting box as seen from the outside (This means it can be smaller than the material inside the box. You can even set the width to 0pt so that the text inside the box will be typeset without influencing the surrounding boxes). Besides the *length* expressions, you can also use `\width`, `\height`, `\depth`, and `\totalheight` in the width parameter. They are set from values obtained by measuring the typeset text. The *pos* parameter takes a one letter value: **center**, **flushleft**, **flushright**, or **spread** the text to fill the box.

The command `\framebox` works exactly the same as `\makebox`, but it draws a box around the text.

The following example shows you some things you could do with the `\makebox` and `\framebox` commands:

```

\makebox[\textwidth]{%
c e n t r a l}\par
\makebox[\textwidth][s]{%
s p r e a d}\par
\framebox[1.1\width]{Guess I'm
framed now!} \par
\framebox[0.8\width][r]{Bummer,
I am too wide} \par
\framebox[1cm][l]{never
mind, so am I}
Can you read this?

```

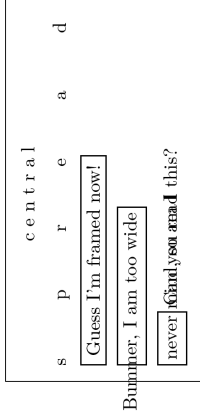


Figure 107

Now that we control the horizontal, the obvious next step is to go for the vertical. No problem for LaTeX. The

```

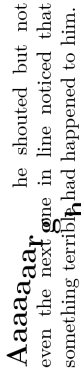
\raisebox{1lift}[extend-above-baseline][extend-below-baseline]{text}

```

command lets you define the vertical properties of a box. You can use `\width`, `\height`, `\depth`, and `\totalheight` in the first three parameters, in order to act upon the size of the box inside the text argument:

```
\raisebox{0pt}[0pt][0pt]{\Large%  
\textbf{Aaaa}\raisebox{-0.3ex}{a}%  
\raisebox{-0.7ex}{aa}%  
\raisebox{-1.2ex}{r}%  
\raisebox{-2.2ex}{g}%  
\raisebox{-4.5ex}{h}}
```

he shouted but not even the next
one in line noticed that something
terrible had happened to him.



Aaaaar he shouted but not
even the next one in line noticed that
something terrible had happened to him.

Figure 108

An alternative to these approaches is the usage of the **framed** environment (you will need to include the "framed" package to use it). This provides an easy way to box a paragraph within a document:

```
\begin{framed}
This is an easy way to box text within a document!
\end{framed}
```

33.3. Rules and Struts

The `\rule` command in normal use produces a simple black box:

```
\rule[lift]{width}{height}
```

Here is an example:

```
\rule{3mm}{.1pt} %  
\rule[-1mm]{5mm}{1cm} %  
\rule{3mm}{.1pt} %  
\rule[1mm]{1cm}{5mm} %  
\rule{3mm}{.1pt}
```

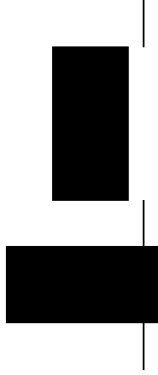


Figure 109

This is useful for drawing vertical and horizontal lines.

A special case is a rule with no width but a certain height. In professional typesetting, this is called a *strut*. It is used to guarantee that an element on a page has a certain minimal height. You could use it in a tabular environment to make sure a row has a certain minimum height.

34. Collaborative Writing of LaTeX Documents

Note: This Wikibook is based on the article *Tools for Collaborative Writing of Scientific LaTeX Documents*¹ by Arne Henningsen² that is published in *The PracTeX Journal* 2007, number 3 (<http://www.tug.org/pracjourn/>).

1 <http://tug.org/pracjourn/2007-3/henningsen/>

2 <http://en.wikibooks.org/wiki/User%3AArnehe>

34.1. Abstract

Collaborative writing of documents requires a strong synchronisation among authors. This Wikibook describes a possible way to organise the collaborative preparation of LaTeX documents. The presented solution is primarily based on the version control system *Subversion* (<http://subversion.apache.org/>). The Wikibook describes how *Subversion* can be used together with several other software tools and LaTeX packages to organise the collaborative preparation of LaTeX documents.

34.1.1. Other Methods

- Online LaTeX editor LIMSUP³ allowing real time collaboration of LaTeX documents.

- The online LaTeX editor ScribTeX⁴ makes sharing your document with others very easy. It provides a full LaTeX environment, with all the usual features of LaTeX like bibtex, images and custom style files. It also provides full version histories of your files, essential for collaborating. The free account allows only 3 projects and only one collaborator per project.
- ShareLaTeX.com⁵ is a new cloud based LaTeX editor designed to making working collaboratively as easy as possible. With real time collaborative editing similar to google documents it is currently in public beta.
- writelateX.com⁶ is like EtherPad⁷ for LaTeX. Start writing with one click and share the link. It supports real time preview, figures, bibliographies and custom styles. Currently in beta.
- publications.li⁸ is a real-time collaborative LaTeX editor.

4 <http://www.scribtex.com>

5 <https://www.sharelatex.com>

6 <http://writelatex.com>

7 <http://en.wikipedia.org/wiki/Etherpad>

8 <http://www.publications.li>

- Verbose⁹ is a professional Online LaTeX Editor that supports collaboration with other users and is free to use. Merge conflicts can easily be resolved by using a built-in merge tool that uses an implementation of the diff-algorithm to generate information required for a successful merge.
- The Monkey TeX¹⁰ is free and allows team sharing.
- Another option for collaboration is dropbox¹¹. It has 2Gb free storage and versioning system. Works like SVN, but more automated and therefore especially useful for beginning latex users.
- As the LaTeX system uses plain text, you can use synchronous collaborative editors like Gobby¹². In Gobby you can write your documents in collaboration with anyone in real time. It is strongly recommended that you use utf8 encoding (especially if there are users on multiple operating systems collaborating) and a stable network (typically wired networks).

9 <http://www.verbosus.com>

10 <http://monkeytex.bradcater.webfactional.com>

11 <http://www.getdropbox.com>

12 <http://en.wikipedia.org/wiki/Gobby>

- Google Documents¹³ or LaTeX Lab¹⁴ also allows real-time simultaneous collaborative editing of text files for anyone with a Google account (and its option to make the document available through a URL makes local download and compilation easily scriptable).
- TitanPad¹⁵ (or other clones¹⁶ of EtherPad¹⁷). To compile use the command line :

```
wget -O filename.tex  
"http://titanpad.com/ep/pad/export/xxxx/latest?format=txt" && (latex  
filename.tex)
```

where 'xxxx' should be replaced by the pad number (something like 'z7rSrfYcH').

-
- 13 <http://docs.google.com>
 - 14 <http://docs.latexlab.org>
 - 15 <http://titanpad.com>
 - 16 <http://etherpad.org/etherpadsites.html>
 - 17 <http://en.wikipedia.org/wiki/EtherPad>

- You could use more modern distributed version control systems like Mercurial¹⁸ or Git¹⁹.

34.2. Introduction

The collaborative preparation of documents requires a considerable amount of coordination among the authors. This coordination can be organised in many different ways, where the best way depends on the specific circumstances.

In this Wikibook, I describe how the collaborative writing of LaTeX documents is organised at our department (Division of Agricultural Policy, Department of Agricultural Economics, University of Kiel, Germany). I present our software tools, and describe how we use them. Thus, this Wikibook provides some ideas and hints that will be useful for other LaTeX users who prepare documents together with their co-authors.

18 <http://en.wikipedia.org/wiki/Mercurial>

19 <http://en.wikibooks.org/wiki/Git>

34.3. Interchanging Documents

There are many ways to interchange documents among authors. One possibility is to compose documents by interchanging e-mail messages. This method has the advantage that common users generally do not have to install and learn the usage of any extra software, because virtually all authors have an e-mail account. Furthermore, the author who has modified the document can easily attach the document and explain the changes by e-mail as well. Unfortunately, there is a problem when two or more authors are working at the same time on the same document. So, how can authors synchronise these files?

A second possibility is to provide the document on a common file server, which is available in most departments. The risk of overwriting each others' modifications can be eliminated by locking files that are currently edited. However, generally the file server can be only accessed from within a department. Hence, authors who are out of the building cannot use this method to update/commit their changes. In this case, they will have to use another way to overcome this problem. So, how can authors access these files?

A third possibility is to use a version control system. A comprehensive list of version control systems can be found at Wikipedia²⁰. Version control systems keep track of all changes in files in a project. If many authors modify a document at the same time, the version control system tries to merge all modifications automatically. However, if multiple authors have modified the same line, the modifications cannot be merged automatically, and the user has to resolve the conflict by deciding manually which of the changes should be kept. Authors can also comment their modifications so that the co-authors can easily understand the workflow of this file. As version control systems generally communicate over the internet (e.g. through TCP/IP connections), they can be used from different computers with internet connections. A restrictive firewall policy might prevent the version control system from connecting to the internet. In this case, the network administrator has to be asked to open the appropriate port. The internet is only used for synchronising the files. Hence, a

20 http://en.wikipedia.org/wiki/List_of_revision_control_software

permanent internet connection is not required. The only drawback of a version control system could be that it has to be installed and configured.

Moreover, a version control system is useful even if a single user is working on a project. First, the user can track (and possibly revoke) all previous modifications. Second, this is a convenient way to have a backup of the files on other computers (e.g. on the version control server). Third, this allows the user to easily switch between different computers (e.g. office, laptop, home).

34.4. The Version Control System *Subversion*

Subversion (SVN)²¹ comes as a successor to the popular version control system CVS. SVN operates on a client-server model in which a central server hosts a project repository that users copy and modify locally. A repository functions similarly to a library in that it permits users to check out the current project,

21 <http://subversion.apache.org/>

make changes, and then check it back in. The server records all changes a user checks in (usually with a message summarizing what changes the user made) so that other users can easily apply those changes to their own local files.

Each user has a local *working copy* of a remote *repository*. For instance, users can *update* changes from the repository to their working copy, *commit* changes from their own working copy to the repository, or (re)view the differences between working copy and repository.

To set up a SVN version control system, the SVN server software has to be installed on a (single) computer with permanent internet access. (If this computer has no static IP address, one can use a service like DynDNS²² to be able to access the server with a static hostname.) It can run on many Unix, modern MS Windows, and Mac OS X platforms.

Users do not have to install the SVN server software, but a SVN "client" software. This is the unique way to access the repositories on the server. Besides the basic SVN command-line client, there are several Graphical

User Interface Tools (GUIs) and plug-ins for accessing the SVN server (see <http://subversion.tigris.org/links.html>).²³ Additionally, there are very good manuals about SVN freely available on the internet (e.g. <http://svnbook.red-bean.com>).

At our department, we run the SVN server on a GNU-Linux system, because most Linux distributions include it. In this sense, installing, configuring, and maintaining SVN is a very simple task.

Most MS Windows users access the SVN server by the TortoiseSVN²⁴ client, because it provides the most usual interface for common users. Linux users usually use SVN utilities from the command-line, or eSvn²⁵--a GUI frontend--with KDiff3²⁶ for showing complex differences.

23 <http://subversion.tigris.org/links.html> .

24 <http://tortoisetsvn.tigris.org/>

25 <http://zoneit.free.fr/esvn/>

26 <http://kdiff3.sourceforge.net/>

34.5. Hosting LaTeX files in Subversion

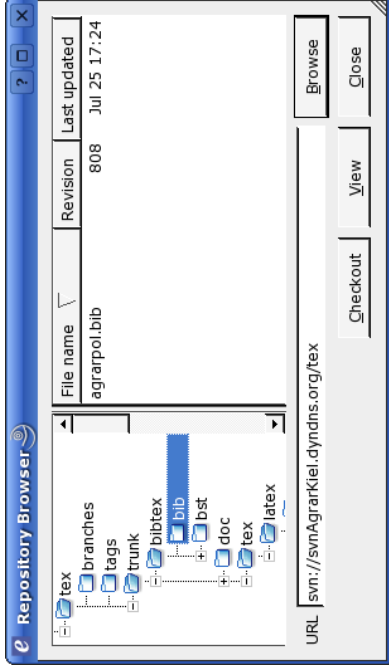


Figure 110 Figure 1: Common texmf tree shown in *eSvms Repository Browser*

On our *Subversion* server, we have one repository for a common `texmf` tree. Its structure complies with the **TeX Directory Structure** guidelines (TDS, <http://www.tug.org/tds/html>, see figure 1). This repository provides LaTeX classes, LaTeX styles, and BibTeX styles that are not available in the LaTeX distributions of the users, e.g. because they were bought or developed for the internal use at our department. All users have a working copy of this repository and have configured LaTeX to use this as their personal `texmf` tree. For instance, teTeX (<http://www.tug.org/tetex/>)²⁷ users can edit their TeX configuration file (e.g. `/etc/texmf/web2c/texmf.cnf`) and set the variable `TEXMFHOME` to the path of the working copy of the common `texmf` tree (e.g. by `TEXMFHOME = $HOME/texmf`); MiKTeX (<http://www.miktex.org/>)²⁸ users can add the path of the working copy of the common `texmf` tree in the 'Roots' tab of the MiKTeX Options.

If a new class or style file has been added (but not if these files have been modified), the users have to update their 'file name data base' (FNDB) before

27 <http://www.tug.org/tetex/>

28 <http://www.miktex.org/>

they can use these classes and styles. For instance, `TeX` users have to execute `texhash`; `MiKTeX` users have to click on the button 'Refresh FNDB' in the 'General' tab of the `MiKTeX Options`.

Furthermore, the repository contains manuals explaining the specific `LaTeX` software solution at our department (e.g. this document).

The *Subversion* server hosts a separate repository for each project of our department. Although branching, merging, and tagging is less important for writing text documents than for writing source code for software, our repository layouts follow the recommendations of the 'Subversion book' (<http://svnbook.red-bean.com>). In this sense, each repository has the three directories `/trunk`, `/branches`, and `/tags`.

The most important directory is `/trunk`. If a single text document belongs to the project, all files and subdirectories of this text document are in `/trunk`. If the project yields two or more different text documents, `/trunk` contains a subdirectory for each text document. A slightly different version (a **branch**) of a text document (e.g. for presentation at a conference) can be prepared

either in an additional subdirectory of `/trunk` or in a new subdirectory of `/branches`. When a text document is submitted to a journal or a conference, we create a **tag** in the directory `/tags` so that it is easy to identify the submitted version of the document at a later date. This feature has been proven very useful. When creating branches and tags, it is important always to use the *Subversion* client (and not the tools of the local file system) for these actions, because this saves disk space on the server and it preserves information about the same history of these documents.

Often the question arises, which files should be put under version control. Generally, all files that are directly modified by the user and that are necessary for compiling the document should be included in the version control system. Typically, these are the LaTeX source code (`*.tex`) files (the main document and possibly some subdocuments) and all pictures that are inserted in the document (`*.eps`, `*.jpg`, `*.png`, and `*.pdf` files). All LaTeX classes (`*.cls`), LaTeX styles (`*.sty`), BibTeX data bases (`*.bib`), and BibTeX styles (`*.bst`) generally should be hosted in the repository of the common `texmf` tree, but they could be included in the respective repository, if some

(external) co-authors do not have access to the common `texmf` tree. On the other hand, all files that are automatically created or modified during the compilation process (e.g. `*.aut`, `*.aux`, `*.bbl`, `*.bix`, `*.blg`, `*.dvi`, `*.gls`, `*.idx`, `*.ilg`, `*.ind`, `*.ist`, `*.lof`, `*.log`, `*.lot`, `*.nav`, `*.nlo`, `*.out`, `*.pdf`, `*.ps`, `*.snm`, and `*.toc` files) or by the (LaTeX or BibTeX) editor (e.g. `*.bak`, `*.bib~`, `*.kilepr`, `*.prj`, `*.sav`, `*.tcp`, `*.tmp`, and `*.tex~` files) generally should be **not** under version control, because these files are not necessary for compilation and generally do not include additional information. Furthermore, these files are regularly modified so that conflicts are very likely.

34.6. *Subversion* really makes the difference

A great feature of a version control system is that all authors can easily trace the workflow of a project by viewing the differences between arbitrary versions of the files. Authors are primarily interested in 'effective' modifications of

the source code that change the compiled document, but not in 'ineffective' modifications that have no impact on the compiled document (e.g. the position of line breaks). Software tools for comparing text documents ('diff tools') generally cannot differentiate between 'effective' and 'ineffective' modifications; they highlight both types of modifications. This considerably increases the effort to find and review the 'effective' modifications. Therefore, 'ineffective' modifications should be avoided.

In this sense, it is very important not to change the positions of line breaks without cause. Hence, automatic line wrapping of the users' LaTeX editors should be turned off and line breaks should be added manually. Otherwise, if a single word in the beginning of a paragraph is added or removed, all line breaks of this paragraph might change so that most diff tools indicate the entire paragraph as modified, because they compare the files line by line. The diff tools *wdiff* (<http://www.gnu.org/software/wdiff/>) and *dwdiff* (<http://os.ghalkes.nl/dwdiff.html>) are not affected by the positions of line breaks, because they compare documents word by word. However, their output is less clear so that modifications are more difficult to

track. Moreover, these tools cannot be used directly with the *Subversion* command-line switch `--diff-cmd`, but a small wrapper script has to be used (<http://textsnippets.com/posts/show/1033>).

A reasonable convention is to add a line break after each sentence and start each new sentence in a new line. Note that this has an advantage also beyond version control: if you want to find a sentence in your LaTeX code that you have seen in a compiled (DVI, PS, or PDF) file or on a printout, you can easily identify the first few words of this sentence and screen for these words on the left border of your editor window.

Furthermore, we split long sentences into several lines so that each line has at most 80 characters, because it is rather inconvenient to search for (small) differences in long lines. (Note: For instance, the LaTeX editor *Kile* (<http://kile.sourceforge.net/>) can assist the user in this task when it is configured to add a vertical line that marks the 80th column.) We find it very useful to introduce the additional line breaks at logical breaks of the sentence, e.g. before a relative clause or a new part of the sentence starts. An example LaTeX code that is formatted according to these guidelines is the source code of the

article *Tools for Collaborative Writing of Scientific LaTeX Documents* by Arne Henningsen²⁹ that is published (including the source code) in *The LaTeX Journal* 2007, Number 3 (<http://www.tug.org/pracjourn/2007-3/henningsen/>).

If the authors work on different operating systems, their LaTeX editors will probably save the files with different newline (end-of-line) characters (<http://en.wikipedia.org/wiki/Newline>). To avoid this type of 'ineffective' modifications, all users can agree on a specific newline character and configure their editor to use this newline character. Another alternative is to add the subversion property 'svn:eol-style' and set it to 'native'. In this case, *Subversion* automatically converts all newline characters of this file to the native newline character of the author's operating system (<http://svnbook.red-bean.com/en/1.4/svn.advanced.props.file-portability.html#svn.advanced.props.special.eol-style>).³⁰

²⁹ <http://en.wikibooks.org/wiki/User%3AArne>

³⁰ <http://svnbook.red-bean.com/en/1.4/svn.advanced.props.file-portability.html#svn.advanced.props.special.eol-style>.

There is also another important reason for reducing the number of 'ineffective' modifications: if several authors work on the same file, the probability that the same line is modified by two or more authors at the same time increases with the number of modified lines. Hence, 'ineffective' modifications unnecessarily increase the risk of conflicts (see section Interchanging Documents³¹).



Figure 111 Figure 2: Reviewing modifications in *KDiff3*

Furthermore, version control systems allow a very effective quality assurance measure: all authors should critically review their own modifications before they commit them to the repository (see figure 2). The differences between the user's working copy and the repository can be easily inspected with a single

Subversion command or with one or two clicks in a graphical *Subversion* client. Furthermore, authors should verify that their code can be compiled flawlessly before they commit their modifications to the repository. Otherwise, the co-authors have to pay for these mistakes when they want to compile the document. However, this directive is not only reasonable for version control systems but also for all other ways to interchange documents among authors.

Subversion has a feature called 'Keyword Substitution' that includes dynamic version information about a file (e.g. the revision number or the last author) into the contents of the file itself (see e.g. <http://svnbook.red-bean.com>,³² chapter 3). Sometimes, it is useful to include these information not only as a comment in the LaTeX source code, but also in the (compiled) DVI, PS, or PDF document. This can be achieved with the LaTeX packages *svn* (<http://www.ctan.org/tex-archive/macros/latex/contrib/svn/>), *svninfol* (<http://www.ctan.org/tex-archive/macros/latex/contrib/svninfol/>), or (preferably) *svn-multi* (<http://www.ctan.org/tex-archive/macros/latex/contrib/svn-multi/>).

32 <http://svnbook.red-bean.com>,

The most important directives for collaborative writing of LaTeX documents with version control systems are summarised in the following box.

Directives for using LaTeX with version control systems

1. Avoid 'ineffective' modifications.
2. Do not change line breaks without good reason.
3. Turn off automatic line wrapping of your LaTeX editor.
4. Start each new sentence in a new line.
5. Split long sentences into several lines so that each line has at most 80 characters.
6. Put only those files under version control that are directly modified by the user.
7. Verify that your code can be compiled flawlessly before committing your modifications to the repository.
8. Use *Subversions diff feature to critically review your modifications before committing them to the repository.*
9. Add a meaningful and descriptive comment when committing your modifications to the repository.

10. Use the *Subversion* client for copying, moving, or renaming files and folders that are under revision control.

If the users are willing to let go of the built-in *diff* utility of SVN and use *diff* tools that are local on their workstations, they can put to use such tools that are more tailored to text documents. The *diff* tool that comes with SVN was designed with source code in mind. As such, it is built to be more useful for files of short lines. Other tools, such as **Compare It!** allows to conveniently compare text files where each line can span hundreds of characters (such as when each line represents a paragraph). When using a *diff* tool that allows convenient views of files with long lines, the users can author the TeX files without a strict line-breaking policy.

34.6.1. Visualizing *diffs* in LaTeX: latexdiff and changebar

The tools `latexdiff`³³ and `changebar`³⁴ can visualize differences of two LaTeX files inside a generated document. This makes it easier to see impact of certain changes or discuss changes with people not custom to LaTeX. `Changebar` comes with a script `chbar.sh` which inserts a bar in the margin indicating parts that have changed. `Latexdiff` allows different styles of visualization. The default is that discarded text is marked as red and added text is marked as blue. It also supports a mode similar to `Changebar` which adds a bar in the margin. `Latexdiff` comes with a script `latexreverse` which can be used to accept or decline changes. It also has a wrapper script to support version control systems such as the discussed Subversion.

An example on how to use `Latexdiff` in the Terminal.

33 <http://www.ctan.org/tex-archive/support/latexdiff/>

34 <http://www.ctan.org/tex-archive/macros/latex/contrib/changebar/>

```
latexdiff old.tex new.tex > diff.tex
old.tex and new.tex are compared and the file visualizing the changes
is written to diff.tex
pdflatex diff.tex
showing the changes
```

Files

Create a PDF

The program DiffPDF³⁵ can be used to compare two existing PDFs visually. There is also a command line tool `comparepdf`³⁶ based on DiffPDF.

34.7. Managing collaborative bibliographies

Writing of scientific articles, reports, and books requires the citation of all relevant sources. BibTeX is an excellent tool for citing references and creating bibliographies (Markey 2005, Fenn 2006). Many different BibTeX styles can

35 <http://www.qtrac.eu/diffpdf.html>

36 <http://www.qtrac.eu/comparepdf.html>

be found on CTAN (<http://www.ctan.org>) and on the LaTeX Bibliography Styles Database (<http://jo.irisson.free.fr/bstdatabase/>). If no suitable BibTeX style can be found, most desired styles can be conveniently assembled with *custombib/makebst* (<http://www.ctan.org/tex-archive/macros/latex/contrib/custom-bib/>). Furthermore, BibTeX style files can be created or modified manually; however this action requires knowledge of the (unnamed) postfix stack language that is used in BibTeX style files (Patashnik 1988).

At our department, we have a common bibliographic data base in the BibTeX format (.bib file). It resides in our common `texmf` tree (see section 'Hosting LaTeX files in Subversion') in the subdirectory `/bibtex/bib/` (see figure 1). Hence, all users can specify this bibliography by only using the file name (without the full path) --- no matter where the user's working copy of the common `texmf` tree is located.

All users edit our bibliographic data base with the graphical BibTeX editor *JabRef* (<http://jabref.sourceforge.net/>). As *JabRef* is written in *Java*, it runs on all major operating systems. As different versions of *JabRef* generally save files in a slightly different way (e.g. by introducing line breaks at different

positions), all users should use the same (e.g. last stable) version of *JabRef*. Otherwise, there would be many differences between different versions of `.bib` files that solely originate from using different version of *JabRef*. Hence, it would be hard to find the real differences between the compared documents. Furthermore, the probability of conflicts would be much higher (see section 'Subversion really makes the difference'). As *JabRef* saves the BibTeX data base with the native newline character of the author's operating system, it is recommended to add the *Subversion* property 'svn:eol-style' and set it to 'native' (see section 'Subversion really makes the difference').



Figure 112 Figure 3: Specify default key pattern in *JabRef*

JabRef is highly flexible and can be configured in many details. We make the following changes to the default configuration of *JabRef* to simplify our work. First, we specify the default pattern for BibTeX keys so that *JabRef* can automatically generate keys in our desired format. This can be done by

selecting Options → Preferences → Key pattern and modifying the desired pattern in the field Default pattern. For instance, we use [auth:lower][shortyear] to get the last name of the first author in lower case and the last two digits of the year of the publication (see figure 3).



Figure 113 Figure 4: Set up general fields in *JabRef*

Second, we add the `BibTeX` field `location` for information about the location, where the publication is available as hard copy (e.g. a book or a copy of an article). This field can contain the name of the user who has the hard copy and where he has it or the name of a library and the shelf-mark. This field can be added in *JabRef* by selecting `Options` → `Set up general fields` and adding the word `location` (using the semicolon (;) as delimiter) somewhere in the line that starts with `General:` (see figure 4).



Figure 114 Figure 5: Specify 'Main PDF directory' in *JabRef*

Third, we put all PDF files of publications in a specific subdirectory in our file server, where we use the BibTeX key as file name. We inform *JabRef* about this subdirectory by selecting `Options` \rightarrow `Preferences` \rightarrow `External programs` and adding the path of the this subdirectory in the field `Main`

PDF directory (see figure 5). If a PDF file of a publication is available, the user can push the `Auto` button left of *JabRefs Pdf field* to automatically add the file name of the PDF file. Now, all users who have access to the file server can open the PDF file of a publication by simply clicking on *JabRefs Pdf icon*.

If we send the LaTeX source code of a project to a journal, publisher, or somebody else who has no access to our common `texmf` tree, we do not include our entire bibliographic data base, but extract the relevant entries with the Perl script *aux2bib* (<http://www.ctan.org/tex-archive/biblio/bibtex/utills/bibttools/aux2bib>).

34.8. Conclusion

This wikibook describes a possible way to efficiently organise the collaborative preparation of LaTeX documents. The presented solution is based on the

Subversion version control system and several other software tools and LaTeX packages. However, there are still a few issues that can be improved.

First, we plan that all users install the same LaTeX distribution. As the *TeX Live* distribution (<http://www.tug.org/texlive/>) is available both for Unix and MS Windows operating systems, we might recommend our users to switch to this LaTeX distribution in the future. (Currently, our users have different LaTeX distributions that provide a different selection of LaTeX packages and different versions of some packages. We solve this problem by providing some packages on our common `texmf` tree.)

Second, we consider to simplify the solution for a common bibliographic data base. Currently it is based on the version control system *Subversion*, the graphical BibTeX editor *JabRef*, and a file server for the PDF files of publications in the data base. The usage of three different tools for one task is rather challenging for infrequent users and users that are not familiar with these tools. Furthermore, the file server can be only accessed by local users. Therefore, we consider to implement an integrated server solution like *WIKINDX* (<http://wikindx.sourceforge.net/>), *Aigaion* (<http://www.aigaion.nl/>),

or *refBASE* (<http://refbase.sourceforge.net/>). Using this solution only requires a computer with internet access and a web browser, which makes the usage of our data base considerably easier for infrequent users. Moreover, the stored PDF files are available not only from within the department, but throughout the world. (Depending on the copy rights of the stored PDF files, the access to the server --- or least the access to the PDF files --- has to be restricted to members of the department.) Even Non-LaTeX users of our department might benefit from a server-based solution, because it should be easier to use this bibliographic data base in (other) word processing software packages, because these servers provide the data not only in BibTeX format, but also in other formats.

All readers are encouraged to contribute to this wikibook by adding further hints or ideas or by providing further solutions to the problem of collaborative writing of LaTeX documents.

34.9. Acknowledgements

Arne Henningsen thanks Francisco Reinaldo and Géraldine Henningsen for comments and suggestions that helped him to improve and clarify this paper, Karsten Heymann for many hints and advices regarding LaTeX, BibTeX, and *Subversion*, and Christian Henning as well as his colleagues for supporting his intention to establish LaTeX and *Subversion* at their department.

34.10. References

- Fenn, Jürgen (2006): Managing citations and your bibliography with BibTeX. The PracTeX Journal, 4. <http://www.tug.org/pracjourn/2006-4/fenn/>.

- Markey, Nicolas (2005): Tame the BeaST. The B to X of BibTeX. http://www.ctan.org/tex-archive/info/bibtex/tamethebeast/ttb_en.pdf.³⁷ Version 1.3.
- Oren Patashnik. Designing BibTeX styles. <http://www.ctan.org/tex-archive/info/biblio/bibtex/contrib/doc/btxhak.pdf>.
- Tools for collaborative paper-writing³⁸

37 http://www.ctan.org/tex-archive/info/bibtex/tamethebeast/ttb_en.pdf.

38 <http://mathoverflow.net/questions/3044/tools-for-collaborative-paper-writing>

35. Export To Other Formats

Strictly speaking, LaTeX source can be used to directly generate two formats:

- DVI using *latex*, the first one to be supported
- PDF using *pdflatex*, more recent

Using other software freely available on Internet, you can easily convert DVI and PDF to other document formats. In particular, you can obtain the PostScript version using software which is included in your LaTeX distribution. Some LaTeX IDE will give you the possibility to generate the PostScript version directly (even if it uses internally a DVI mid-step, e.g. LaTeX \rightarrow DVI \rightarrow PS). It is also possible to create PDF from DVI and vice versa. It doesn't seem

logical to create a file with two steps when you can create it straight away, but some users might need it because, as you remember from the first chapters, the format you can generate depends upon the formats of the images you want to include (EPS for DVI, PNG and JPG for PDF). Here you will find sections about different formats with description about how to get it.

Other formats can be produced, such as RTF (which can be used in Microsoft Word) and HTML. However, these documents are produced from software that parses and interprets the LaTeX files, and do not implement all the features available for the primary DVI and PDF outputs. Nonetheless, they do work, and can be crucial tools for collaboration with colleagues who do not edit documents with LaTeX.

35.1. Convert to PDF

35.1.1. Directly

```
pdflatex my_file
```

35.1.2. DVI to PDF

```
dvi2pdf my_file.dvi
```

will create `my_file.pdf`. Another way is to pass through PS generation:

```
dvi2ps myfile.dvi  
ps2pdf myfile.ps
```

you will get also a file called *my_file.ps* that you can delete.

35.1.1.3. Merging PDF

If you have created different PDF documents and you want to merge them into one single PDF file you can use the following command-line command. You need to have Ghostscript installed:

For Windows:

```
gswin32 -dNOPAUSE -sDEVICE=pdfwrite -sOUTPUTFILE=Merged.pdf -dBATCH  
1.pdf 2.pdf 3.pdf
```

For Linux:

```
gs -dNOPAUSE -sDEVICE=pdfwrite -sOUTPUTFILE=Merged.pdf -dBATCH 1.pdf  
2.pdf 3.pdf
```

Alternatively, PDF-Shuffler¹ is a small python-gtk application, which helps the user to merge or split pdf documents and rotate, crop and rearrange their pages using an interactive and intuitive graphical interface. This program may be available in your Linux distribution's repository.

Another option to check out is `pdftk2` (or PDF toolkit), which is a command-line tool that can manipulate PDFs in many ways. To merge one or more files, use:

```
pdftk 1.pdf 2.pdf 3.pdf cat output 123.pdf
```

Note: If you are merging external PDF documents into a Latex document which is compiled with `pdflatex`, a much simpler option is to use the `pdfpages` package, e.g.:

-
- 1 <http://pdfshuffler.sourceforge.net/>
 - 2 <http://www.accesspdf.com/>


```
\usepackage{pdfpages}
...
\includepdf [pages=--] {Document1.pdf}
\includepdf [pages=--] {Document2.pdf}
...
```

Three simple shell³ scripts using the pdfpages package are provided in the pdfjam bundle⁴ by D. Firth. They include options for merge several pdf (pdfjoin), put several pages in one physical sheet (pdfnup) and rotate pages (pdf90).

35.1.4. XeTeX

You can also use XeTeX (or, more precisely, XeLaTeX), which works in the same way as pdflatex: it creates a PDF file directly from LaTeX source. One

3 <http://en.wikipedia.org/wiki/Shell%20%28computing%29>

4 <http://www2.warwick.ac.uk/fac/sci/statistics/staff/academic/firth/software/pdfjam>

advantage of XeTeX over standard LaTeX is support for Unicode and modern typography. See its Wikipedia entry⁵ for more details.

Customization of PDF output in XeTeX (setting document title, author, keywords etc.) is done using the configuration of `hyperref`⁶ package.

35.2. Convert to PostScript

from PDF

```
pdf2ps my_file.pdf
```

from DVI

5 <http://en.wikipedia.org/wiki/XeTeX>

6 Chapter 14.3 on page 445

35.3. Convert to RTF

LaTeX can be converted into an RTF file, which in turn can be opened by a word processor such as OpenOffice.org Writer⁷ or Microsoft Word⁸. This conversion is done through `latex2rtf`⁹, which can run on any computer platform. The program operates by reading the LaTeX source, and mimicking the behaviour of the LaTeX program. `latex2rtf` supports most of the standard implementations of LaTeX, such as standard formatting, some math typesetting, inclusion of EPS, PNG or JPG graphics, and tables. As well, it has some

7 <http://en.wikipedia.org/wiki/OpenOffice.org%20Writer>

8 <http://en.wikipedia.org/wiki/Microsoft%20Word>

9 <http://latex2rtf.sourceforge.net/>

limited support for packages, such as `varioref`, and `natbib`. However, many other packages are not supported.

`latex2rtf` is simple to use. The Windows version has a GUI (`l2rshell.exe`), which is straightforward to use. The command-line version is offered for all platforms, and can be used on an example `mypaper.tex` file:

```
latex mypaper
bibtex mypaper # if you use bibtex
latex2rtf mypaper
```

Both `latex` and (if needed) `bibtex` commands need to be run *before* `latex2rtf`, because the `.aux` and `.bbl` files are needed to produce the proper output. The result of this conversion will create `myfile.rtf`, which you may open in many modern word processors such as Microsoft Word or Open Office.

35.4. Convert to HTML

There are many converters to HTML.

hevea

One option is the HEVEA¹⁰ program:

```
hevea mylatexfile
```

latex2html

```
latex2html -html_version 4.0, latin1, unicode -split 1 -nonavigation  
-noinfo -title "MyDocument" MyDocument.tex
```

TeX4ht

¹⁰ <http://hevea.inria.fr>

TeX4ht¹¹ is a very powerful conversion program, but its configuration is not straightforward. Basically a configuration file has to be prepared, and then the program is called.

bibtex2html

for BibTeX

```
bibtex2html mybibtexfile
```

35.5. Convert to image formats

In some cases it may be sufficient to simply copy a region of a PDF (or PS) file using the tools available in a PDF viewer (for example using LaTeX to typeset

¹¹ <http://www.cse.ohio-state.edu/~gurari/TeX4ht/>

a formula for pasting into a presentation). This however will not generally have sufficient resolution for whole pages or large areas.

On *nix the LaTeX package `preview` and the tool `dvipng` may be used to generate png output for small amounts of LaTeX code, for example to render equations for use in presentations. See `LaTeX/Tips_and_Tricks#Generate_png_screenshots`¹² for more details.

35.5.1. PNG

To convert from PDF, open your file with GIMP¹³. It will ask you which page you want to convert, whether you want to use anti-aliasing (choose *strong* if you want to get something similar to what you see on the screen). Try different resolutions to fit your needs, but 100 dpi should be enough. Once you have the image within GIMP, you can post-process it as you like and save it to any

12 Chapter 38.5.3 on page 1023

13 <http://en.wikibooks.org/wiki/GIMP>

format supported by GIMP, as PNG for example. A method for DVI files is `dvipng`¹⁴ (usage is the same as `dvipdfm`). Also, the "convert" command from the ImageMagick¹⁵ suite can convert both DVI and PDF files to PNG.

You can optimize the resulting image using `optipng`¹⁶ so that it will take up less space.

35.5.2. SVG

Direct conversion from PDF to SVG can be done using the command line tool `pdf2svg`¹⁷. (If you are using Linux or FreeBSD or MacPorts, `pdf2svg` might also be installable via the package installer.)

14 <http://savannah.nongnu.org/projects/dvipng/>

15 <http://www.imagemagick.org/>

16 <http://optipng.sourceforge.net/>

17 <http://www.cityinthesky.co.uk/pdf2svg.html>


```
pdf2svg file.pdf file.svg
```

Alternatively DVI or PDF can be converted to PS as described before, then the bash script `ps2svg.sh`¹⁸ can be used (as all the software used by this script is multiplatform, this is also possible in Windows, a step-by-step guide could be written).

One can also use `dvisvgm`¹⁹, an open source utility that converts from DVI to SVG.

18 http://en.wikipedia.org/wiki/Wikipedia:WikiProject_Electronics/Ps2svg.sh

19 <http://dvisvgm.sourceforge.net/>

35.6. Convert to plain text

If you are thinking of converting to plain text for spell-checking or to count words, there may be an easier way -- read *Tips and Tricks*²⁰ first.

Most LaTeX distributions come with `detex` program, which strips LaTeX commands. It can handle multi-file projects, so all you need is to give one command:

```
detex yourfile
```

(note the omission of `.tex` extension). This will output result to standard output. If you want the plain text go to a file, use

```
detex yourfile > yourfile.txt
```

If the output from `detex` does not satisfy you, you can try a newer version available on [Google Code](#)²¹, or use HTML conversion first and then copy text from your browser.

²¹ <http://code.google.com/p/opedetex/>

36. Internationalization

When you write documents in languages other than English, areas where LaTeX has to be configured appropriately:

1. LaTeX needs to know how to hyphenate the language(s) you are using.
2. You need to use language-specific typographic rules. In French for example, there is a mandatory space before each colon character (:).
3. You want to be able to insert all the language-specific special characters directly from your keyboard instead of using cumbersome coding (for example, type `ä` instead of `\ " { a }`).

If you simply need to add a few words from another language, you may find LaTeX/Accents¹ an easier way.

36.1. Hyphenating

The `babel` package by Johannes Braams will take care of everything. You can load it in your preamble, providing as an argument the language you want to use:

```
\usepackage[language]{babel}
```

You should place it soon after the `\documentclass` command, so that all the other packages you load afterwards will know the language you are using. A list of the languages built into your LaTeX system will be displayed every time the compiler is started. Babel will automatically activate the appropriate

hyphenation rules for the language you choose. If your LaTeX format does not support hyphenation in the language of your choice, babel will still work but will disable hyphenation, which has quite a negative effect on the appearance of the typeset document. Babel also specifies new commands for some languages, which simplify the input of special characters. See the sections about languages below for more information.

If you call babel with multiple languages:

```
\usepackage[languageA, languageB]{babel}
```

then the last language in the option list will be active (i.e. languageB), and you can use the command

```
\selectlanguage{languageA}
```

to change the active language. You can also add short pieces of text in another language using the command

```
\foreignlanguage{languageB}{Text in another language}
```

Babel also offers various environments for entering larger pieces of text in another language:

```
\begin{otherlanguage}{languageB}
Text in language B. This environment switches all language-related
definitions, like the language
specific names for figures, tables etc. to the other language.
\end{otherlanguage}
```

The starred version of this environment typesets the main text according to the rules of the other language, but keeps the language specific string for ancillary things like figures, in the main language of the document. The environment `hyphenrules` switches only the hyphenation patterns used; it can also be used to disallow hyphenation by using the language name 'nohyphenation'.

The [<ftp://ftp.tex.ac.uk/tex-archive/macros/latex/required/babel/babel.pdf>] babel manual] provides much more information on these and many other options.

36.2. Text encoding

Most of the modern computer systems allow you to input letters of national alphabets directly from the keyboard. In order to handle variety of input encoding used for different groups of languages and/or on different computer platforms LaTeX employs the `inputenc` package:

```
\usepackage[encoding]{inputenc}
```

`inputenc` package tells LaTeX what the text encoding format of your `.tex` files is. The encoding depends on your operating system but often a software's encoding can be changed from the settings (this happens at least with some editors, the PuTTY² terminal and TeXmaker³). You may choose whichever encoding you like, but you must say so in the preamble, so for example, if you prefer to use the ISO-8859-1, write

² <http://en.wikipedia.org/wiki/PuTTY>

³ <http://en.wikipedia.org/wiki/TeXmaker>


```
\usepackage[latin1]{inputenc}
```

Most modern operating systems use Unicode (utf-8) as a default encoding for text. On such system (for example Ubuntu) you can use:

```
\usepackage[utf8]{inputenc}
```

The supported encoding by the LaTeX team is `utf8` and covers a fairly specific/limited range of unicode input characters. It only defines those symbols that are known to be available with the current *font encoding*. `utf8x` is not officially supported, but covers a much broader range of input symbols.

You might encounter a situation where using `\usepackage[utf8]{inputenc}` might result in error:

```
! Package inputenc Error: Unicode char \u8:ũ not set up for use with LaTeX.
```

This is due to the utf8 definition not necessarily having a mapping of all the character glyphs you are able to enter on your keyboard. Such characters are for example `ÿ ũ Ū Ě Ī`. In such case, you need to use the `utf8x` option to define more character combinations. This might break up compatibility with some packages like `csquotes`.

When using the `inputenc` package, you should consider that other people might not be able to display your input files on their computer, because they use a different encoding. For example, the German umlaut ä on OS/2 is encoded as 132, on Unix systems using ISO-LATIN 1 it is encoded as 228, while in Cyrillic encoding cp1251 for Windows this letter does not exist at all; therefore you should use this feature with care. The following encodings may come in handy, depending on the type of system you are working on:

Operating system	Encodings	
	Western Latin	Cyrillic
Mac	<code>applemac</code>	<code>maccyr</code>
Unix	<code>latin1</code>	<code>koi8-ru</code>

Operating system	Encodings	
		Western Latin
Windows	ansinew	cp1251
DOS, OS/2	cp850	cp866nav

36.3. Output encoding

`fontenc` package tells LaTeX how to output the text you have produced. It defines at which position inside a TeX-font each letter is stored. Multiple input encodings could be mapped into one font encoding, which reduces number of required font sets. LaTeX can produce either bitmap-fonts (usually rasterized to 300 or 600 ppi) or scalable vector fonts (such as Type 1 fonts). There are many different font sets available containing different sets of glyphs (characters).

Font encoding is set with:

```
\usepackage[encoding]{fontenc}
```

where encoding is font encoding. It is possible to load several encodings simultaneously.

The default LaTeX font encoding is OT1, the encoding of the original Computer Modern TeX text fonts. It contains only 128 characters, many from ASCII, but leaving out some others and including a number that are not in ASCII. When accented characters are required, TeX creates them by combining a normal character with an accent. While the resulting output looks perfect, this approach stops the automatic hyphenation from working inside words containing accented characters. Besides, some of Latin letters could not be created by combining a normal character with an accent, to say nothing about letters of non-Latin alphabets, such as Greek or Cyrillic.

To overcome these shortcomings, several 8-bit CM-like font sets were created. *Extended Cork* (EC) fonts in T1 encoding contains letters and punctuation characters for *most of the European languages* based on Latin script. The

LH font set contains letters necessary to typeset documents in languages using Cyrillic script. Because of the large number of Cyrillic glyphs, they are arranged into four font encodings—T2A, T2B, T2C, and X2. The CB bundle contains fonts in LGR encoding for the composition of Greek text. By using these fonts you can improve/enable hyphenation in non-English documents. Another advantage of using new CM-like fonts is that they provide fonts of CM families in all weights, shapes, and optically scaled font sizes

Here is a collection of suggestions about writing a LaTeX document in a language other than English. If you have experience in a language not listed below, please add some notes about it.

36.4. Hyphenating languages

36.4.1. Arabic script

For languages which use the Arabic script, including Arabic, Persian, Urdu, Pashto, Kurdish, Uyghur, etc., add the following code to your preamble:

```
\usepackage{arabtext}
```

You can input text in either romanized characters or native Arabic script encodings. Use any of the following commands/environment to enter in text:

```
\< . . . >  
\RL{ . . . }  
\begin{arabtext} . . . \end{arabtext}.
```

See the ArabTeX⁴ Wikipedia article for further details.

⁴ <http://en.wikipedia.org/wiki/ArabTeX>

You may also use the Arabi package within babel to typeset Arabic and Persian

```
\usepackage[LAE,LFE]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[farsi,arabic]{babel}
```

You may use Arabi with Lyx, or with tex4ht to produce HTML. You may also copy and paste from PDF files produced with Arabi thanks to the support of the cmap package.

See Arabi page on CTAN⁵

36.4.2. Persian script

For Persian language, there is a dedicated package called XePersian which uses XeLaTeX as the typesetting engine. Just add the following code to your preamble:

⁵ <http://tug.ctan.org/tex-archive/language/arabic/arabi/>

```
\usepackage{xepersian}
```

See [XePersian page on CTAN](#)⁶

Moreover, Arabic script can be used to type Persian as illustrated in the previous subsection.

36.4.3. Cyrillic script

Please add the section "Writing in Cyrillic" from <http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf>⁷. You are allowed to copy it.

See also the Bulgarian translation of the "Not so Short Introduction to LaTeX 2e" from <http://www.ctan.org/tex-archive/info/lshort/bulgarian/lshort-bg.pdf>

⁶ <http://tug.ctan.org/tex-archive/macros/xetex/latex/xepersian/>

⁷ <http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf>

This enables you to type cyrillic letters directly via your keyboard, but with a different distribution than a standard cyrillic keyboard! To get the standard distribution, only include: `\usepackage[OT1]{fontenc}`
`\usepackage[russian]{babel}`

36.4.4. Czech

Czech is fine using

```
\usepackage[czech]{babel}  
\usepackage[T1]{fontenc}  
\usepackage[utf8]{inputenc}
```

You may use different encoding, but UTF-8 is becoming standard and it allows you to have „czech quotation marks“ directly in your text. Otherwise, there are macros `\glqq` and `\grqq` to produce left and right quote or place quoted text “`\uv{here}`”.

36.4.5. Finnish

Finnish language hyphenation is enabled with:

```
\usepackage[finnish]{babel}
```

This will also automatically change document language (section names, etc.) to Finnish.

Remember to use Unicode encoding for Finnish if you're using an editor in utf8 mode:

```
\usepackage[utf8]{inputenc}
```

The default encoding system can often be changed regardless of the operating system⁸ (this happens at least with some editors, the PuTTY⁹ terminal and

⁸ <http://en.wikipedia.org/wiki/Operating%20system>

⁹ <http://en.wikipedia.org/wiki/PuTTY>

TeXmaker¹⁰). You may choose whichever encoding you like, but you must say so in the preamble, so for example, if you prefer to use the ISO-8859-1, write instead

```
\usepackage[latin1]{inputenc}
```

The encoding is important and makes sure you can simply write the *ääkköset* ö ä å Ö Ä Å as such, which is less cumbersome than having to write

```
\"{a}
```

to get the letter ä. Actually letters like ñ ô ó Ò ñ work as well so the same idea applies also to other European languages like Spanish¹¹.

If you want to use *European Computer Modern* fonts, you should use:

```
\usepackage[T1]{fontenc}
```

10 <http://en.wikipedia.org/wiki/TeXmaker>

11 Chapter 36.4.16 on page 992

For creating scalable (vector) Type1 fonts instead of bitmapped fonts, you can substitute the line above with:

```
\usepackage{ae}
```

The ae-package changes encoding to T1 and also loads the scalable version of *Almost European Computer Modern* fonts.

36.4.6. French

Some hints for those creating French documents with LaTeX: you can load French language support with the following command:

```
\usepackage[french]{babel}  
\usepackage[latin1]{inputenc}  
\usepackage[T1]{fontenc}
```

There are multiple options for typesetting French documents, depending on the flavor of French: `french`, `frenchb`, and `français` for Parisian French, and `acadian` and `canadien` for new-world French. All enable French hyphenation, if you have configured your LaTeX system accordingly. All of these also change all automatic text into French: `\chapter` prints *Chapitre*, `\today` prints the current date in French and so on. A set of new commands also becomes available, which allows you to write French input files more easily. Check out the following table for inspiration:

input code	rendered output
<code>\og guillemets \fg{}</code>	« guillemets »
<code>M\up{me}, D\up{r}</code>	M ^{me} , D ^r
<code>1\ier{} , 1\iere{} , 1\ieres{}</code>	1 ^{er} , 1 ^{re} , 1 ^{res}
<code>2\ieme{} 4\iemes{}</code>	2 ^e 4 ^{es}
<code>\No 1, \no 2</code>	N ^o 1, n ^o 2
<code>20~\degres C, 45\degres</code>	20 °C, 45°
<code>M. \bsc{Durand}</code>	M. Durand

input code	rendered output
<code>\nombre{1234,56789}</code>	1 234,567 89
<code>\OEuvre or \OE{ }uvre and \oeuvre</code>	Œuvre et œuvre
<code>re\c coivent or re\c{c}oivent</code>	reçoivent
<code>héro\ " isme</code>	héroïsme

You will also notice that the layout of lists changes when switching to the French language. For more information on what the *french* option of babel does and how you can customize its behavior, run LaTeX on file frenchb.dtx and read the produced file frenchb.pdf or frenchb.dvi.

36.4.7. German

You can load German language support using *either one* of the two following commands.

For old German orthography use

```
\usepackage[german]{babel}
```

or for new German orthography use

```
\usepackage[ngerman]{babel}
```

This enables German hyphenation, if you have configured your LaTeX system accordingly. It also changes all automatic text into German. Eg. “Chapter” becomes “Kapitel.” A set of new commands also becomes available, which allows you to write German input files more quickly even when you don’t use the inputenc package. Check out the table below for inspiration. With inputenc, all this becomes moot, but your text also is locked in a particular encoding world.

German Special Characters.	
"a	ä

German Special Characters.

"s	ß
"^ or \glqq	”
"' or \grqq	“
"< or \flqq	«
"> or \frqq	»
\flq	<
\frq	>
\dq	"

In German books you often find French quotation marks («guillemets»). German typesetters, however, use them differently. A quote in a German book would look like »this«. In the German speaking part of Switzerland, typesetters use «guillemets» the same way the French do. A major problem arises from the use of commands like \flq: If you use the OT1 font (which is the default font) the guillemets will look like the math symbol "«", which turns a typesetter's stomach. T1 encoded fonts, on the other hand, do contain the

required symbols. So if you are using this type of quote, make sure you use the T1 encoding. (`\usepackage[T1]{fontenc}`)

36.4.8. Greek

This is the preamble you need to write in the Greek language.

```
\usepackage[english,greek]{babel}  
\usepackage[iso-8859-7]{inputenc}
```

This preamble enables hyphenation and changes all automatic text to Greek. A set of new commands also becomes available, which allows you to write Greek input files more easily. In order to temporarily switch to English and vice versa, one can use the commands `\textlatin{english text}` and `\textgreek{greek text}` that both take one argument which is then typeset using the requested font encoding. Otherwise you can use the command `\selectlanguage{...}` described in a previous section. Use `\euro` for the Euro symbol.

36.4.9. Hungarian

Similar to Italian, but use the following lines:

```
\usepackage[magyar]{babel}  
\usepackage[latin2]{inputenc}  
\usepackage[T1]{fontenc}
```

- More information in [hungarian](#)¹².

The Hungarian version of BaBeL included with standard LaTeX distribution is not perfect, a much better version can be downloaded from the previous page.

36.4.10. Icelandic / Faroese

The following lines can be added to write Icelandic text:

12 <http://www.math.bme.hu/latex/>

```
\usepackage[icelandic]{babel}
\usepackage[T1]{fontenc}
```

and for some users

```
\usepackage[utf8]{inputenc}
```

is needed. This allows the user to write with Icelandic characters and changes text like the *Abstract* in

```
\begin{abstract}
Þetta er útdráttur.
\end{abstract}
```

into "Útdráttur" and turns *Partinto* "*Hluti*".

Icelandic Special Characters.	
" ` or \glqq	" "
\grqq	" "

Icelandic Special Characters.

<code>\TH</code>	Þ
<code>\th</code>	þ
<code>\" {O}</code>	Ö
<code>\" {o}</code>	ö
<code>\AE</code>	Æ
<code>\ae</code>	æ
<code>\DH</code>	Ð
<code>\dh</code>	ð

36.4.11. Italian

Italian is well supported by LaTeX. Just add `\usepackage[italian]{babel}` at the beginning of your document and the output of all the commands will be translated properly. You can add letters with accents without any particular setting, just write `\'a` `\'e` `\'e` `\'i` `\'o` `\'u` and you will get `à` `é` `è` `ì` `ò` `ù` (NB: the symbol changes if the inclination of the accent changes). Anyway,

if you do so, it could be quite annoying since it's time-wasting. Moreover, if you are using any spell-checking program, "città" is correct, but "citt\`a" will be seen as a mistake. If you add `\usepackage[latin1]{inputenc}` at the beginning of your document, LaTeX will include correctly all your accented letters. To sum up, just add

```
\usepackage[italian]{babel}  
\usepackage[latin1]{inputenc}
```

at the beginning of your document and you can write in Italian without being worried of translations and fonts. If you are writing your document without getting any error, then don't worry about anything else. If you start getting some unknown errors whenever you use an Italian letter, then you have to worry about the encoding of your files. As known, any LaTeX source is just plain text, so you'll have to insert accented letters properly within the text file. If you write your document using always the same program on the same computer, you should not have any problem. If you are writing your document using different programs, it could start getting some strange errors from the

compiler. The reason could be that the accented letters were not included properly within your source file and LaTeX can't recognize them. The reason is that an editor modified your document with a different encoding from the one that was used when creating it. Most of the operating systems use UTF-8 as default, but this could create problems if are using programs based on different libraries or different operating systems. The best way to solve this problem is to change the encoding to ISO-8859-1, that includes all the letters you need. Some text editors let you change the encoding in the settings.

36.4.12. Korean

To use LaTeX for typesetting Korean, we need to solve three problems:

1. We must be able to edit Korean input files. Korean input files must be in plain text format
 - a) *Korean using Unicode*. These days all three major operating systems (Mac OS, Unix, Windows) come equipped with pretty decent

multilingual support and internationalization features so that editing Korean text file is not so much of a problem anymore, even on non-Korean operating systems. In your text editor (not your word processor), go to "Settings" and choose UTF-8. Set the preamble in latex to UTF-8 as well (*usepackage* with values *ucs* and *utf8x*). Open your Korean keyboard from the keyboard menu, and start writing.

b) *Korean using older formats.* The two most widely used encodings for Korean text files are EUC-KR and its upward compatible extension used in Korean MS-Windows, CP949/Windows-949/UHC. In these encodings each US-ASCII character represents its normal ASCII character similar to other ASCII compatible encodings such as ISO-8859-x, EUC-JP, Big5, or Shift_JIS. On the other hand, Hangeul syllables, Hanjas (Chinese characters as used in Korea), Hangeul Jamos, Hiraganas, Katakanas, Greek and Cyrillic characters and other symbols and letters drawn from KS X 1001 are represented by two consecutive octets. The first has its MSB set. Until the mid-1990's, it took a considerable amount of time and ef-

fort to set up a Korean-capable environment under a non-localized (non-Korean) operating system. You can skim through the now much-outdated <http://jshin.net/faq>¹³ to get a glimpse of what it was like to use Korean under non-Korean OS in mid-1990's.

2. TEX and LATEX were originally written for scripts with no more than 256 characters in their alphabet. To make them work for languages with considerably more characters such as Korean or Chinese, a subfont mechanism was developed. It divides a single CJK font with thousands or tens of thousands of glyphs into a set of subfonts with 256 glyphs each. For Korean, there are three widely used packages; HLA_{TEX} by UN Koaunghi, hL_{ATEX}p by CHA Jaecheon and the CJK package by Werner Lemberg. HL_{ATEX} and hL_{ATEX}p are specific to Korean and provide Korean localization on top of the font support. They both can process Korean input text files encoded in EU_C-KR. HL_{ATEX} can even process input files encoded in CP949/Windows-949/UHC and UTF-8 when used along with Λ , Ω . The CJK package is not specific to

13 <http://jshin.net/faq>

Korean. It can process input files in UTF-8 as well as in various CJK encodings including EUC-KR and CP949/Windows-949/UHC, it can be used to typeset documents with multilingual content (especially Chinese, Japanese and Korean). The CJK package has no Korean localization such as the one offered by H_LA_TE_X and it does not come with as many special Korean fonts as H_LA_TE_X.

3. The ultimate purpose of using typesetting programs like T_EX and L_A-T_EX is to get documents typeset in an ‘aesthetically’ satisfying way. Arguably the most important element in typesetting is a set of well-designed fonts. The H_LA_TE_X distribution includes U_HC PostScript fonts of 10 different families and Munhwabu fonts (TrueType) of 5 different families. The CJK package works with a set of fonts used by earlier versions of H_LA_TE_X and it can use Bitstream’s cyberbit True-Type font.

To use the H_LA_TE_X package for typesetting your Korean text, put the following declaration into the preamble of your document:

```
\usepackage{hangu1}
```

This command turns the Korean localization on. The headings of chapters, sections, subsections, table of content and table of figures are all translated into Korean and the formatting of the document is changed to follow Korean conventions. The package also provides automatic “particle selection.” In Korean, there are pairs of post-fix particles grammatically equivalent but different in form. Which of any given pair is correct depends on whether the preceding syllable ends with a vowel or a consonant. (It is a bit more complex than this, but this should give you a good picture.) Native Korean speakers have no problem picking the right particle, but it cannot be determined which particle to use for references and other automatic text that will change while you edit the document. It takes a painstaking effort to place appropriate particles manually every time you add/remove references or simply shuffle parts of your document around. H_LA_TE_X relieves its users from this boring and error-prone process.

In case you don't need Korean localization features but just want to typeset Korean text, you can put the following line in the preamble, instead.

```
\usepackage{hfont}
```

For more details on typesetting Korean with H_LA_TE_X, refer to the H_LA_TE_X Guide. Check out the web site of the Korean T_EX User Group (KTUG) at <http://www.ktug.or.kr/>.

36.4.13. Polish

If you plan to use Polish in your utf-8 encoded document, use the following code

```
\usepackage[utf8]{inputenc}  
\usepackage{poljski}  
\usepackage[polish]{babel}
```

The above code merely allows to use polish letters and translates the automatic text to polish, so that "chapter" becomes "rozdział". There are a few additional things one must remember about.

Connectives

Polish has many single letter connectives: "a", "o", "w", "i", etc., grammar and typography rules don't allow for them to end a printed line. To ensure that LaTeX won't set them as last letter in the line, you have to use non breakable space:

Noc była sierpniowa, ciepła i~słodka, Księżyc oświecał srebrnym światłem w~głębienie, tak, że twarze małego rycerza i~Basi były skapane w blasku. Poniżej, na podwórzu zamkowem, widać było uśpione kupy żołnierzy, a~także i~ciała zabitych podczas dziennej strzelaniny, bo nie znaleziono dotąd czasu na ich pogrzebanie.

Numerals

According to polish grammar rules, you have to put dots after numerals in chapter, section, subsection, etc. headers.

This is achieved by redefining few LaTeX macros.

For books:

```
\renewcommand\thechapter{\arabic{chapter}.}
\renewcommand\thesection{\arabic{chapter}.\arabic{section}.}
\renewcommand\thesubsection{\arabic{chapter}.\arabic{section}.\arabic{subsection}.}
\renewcommand\thesubsubsection{\arabic{chapter}.\arabic{section}.\arabic{subsection}.\arabic{subsubsection}.}
\arabic{subsubsection}.}
```

For articles:

```
\renewcommand\thesection{\arabic{section}.}
\renewcommand\thesubsection{\arabic{section}.\arabic{subsection}.}
\renewcommand\thesubsubsection{\arabic{section}.\arabic{subsection}.\arabic{subsubsection}.}
```

mwcls

Alternatively you can use the `mwart` class instead of `article`, `mwbk` instead of `book` and `mwrep` instead of `report` (those classes have much more European typography settings but *do not* require the use of polish babel settings or character encoding).

Simple usage:

```
\documentclass{mwart}
\usepackage[polish]{babel}
\usepackage{polski}
\begin{document}
Pójdź kińże tę chmurność w głąb flaszę.
\end{document}
```

Full documentation for those classes is available at <http://www.ci.pwr.wroc.pl/pmazur/LaTeX/mwclsdoc.pdf>¹⁴ (polish)

Indentation

It may be customary (depending on publisher) to indent the first paragraph in sections and chapters:

```
\usepackage{indentfirst}
```

Hyphenation and typography

It's much more frowned upon to set pages with hyphenation between pages than it is customary in American typesetting.

To adjust penalties for hyphenation spanning pages, use this command:

```
\brokenpenalty=1000
```

To adjust penalties for leaving widows and orphans (clubs in TeX nomenclature) use those commands:

```
\clubpenalty=1000  
\widowpenalty=1000
```

Commas in math

According to Polish typography rules, fractional parts of numbers should be delimited by a comma, not a dot. To make LaTeX not insert additional space in math mode after a comma (unless there is a space after the comma), use the `icomma` package.

```
\usepackage{icomma}
```

Unfortunately, it is partially incompatible with the `dcolumn` package. One needs to either use dots in columns with numerical data in the source file and make `dcolumn` switch them to commas for display *or* define the column as follows:


```
\begin{tabular}{... D{,}{\mathord\mathcomma}{2} ...}
```

The alternative is to use the `numprint` package, but it is much less convenient.

Further information

Refer the *Słownik Ortograficzny*¹⁵ (in Polish) for additional information on polish grammar and typography rules.

Good extract is available at *Zasady Typograficzne Składania Tekstu*¹⁶ (in Polish)

¹⁵ <http://so.pwn.pl/zasady.php>

¹⁶ <http://dtp.msstudio.com.pl/typo.html>

36.4.14. Portuguese

Add the following code to your preamble:

```
\usepackage[portuguese]{babel}  
\usepackage[latin1]{inputenc}  
\usepackage[T1]{fontenc}
```

if you are in Brazil, you can substitute the language for brazilian portuguese by choosing: `brazilian`. The first line is to get everything translated properly, the second is for being able to input text correctly and the third one to get the hyphenation of words with diacritics right. Note that we are using the `latin1` input encoding here, so this will not work on a Mac or on DOS. Just use the appropriate encoding for your system. If you are using Linux, use

```
\usepackage[utf8]{inputenc}
```

36.4.15. Slovak

Basic settings are fine when left the same as Czech, but Slovak needs special signs for d',t',l'. To be able to type them from keyboard use the following settings

```
\usepackage[slovak]{babel}  
\usepackage[IL2]{fontenc}  
\usepackage[utf8]{inputenc}
```

36.4.16. Spanish

To enable Spanish writing, besides installing the appropriate hyphenation patterns, you type:

```
\usepackage[spanish]{babel}
```

The trick is that Spanish has several options and commands to control the layout. The options may be loaded either at the call to Babel, or before, by defining the command `\spanishoptions`. Therefore, the following commands are roughly equivalent:

```
\def\spanishoptions{mexico}  
\usepackage[spanish]{babel}  
  
\usepackage[spanish,mexico]{babel}
```

On average, the former syntax should be preferred, as the latter is a deviation from standard Babel behavior, and thus may break other programs (LyX, latex2rtf2e) interacting with LaTeX.

Two particularly useful options are `es-noquoting`, `es-nolists`: some packages and classes are known to collide with Spanish in the way they handle active characters, and these options disable the internal workings of Spanish to allow you to overcome these common pitfalls. Moreover, these options may

simplify the way LyX customizes some features of the Spanish layout from inside the GUI.

The options `mexico`, `mexico-com` provide support for local custom in Mexico: the former using decimal dot, as customary, and the latter allowing decimal comma, as required by the Mexican Official Norm (NOM) of the Department of Economy for labels in foods and goods. More localizations are in the making.

The other commands modify the spanish layout after loading `babel`. Two particularly useful commands are `\spanishoperators` and `\spanishdeactivate`.

The macro `\spanishoperators{list of operators}` contains a list of spanish mathematical operators, and may be redefined at will. For instance, the command `\def\spanishoperators{sen}` only defines `sen`, overriding all other definitions; the command `\let\spanishoperators\relax` disables them all. This command supports accented or spaced operators: the

`\acute{<letter>}` command puts an accent, and the `\` command adds a small space. For instance, the following operators are defined by default.

```
\acute{i}m \acute{i}m\, sup \acute{i}m\, inf m\acute{a}cute{a}x
\acute{inf} m\acute{i}n sen tg arc\, sen arc\, cos arc\, tg
cotg cosec senh tgh
```

Finally, the macro `\spanishdeactivate{list of characters}` disables some active characters, to keep you out of trouble if they are redefined by other packages. The candidates for deactivation are the set `<>."'`. Please, beware that some option preemptly the availability of some active characters. In particular, you should not combine the `es-noquoting` option with `\spanishdeactivate{<>}`, or the `es-noshorthands` with `\spanishdeactivate{<>."}'`.

Please check the documentation for Babel or `spanish.dtx` for further details.

Spanish with `inputenc` package

However, by using `latin1` or `UTF8`-encoding it is possible to use nearly all characters without special commands, like: á, é, í, ó, ú, ü, ñ, Á, É, Í, Ó, Ú, Ü, and Ñ.

```
\usepackage[latin1]{inputenc}  
\usepackage[utf8]{inputenc}
```

37. Accents

The rules for producing characters with diacritical marks, such as accents, differ somewhat depending whether you are in text mode, math mode, or the tabbing environment.

37.1. Text mode

37.1.1. Direct input

The Unicode character encoding UTF8 includes several special characters and characters with accents. The following code specifies that the encoding of the LaTeX document source file is UTF8. Font encoding is specified as T1, because it supports the encoding of extended character sets in fonts.

```
\usepackage[utf8]{inputenc}  
\usepackage[T1]{fontenc}
```

Of course, the encoding in the text editor needs to be set to utf8, as well. Depending on the used UTF8 characters, one needs to specify a font that actually includes them (and supports the T1 font encoding). E.g. for German related special characters:

```
\usepackage{lmodern}
```

With XeTeX and LuaTeX the inputenc and fontenc package are no longer needed and replaced by the fontspec package. Both engines support UTF-8 directly and allow the use of TTF and OpenType fonts to support Unicode characters. See the XeTeX section¹ in the Fonts chapter of this book for more information.

37.1.2. Escaped codes

In addition to direct UTF8 input, LaTeX supports the composition of special characters.

The following accents may be placed on letters. Although "o" is used in most of the examples, the accents may be placed on any letter. Accents may even be placed above a "missing" letter; for example, `\~{ }` produces a tilde over a blank space.

¹ Chapter 16.1 on page 494

The following commands may be used only in paragraph (default) or LR (left-right) mode.

LaTeX command	Sample	Description
<code>\`{o}</code>	ò	grave accent
<code>\' {o}</code>	ó	acute accent
<code>\^ {o}</code>	ô	circumflex
<code>\" {o}</code>	ö	umlaut or dieresis
<code>\H{o}</code>	ő	long Hungarian umlaut (double acute)
<code>\~ {o}</code>	õ	tilde
<code>\c{c}</code>	ç	cedilla
<code>\k{a}</code>	ą	ogonek
<code>\l</code>	ł	l with stroke
<code>\={o}</code>	ō	macron accent (a bar over the letter)
<code>\b{o}</code>	ō	bar under the letter

LaTeX command	Sample	Description
<code>\cdot{o}</code>	ó	dot over the letter
<code>\d{u}</code>	ù	dot under the letter
<code>\r{a}</code>	å	ring over the letter (for å there is also the special command <code>\{aa}</code>)
<code>\u{o}</code>	ö	breve over the letter
<code>\v{s}</code>	š	caron/hacek ("v") over the letter
<code>\t{oo}</code>	oô	"tie" (inverted u) over the two letters

To place a diacritic on top of an i or a j, its dot has to be removed. The dotless version of these letters is accomplished by typing

`\i`

and

\j

. For example:

- `\^{\i}` should be used for *i*, circumflex, \hat{i}
- `\"{\i}` should be used for *i*, umlaut, \ddot{i}

If a document is to be written completely in a language that requires particular diacritics several times, then using the right configuration allows those characters to be written directly in the document. For example, to achieve easier coding of umlauts, the babel package can be configured as

```
\usepackage[ngerman]{babel}
```

. This provides the short hand "o for "\{o}. This is very useful if one needs to use some text accents in a label, since no backslash will be accepted otherwise.

More information regarding language configuration can be found in the Internationalization² section.

37.2. Math mode

Several of the above and some similar accents can also be produced in math mode. The following commands may be used only in math mode.

LaTeX command	Sample	Description	Text-mode equivalence
<code>\hat{o}</code>	\hat{o}	circumflex	<code>\^</code>
<code>\widehat{oo}</code>	\widehat{oo}	wide version of <code>\hat</code> over several letters	
<code>\check{o}</code>	\check{o}	vee or check	<code>\v</code>
<code>\tilde{o}</code>	\tilde{o}	tilde	<code>\~</code>
<code>\widetilde{ooo}</code>	\widetilde{ooo}	wide version of <code>\tilde</code> over several letters	
<code>\acute{o}</code>	\acute{o}	acute accent	<code>\'</code>

LaTeX command	Sample	Description	Text-mode equivalence
<code>\grave{o}</code>	ò	grave accent	<code>\`</code>
<code>\dot{o}</code>	ô	dot over the letter	<code>\.</code>
<code>\ddot{o}</code>	ö	two dots over the letter (umlaut in text-mode)	<code>\"</code>
<code>\breve{o}</code>	ö	breve	<code>\u</code>
<code>\bar{o}</code>	ō	macron	<code>\=</code>
<code>\vec{o}</code>	→	vector (arrow) over the letter	

When applying accents to letters `i` and `j`, you can use `\imath` and `\jmath` to keep the dots from interfering with the accents:

LaTeX command	Sample	Description	Sample with upper dot
<code>\hat{\imath}</code>	\hat{i}	circumflex on letter <i>i</i> without upper dot	$\hat{\dot{i}}$
<code>\vec{\jmath}</code>	\vec{j}	vector (arrow) on letter <i>j</i> without upper dot	$\vec{\dot{j}}$

37.3. Tabbing environment

Some of the accent marks used in running text have other uses in the tabbing environment. In that case they can be created with the following command:

- `\a'` for an acute accent
- `\a`` for a grave accent
- `\a=` for a macron accent

37.4. Degree symbol for temperature and math

The basic trick is to use the `\circ` command:

```
$90^\circ$
```

If you find it awkward, you can use the `gensymb` package which provides a `\degree` command.

37.5. Related issues

The actual entering of special characters is system dependent. For example under X-Windows umlauts are entered via `compose+caps+` o, when the keyboard layout does not directly includes them.

Whether LaTeX is actually able to typeset certain UTF8 characters depends on the loaded packages. For example trying to typeset the euro sign (€) may yield following error message:

```
Package inputenc Error: Unicode char \u8:€ not set up for use with LaTeX.
```

Using the package `textcomp` sets this and other characters up.

The usage of the T1 font encoding influences the usability of pdf output (generated via `pdflatex`). For example without specifying T1, extracting the umlaut Ä via a PDF viewer actually extracts the two characters "A. Analog to that, a PDF viewer cannot find words with umlauts in a PDF document which was generated via `pdflatex` without the T1 font encoding. With T1 font encoding (and even with composed special characters) the PDF contains the correct text information.

The package `ae` (almost european) is obsolete. It provided some workarounds for hyphenation of words with special characters. These are not necessary any

more with fonts like lmodern. Using the ae package leads to text encoding problems in PDF files generated via pdf`latex` (e.g. text extraction and searching), besides typographic issues.

37.6. External links

- A few other LaTeX accents and symbols³
- NASA GISS: Accents⁴

3 http://spectroscopy.mps.ohio-state.edu/symposium_53/latexinstruct.html

4 <http://www.giss.nasa.gov/tools/latex/latex-ltx-401.html>

38. Tips and Tricks

38.1. Always writing LaTeX in roman

If you insert the `\LaTeX` command in an area with a non-default font, it will be formatted accordingly. If you want to keep LaTeX written in Computer Modern roman shape, you must redefine the function. However, the naive

```
\renewcommand{\LaTeX}{\rm \LaTeX}
```

will output:

TeX capacity exceeded , sorry [grouping levels =255].

So you need to create a temporary variable.

Sadly,

```
\newcommand{\LaTeXtemp}{\LaTeX}  
\renewcommand{\LaTeX}{\{\rm \LaTeXtemp}}
```

does not work as well.

We must use the TeX primitive \let instead.

```
\let\LaTeXtemp\LaTeX  
\renewcommand{\LaTeX}{\{\rm \LaTeXtemp \}}
```

38.2. *id est* and *exempli gratia* (i.e. and e.g.)

If you simply use the forms "i.e." or "e.g.", LaTeX will treat the periods as end of sentence periods (i.e. full stop¹) since they are followed by a space, and add more space before the next "sentence". To prevent LaTeX from adding space after the last period, the correct syntax is either "i.e.\ " or "e.g.\ " .

Depending on style (e.g., *The Chicago Manual of Style*²), a comma can be used afterwards, which is interpreted by LaTeX as part of a sentence, since the period is not followed by any space. In this case, "i.e.," and "e.g.," do not need any special attention.

If the command `\frenchspacing` is used in the preamble, the space between sentences is always consistent.

¹ <http://en.wikipedia.org/wiki/Full%20stop>

² <http://en.wikipedia.org/wiki/The%20Chicago%20Manual%20of%20Style>

38.3. Grouping Figure/Equation Numbering by Section

For long documents the numbering can become cumbersome as the numbers reach into double and triple digits. To reset the counters at the start of each section and prefix the numbers by the section number, include the following in the preamble.

```
\usepackage{amsmath}
\numberwithin{equation}{section}
\numberwithin{figure}{section}
```

The same can be done with similar counter types and document units such as "subsection".

38.4. Generic header

As explained in the previous sections, a LaTeX source can be used to generate both a DVI and a PDF file. For very basic documents the source is the same but, if the documents gets more complicated, it could be necessary to make some changes in the source so that it will work for a format but it will not for the other. For example, all that is related to graphics has to be adapted according to the final format. As discussed in the section about floating objects, even if you should use different pictures according to the final format, you can override this limit putting in the same folder pictures in different formats (e.g., EPS and PNG) with the same name and link them without writing the extension. There is a simple way to solve this problem:

```
\usepackage{i.fpdf}
```

or, if you don't have this package, you can add the following text just after `\documentclass [. . .] { . . . }` :


```

\newif\ifpdf
\ifx\pdfoutput\undefined
\pdffalse
\else
\ifnum\pdfoutput=1
\pdftrue
\else
\pdffalse
\fi
\fi

```

this is plain TeX code. The *ifpdf* package and this code, both define a new *if-else* you can use to change your code according to the compiler you are using. After you have used this code, you can use whenever you want in your document the following syntax:

```

\ifpdf
% we are running pdflatex
\else
% we are running latex
\fi

```

place after `\input` the code you want to insert if you are compiling with *pdf_latex*, place after `\else` the code you want to insert if you are compiling with *latex*. For example, you can use this syntax to load different packages or different graphic file formats³ according to the compiler.

38.5. Graphics and Graph editors

38.5.1. Vector image editors with LaTeX support

It is often preferable to use the same font and font size in your images as in the document. Moreover, for scientific images, you may need mathematical formulae or special characters (such as Greek letters). Both things can be achieved easily if the image editor allows you to use LaTeX code in your image. Most vector image editors do not offer this option. There are, however, a few exceptions.

In early days, LaTeX users used Xfig⁴ for their drawings. The editor is still used by quite a few people nowadays because it has special 'export to LaTeX' features. It also gives you some very basic ways of encapsulating LaTeX text and math in the image (setting the text's 'special flag' to 'special' instead of 'normal'). When exporting, all LaTeX text will be put in a .tex-file, separately from the rest of the image (which is put in a .ps file).

A newer and easier-to-use vector image editor specially tailored to LaTeX use is IPE⁵. It allows any LaTeX command, including but not limited to mathematical formulae in the image. The program saves its files as editable .eps or .pdf files, which eliminates the need of exporting your image each time you have edited it.

A very versatile vector image editor is Inkscape⁶. It does not support LaTeX text by itself, but you can use the plugin `Textext`⁷ for that. This allows you to

4 <http://en.wikipedia.org/wiki/Xfig>

5 http://en.wikipedia.org/wiki/Ipe_%28program%29

6 <http://en.wikipedia.org/wiki/Inkscape>

7 <http://pav.iki.fi/software/textext/>

put any block of LaTeX code in your image. Additionally since version 0.48 you can export to vectorgraphics with texts separated in a .tex file. Using this way text is rendered by the latex compiler itself.

LaTeXDraw is a free and open source graphical PSTricks generator and editor. It allows you to draw basic geometric objects and save the result in a variety of formats including .jpg, .png, .eps, .bmp as well as .tex. In the last case the saved file contains PSTricks/LaTeX code only. Owing to that you can include any possible LaTeX code in the picture, since the file is rendered by your LaTeX environment directly.

Another way to generate vectorgraphics is using the Asymptote⁸ language. It is a programming language which produces vector images in encapsulated postscript format and supports LaTeX syntax in any textlabels.

8 http://en.wikipedia.org/wiki/Asymptote_%28vector_graphics_language%29

38.5.2. Graphs with gnuplot

A simple method to include graphs and charts in LaTeX documents is to create it within a common spreadsheet software (OpenOffice Calc or MS Office Excel etc.) and include it in the document as a cropped screenshot. However, this produces poor quality rasterized images. Calc also allows you to copy-paste the charts into OpenOffice Draw and save them as PDF files.

Using Microsoft Excel 2010, charts can be copied directly to Microsoft Expression Design 4, where they can be saved as PDF files. These PDF files can be included in LaTeX. This method produces high quality vectorized images.

An excellent method to render graphs is through **gnuplot**⁹, a free and versatile plotting software, that has a special output filter directly for exporting files to LaTeX. We assume, that the data is in a CSV file (comma separated text) in the first and third column. A simple gnuplot script to plot the data can look like this:

9 <http://en.wikipedia.org/wiki/gnuplot>

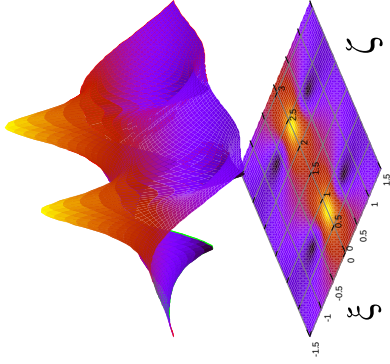


Figure 115 gnuplot can plot various numerical data, functions, error distribution as well as 3D graphs and surfaces

```

set format "$%g$"
set title "Graph 3: Dependence of $V_p$ on $R_0$"
set xlabel "Resistance $R_0$ [$\Omega$]"

```

```
set ylabel "Voltage $V_p$ [V]"
set border 3
set xtics nomirror
set ytics nomirror
set terminal epslatex
set output "graph1.eps"
plot "graph1.csv" using 1:3 #Plot the data
```

Now gnuplot produces two files: the graph drawing in `graph.eps` and the text in `graph.tex`. The second includes the EPS image, so that we only need to include the file `graph.tex` in our document:

```
\input{graph1.tex}
```

The above steps can be automated by the package `gnuplottex`. By placing `gnuplot` commands inside `\begin{gnuplot}\end{gnuplot}`, and compiling with `latex -shell-escape`, the graphs are created and added into your document.

When you are using gnuplottex it is also possible to directly pass the terminal settings as an argument to the environment

```
\begin{gnuplot}[terminal=epslatex, terminaloptions=color,  
scale=0.9, linewidth=2 ]  
...  
\end{gnuplot}
```

Using gnuplottex can cause fraudulent text-highlighting in some editors when using algebraic functions on imported data, such as:

```
(2*( $1 ) ):2
```

Some editors will think of all following text as part of a formula and highlight it as such (because of the '\$' that is interpreted as part of the latex code). This can be avoided by ending with:


```
#$  
\end{gnuplot}
```

As it uncomments the dollar sign for the gnuplot interpreter, but is not affecting the interpretation of the .tex by the editor.

When using pdfLaTeX instead of simple LaTeX, we must convert the EPS image to PDF and to substitute the name in the graph1.tex file. If we are working with a Unix-like shell, it is simply done using:

```
eps2pdf graph1.eps  
sed -i s"/.eps"/".pdf"/g graph1.tex
```

With the included tex file we can work as with an ordinary image.

Instead of calling eps2pdf directly, we can also include the epstopdf package that automates the process. If we include a graphics now and leave out the file extension, epstopdf will automatically transform the .eps-file to

PDF and insert it in the text.

```
\includegraphics{graph1}
```

This way, if we choose to output to PS or DVI, the EPS version is used and if we output to PDF directly, the converted PDF graphics is used. Please note that usage of `epstopdf` requires compiling with `latex -shell-escape`.

Note: Emacs AucTeX users might want to check out `Gnuplot-mode`¹⁰.

38.5.3. Generate png screenshots

This section describes how to generate a png screenshot of a LaTeX page using the `*nix` tool `dvi2png` and the LaTeX package `preview`. This screenshots are useful, for example, if you want to include a LaTeX generated formula

¹⁰ <http://cars9.uchicago.edu/~ravel/software/gnuplot-mode.html>.

on a presentation using your favorite slideware like Powerpoint, Keynote or OpenOffice Impress. First, start by making sure you have the two required tools `dvipng` and `preview`. To check for the LaTeX package type

```
locate preview.sty
```

on a console and if `locate` returns a directory, then you are good. Next, type

```
which dvipng
```

to see if you have `dvipng` installed on your machine and ready to use. Again, if you get a directory, you are ready to start doing screenshot of LaTeX pages. Say you want to take a screenshot of

$$\pi = \sqrt{12} \sum_{k=0}^{\infty} \frac{(-3)^{-k}}{2k+1}.$$

Write this formula on a TeX file name `foo.tex` in this way:

```

\documentclass{article}
\usepackage[active]{preview}
\begin{document}
\begin{preview}
\l
\pi = \sqrt{12} \sum^{\infty}_{k=0} \frac{(-3)^{-k}}{2k+1}
\r
\end{preview}
\end{document}

```

Run LaTeX as usual to generate the dvi file `foo.dvi`. Note the active option in the package declaration and the preview environment around the equation's code. Without any of these two, you won't get an dvi output. Now, we want an X font size formula, where X is measure in pixels. You need to convert this, to dots per inch (dpi). The formula is: $\langle \text{dpi} \rangle = \langle \text{font_px} \rangle * 72.27 / 10$. If you want, for instance, X = 32, then the size in dpi corresponds to 231.26. This value will be passed to `dvipng` using the flag `-D`. To generate the desired png file run the command as follows:

```
dvipng -T tight -D 231.26 -o foo.png foo.dvi
```

The flag `-T` sets the size of the image. The option `tight` will only include all ink put on the page. The option `-o` sends the output to the file name `foo.png`.

38.6. Spell-checking and Word Counting

If you want to spell-check your document, you can use command-line `aspell` `hunspell` (preferably) or `ispell` programs.

```
ispell yourfile.tex
aspell -c yourfile.tex
hunspell -l -t -i utf-8 yourfile.tex
```

Both understand LaTeX and will skip LaTeX commands. You can also use a LaTeX editor with built-in spell checking, such as LyX¹¹, Kile¹², or Emacs¹³. Last another option is to convert LaTeX source to plain text¹⁴ and open resulting file in a word processor like OpenOffice.org or KOffice.

If you want to count words you can, again, use LyX or convert your LaTeX source to plain text and use, for example, UNIX `wc` command:

```
detex yourfile | wc
```

An alternative to the `detex` command is the `pdftotext` command which extracts an ASCII text file from PDF:

-
- 11 <http://en.wikipedia.org/wiki/LyX>
 - 12 <http://en.wikipedia.org/wiki/Kile>
 - 13 <http://en.wikipedia.org/wiki/Emacs>
 - 14 Chapter 35.6 on page 951

1. pdf`latex` `yourfile.tex`
2. pdf`totext` `yourfile.pdf`
3. `wc` `yourfile.txt`

38.7. New even page

In the twoside-mode you have the ability to get a new odd-side page by:

```
\cleardoublepage
```

However, LaTeX doesn't give you the ability to get a new even-side page. The following method opens up this;

The following must be put in your document preamble:

```
\usepackage{ifthen}
```

```

\newcommand{\newevenside}{
  \ifthenelse{\isodd{\thepage}}{\\newpage}{
    \newpage
  }
  \phantom{placeholder} % doesn't appear on page
  \thispagestyle{empty} % if want no header/footer
  \newpage
}
}

```

To activate the new even-side page, type the following where you want the new even-side:

```
\newevenside
```

If the given page is an odd-side page, the next new page is subsequently an even-side page, and LaTeX will do nothing more than a regular `\newpage`. However, if the given page is an even page, LaTeX will make a new (odd) page, put in a placeholder, and make another new (even) page. A crude but effective method.

38.8. Sidebar with information

If you want to put a sidebar with information like copyright and author, you might want to use the `eso-pic` package. Example:

```
\usepackage{eso-pic}
...
\AddToShipoutPicture{%
  \AtPageLowerLeft{%
    \rotatebox{90}{%
      \begin{minipage}{\paperheight}
        \centering\textcopyright~\today{} Humble me
      \end{minipage} %
    } %
  } %
}
```

If you want it on one page only, use the starred version of the `AddToShipoutPicture` command at the page you want it. (`\AddToShipoutPicture*{...}`)

38.9. Hide auxiliary files

If you're using pdflatex you can create a folder in which all the output files will be stored, so your top directory looks cleaner.

```
pdflatex -output-directory tmp
```

Please note that the folder tmp should exist. However if you're using linux you can do something like this:

```
alias pdflatex='mkdir -p tmp; pdflatex -output-directory tmp'
```

Or for vim modify your .vimrc:

```
" use pdflatex
let g:Tex_DefaultTargetFormat='pdf'
let g:Tex_MultipleCompileFormats='pdf,dvi'
let g:Tex_CompilationRule_pdf = 'mkdir -p tmp; pdflatex -output-directory
```

```
tmp
-interaction=nonstopmode $*; cp tmp/*.pdf .'
```

Part VI.

Appendix

39. Authors

39.1. Included books

The following books have been included in this wikibook (or we are working on it!), with permission of the author:

- Andy Roberts' Getting to grips with LaTeX¹. (**Done!**)

¹ <http://www.andy-roberts.net/misc/latex/index.html>

- Not So Short Introduction to LaTeX2e² by Tobias Oetiker, Hubert Partl and Irene Hyna. We have contacted the authors by email asking for permission: they allowed us to use their material, but they never edited directly this wikibook. That book is released under the GPL, that is not compatible with the GFDL used here in Wikibooks. Anyway, we have the permission of the authors to use their work. You can freely copy text from that guide to here. If you find text on both the original book and here on Wikibooks, then that text is double licensed under GPL and GFDL. For more information about Tobias Oetiker and Hubert Partl, their websites are <http://it.oetiker.ch/>³ and <http://homepage.boku.ac.at/partl/>⁴ respectively. **(Done!)**
 - Peter Flynn's Formatting information, a beginner's guide to typesetting with LaTeX⁵. We have contacted him by email asking for permission to use his
-
- 2 <http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf>
 - 3 <http://it.oetiker.ch/>
 - 4 <http://homepage.boku.ac.at/partl/>
 - 5 <http://www.ctan.org/tex-archive/info/beginlatex/beginlatex-3.6.pdf>

work. The original book is released under the *GNU Free Documentation License*, the same as Wikibooks. For more information, his personal website is <http://silmaril.ie/cgi-bin/blog>⁶ (**Working...**)

- LaTeX Primer⁷ from the Indian TeX Users Group. Their document is released under the *GNU Free Documentation License*, the same as Wikibooks, so we can include parts of their document as we wish. In any case, we have contacted Indian TeX Users Group and they allowed us to do it. (**Working...**)
- David Wilkins' Getting started with LaTeX⁸. The book is not released under any free license, but we have contacted the author asking him for the permission to use parts of his book on Wikibooks. He agreed: his work is still protected but you are allowed to copy the parts you want on this Wikibook. If you see text on both the original work and here, then that part (and only that part) is released under the terms of GFDL, like any other text here on Wikibooks. (**Working...**)

6 <http://silmaril.ie/cgi-bin/blog>

7 <http://sarovar.org/projects/ltxprimer/>

8 <http://www.maths.tcd.ie/~dwilkins/LaTeXPrimer/>

39.2. Wiki users

Major contributors to the book on Wikibooks are:

- Alessio Damato⁹
- Jtwdog¹⁰

⁹ <http://en.wikibooks.org/wiki/User%3AAlejo2083>

¹⁰ <http://en.wikibooks.org/wiki/User%3AJtwdog>

40. Links

w:TeX¹ w:LaTeX²

Here are some other online resources available:

-
- 1 <http://en.wikipedia.org/wiki/TeX>
 - 2 <http://en.wikipedia.org/wiki/LaTeX>

40.0.1. Community

- The TeX Users Group³ Includes links to free versions of (La)TeX for many kinds of computer.
- UK-TUG⁴ The UK TeX Users' Group
- TUGIndia⁵ The Indian TeX Users Group
- [news:comp.text.tex comp.text.tex] Newsgroup for (La)TeX related questions
- CTAN⁶ hundreds of add-on packages and programs

3 <http://www.tug.org/>

4 <http://uk.tug.org/>

5 <http://www.tug.org.in/>

6 <http://www.ctan.org/>

40.0.2. Tutorials/FAQs

- Tobias Oetiker's Not So Short Introduction to LaTeX2e:
<http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf>⁷ also at
<http://people.ee.ethz.ch/~oetiker/lshort/lshort.pdf>
- Vel's introduction to LaTeX: What is it, why should you use it, who should use it and how to get started:
http://www.vel.co.nz/vel.co.nz/Blog/Entries/2009/11/4_LaTeX_-_Document_Preparation_System.html
- Peter Flynn's beginner's guide (formatting):
<http://www.ctan.org/tex-archive/info/beginlatex/beginlatex-3.6.pdf>
- The AMS Short Math Guide for LaTeX, a concise summary of math formula typesetting features
<http://www.ams.org/tex/amslatex.html>⁸

7 <http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf>

8 <http://www.ams.org/tex/amslatex.html>

- `amsmath` users guide (PDF) and related files:
<http://www.ctan.org/tex-archive/macros/latex/required/amslatex/math/>
- LaTeX Primer from the Indian TeX Users Group:
<http://sarovar.org/projects/ltxprimer/>
- LaTeX Primer
<http://www.maths.tcd.ie/~dwilkins/LaTeXPrimer/>
- PSTricks--fancy graphics exploiting PDF capabilities
<http://sarovar.org/projects/pstricks/>
- PDFScreen--create LaTeX PDF files that have navigation buttons used for presentations:
<http://sarovar.org/projects/pdfscreen/>
- David Bausum's list of TeX primitives (these are the fundamental commands used in TeX):
<http://www.tug.org/utilities/plain/cseq.html>
- Leslie Lamport's manual for the commands that are unique to LaTeX (commands not used in plain TeX):
<http://www.tex.uniyar.ac.ru/doc/latex2e.pdf>

- The UK TeX FAQ List of questions and answers that are frequently posted at `comp.text.tex`
<http://www.tex.ac.uk/faq>
- TeX on Mac OS X: Guide to using TeX and LaTeX on a Mac
<http://www.ma.nl/tex.html>
- Text Processing using LaTeX
<http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/>
- The (La)TeX encyclopaedia
<http://tex.loria.fr/index.html>
- Hypertext Help with LaTeX
<http://www.giss.nasa.gov/latex/>
- EpsLatex: a very comprehensive guide to images, figures and graphics
<http://www.ctan.org/tex-archive/info/epslatex.pdf>⁹
- The Comprehensive LaTeX Symbol List (in PDF)
<http://www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-a4.pdf>

- Getting to Grips with LaTeX (HTML) Collection of LaTeX tutorials taking you from the very basics towards more advanced topics <http://www.andy-roberts.net/misc/latex/index.html>
- Chapter 8 (about typesetting mathematics) of the *LaTeX companion* <http://www.macrotex.net/textbooks/latexcomp-ch8.pdf>

40.0.3. Reference

- LaTeX Project Site¹⁰
- The Comprehensive TeX Archive Network¹¹ Latest (La)TeX-related packages and software
- TeX Directory Structure¹², used by many (La)TeX distributions

¹⁰ <http://www.latex-project.org/>

¹¹ <http://www.ctan.org>

¹² <http://www.tug.org/tds/>

- Natural Math¹³ converts natural language math formulas to LaTeX representation
- Obsolete packages and commands¹⁴
- Lamport's book *LaTeX: A Document Preparation System*

40.0.4. Templates

- A resource for free high quality LaTeX templates for a variety of applications¹⁵
- LaTeX template for writing PhD thesis¹⁶, 2007
- UCL computer department thesis template¹⁷

¹³ <http://www.math.missouri.edu/~stephen/naturalmath/>

¹⁴ <http://www.ctan.org/tex-archive/info/l2tabu/english/l2tabuen.pdf>

¹⁵ <http://www.LaTeXTemplates.com>

¹⁶ http://openwetware.org/wiki/LaTeX_template_for_PhD_thesis

¹⁷ <http://www.cs.ucl.ac.uk/research/students/Latexforthesis.htm>

- UT thesis template¹⁸, 2006

¹⁸ <http://www.cs.utexas.edu/users/jbednar/latex/>

41. Installation

Installing "LaTeX" is not a simple one-click download and install. Multiple programs often need to be downloaded and installed in order to have a suitable computer system that can be used to create publishable output, such as PDFs. The basic requirement is to have TeX and LaTeX. Optional, and recommended installations include an attractive editor to write LaTeX source documents (this is probably where you will spend most of your time), and a bibliographic management program to manage references.

41.1. TeX and LaTeX

TeX and LaTeX are available for most computer platforms, since they were programmed to be very portable. They are most commonly installed using a distribution, such as TeTeX, MiKTeX, or MacTeX. This, however, does not include any editor or advanced graphical user interface. Other programs, that are not part of the distribution, are used to write and prepare TeX and LaTeX files.

41.1.1. UNIX/Linux

UNIX and Linux users have a wide choice of distributions; the most common are **teTeX** and **TeX Live**. As of May 2006 teTeX is no longer actively maintained and its former maintainer Thomas Esser recommended TeX Live as the

replacement.¹ Ubuntu² and Debian³ users can install one (not both!) of these systems using the system's apt⁴ package manager.

41.1.2. Mac OS X

Mac OS X users may use MacTeX⁵, supporting TeX, LaTeX, AMSTeX, ConTeXt, XeTeX and many other core packages.

TeX Live is also available through MacPorts⁶ and Fink⁷.

-
- 1 [teTeX Home Page {http://www.tug.org/tetex/}](http://www.tug.org/tetex/) (Retrieved January 31, 2007)
 - 2 <http://en.wikipedia.org/wiki/Ubuntu%20%28Linux%20distribution%29>
 - 3 <http://en.wikipedia.org/wiki/Debian>
 - 4 <http://en.wikipedia.org/wiki/Advanced%20Packaging%20Tool>
 - 5 <http://tug.org/mactex/>
 - 6 <http://www.macports.org/>
 - 7 <http://www.finkproject.org/>

41.1.3. Microsoft Windows

Microsoft Windows users can install MiKTeX⁸ onto their computer. This distribution has advanced features, such as automatic installation of packages, and simple interfaces to modify settings, such as default paper sizes.

41.1.4. Online

To get started without needing to install anything, users can use ScribTeX⁹ which is a completely online LaTeX compiler, based on the TeX Live distribution.

8 <http://miktex.org/>

9 <http://www.scribtex.com/>

41.2. Editors

TeX and LaTeX source documents (as well as related files) are all text files, and can be opened and modified in almost any text editor. A few recommended editors include:

Cross-platform

- **TeXStudio**¹⁰
- **Texmaker**¹¹
- **Vim**¹² with **Vim-LaTeX**¹³ **Vimscript** plugin
- **Emacs**¹⁴ with **AUCTeX**¹⁵

10 <http://texstudio.sourceforge.net/>

11 <http://www.xmlmath.net/texmaker/>

12 <http://www.vim.org/>

13 <http://vim-latex.sourceforge.net/>

14 <http://www.gnu.org/software/emacs/>

15 <http://www.gnu.org/software/auctex/>

- ScribTeX¹⁶ (browser based)
- Verbosus¹⁷ (browser based)
- TeXworks¹⁸
- Gummi¹⁹

Linux-only

- Kile²⁰

Mac OS X-only

- TeXShop²¹

Windows-only

-
- 16 <http://www.scribtex.com/>
 - 17 <http://www.verbosus.com/>
 - 18 <http://www.tug.org/texworks/>
 - 19 <http://gummi.midnightcoding.org/>
 - 20 <http://kile.sourceforge.net/>
 - 21 <http://www.uoregon.edu/~koch/texshop/>

- LEd²²
- TeXnicCenter²³
- WinEdt²⁴
- WinShell²⁵

41.3. Bibliography management

Bibliography files (`*.bib`) are most easily edited and modified using a management system. These graphical user interfaces all feature a database form, where information is entered for each reference item, and the resulting text file can be used directly by BibTeX.

Cross-platform

22 <http://www.latexeditor.org/>

23 <http://www.texniccenter.org/>

24 <http://www.winedt.com/>

25 <http://www.winshell.de/>

- JabRef²⁶
- Mendeley²⁷

Mac OS X-only

- BibDesk²⁸

²⁶ <http://jabref.sourceforge.net/>

²⁷ <http://www.mendeley.com//>

²⁸ <http://bibdesk.sourceforge.net/>

41.4. Graphics tools

41.4.1. Xfig

Xfig²⁹ is a basic program that can produce vector graphics, which can be exported to PSTEX. It can be installed on UNIX/Linux platforms. With Ubuntu or Debian distributions, it can be easily installed using apt.

On Microsoft Windows systems, Xfig can only be installed using Cygwin-X³⁰; however, this will require a fast internet connection and about 2 gigabytes of space on your computer. With Cygwin, to run Xfig, you need to first start the "Start X - Server", then launch "xterm" to bring up a terminal. In this terminal type "xfig" (without the quotation marks) and press return.

29 <http://en.wikipedia.org/wiki/Xfig>

30 <http://www.cygwin.com/>

41.4.2. Inkscape

Inkscape³¹ is an open source vector graphics editor, which can export images to EPS³² or PDF³³ files, which may then be imported into LaTeX (see ../Importing Graphics³⁴). It can run natively under Windows, Linux or Mac OS.

An extremely useful plug-in is `texttext`³⁵, which can import LaTeX objects. This can be used for inserting mathematical notation or LaTeX fonts into graphics (which may then be imported into LaTeX documents).

31 <http://en.wikipedia.org/wiki/Inkscape>

32 <http://en.wikipedia.org/wiki/Encapsulated%20PostScript>

33 <http://en.wikipedia.org/wiki/Portable%20Document%20Format>

34 **Chapter 11 on page 343**

35 <http://www.elisanet.fi/ptvirtan/software/texttext/index.html>

41.4.3. OpenOffice.org

It is also possible to export vector graphics to EPS format using OpenOffice.org³⁶ Draw, which is an open source office suite available for Windows, Linux and Mac.

41.5. See also

- Installing Extra Packages³⁷

41.6. References

36 <http://en.wikipedia.org/wiki/OpenOffice.org>

37 <http://en.wikibooks.org/wiki/LaTeX%2FPackages%2FInstalling%20Extra%20Packages>

42. Useful Measurement Macros

A list of macros and their values

42.1. Units

First, we introduce the LaTeX measurement units. All LaTeX units are two-letter abbreviations. You can choose from a variety of units. Here are the most common ones.¹

¹ <http://www.uz.ac.zw/science/mathslatex/latex-86.html>

Abbreviation	Definition	Value in points (pt)
pt	a point is $1/72.27$ inch, that means about 0.0138 inch or 0.3515 mm.	1
mm	a millimeter	2.84
cm	a centimeter	28.4
in	inch	72.27
ex	roughly the height of an 'x' in the current font	<i>undefined, depends on the font used</i>
em	roughly the width of an 'M' (uppercase) in the current font	<i>undefined, depends on the font used</i>

And here are some less common units.²

² <http://www.uz.ac.zw/science/mathslatex/ltx-86.html>

Abbreviation	Definition	Value in points (pt)
bp	a big point is 1/72 inch, that means about 0.0139 inch or 0.3527 mm.	1.00375
pc	pica	12
dd	didôt (1157 didôt = 1238 points)	1.07
cc	cîcero (12 didôt)	12.84
sp	scaled point (65536sp per point)	0.000015

42.2. Length 'macros'

Some length commands are;

\baselineskip

The normal vertical distance between lines in a paragraph

`\baselinestretch`

Multiplies `\baselineskip`

`\columnsep`

The distance between columns

`\columnwidth`

The width of the column

`\evensidemargin`

The margin for 'even' pages (think of a printed booklet)

`\linewidth`

The width of a line in the local environment

`\oddsidemargin`

The margin for 'odd' pages (think of a printed booklet)

`\paperwidth`

The width of the page

`\paperheight`

The height of the page

`\parindent`

The normal paragraph indentation

`\parskip`

The extra vertical space between paragraphs

`\tabcolsep`

The default separation between columns in a tabular environment

`\textheight`

The height of text on the page

`\textwidth`

The width of the text on the page

`\topmargin`

The size of the top margin

`\unitlength`

Units of length in Picture Environment

42.3. Length manipulation macros

You can change the values of the variables defining the page layout with two commands. With this one you can set a new value:

```
\setlength{parameter}{length}
```

with this other one, you can add a value to the existing one:

```
\addtolength{parameter}{length}
```

You can create your own length with the command:

```
\newlength{parameter}
```

You may also set a length from the size of a text with one of these commands:

```
\settowidth{parameter}{some text}  
\settoheight{parameter}{some text}  
\settodepth{parameter}{some text}
```

When using these commands, you may duplicate the text that you want to use as reference if you plan to also display it. But LaTeX also provides a set of commands to avoid this duplication:

```
\newsavebox{boxname}  
\savebox{boxname}{some text}  
\usebox{boxname}
```

You may wish to look at the example below to see how you can use these. The command `\newsavebox` creates a placeholder for storing a text; the command `\savebox` stores the specified text in this placeholder, and does not display anything in the document; and `\usebox` recalls the content of the placeholder into the document.

See `LaTeX/Page_Layout`³ for samples using these.

42.4. Samples

Resize an image to take exactly half the text width :

```
\includegraphics[width=0.5\textwidth]{mygraphic}
```

Make distance between items larger (inside an itemize environment) :

```
\addtolength{\itemsep}{0.5\baselineskip}
```

Use of *savebox* to resize an image to the height of the text:

```
% Create the holders we will need for our work
\newlength{\mytitleheight}
\newsavebox{\mytitletext}
% Create the reference text for measures
\savebox{\mytitletext}{%
  \Large\bfseries This is our title%
}
\settoheight{\mytitleheight}{\usebox{\mytitletext}}
% Now creates the actual object in our document
\framebox[\textwidth][l]{%
  \includegraphics[height=\mytitleheight]{my_image}%
  \hspace{2mm}%
  \usebox{\mytitletext}%
}
```

42.5. References

42.6. See also

- University of Cambridge > Engineering Department > computing help > LaTeX > Squeezing Space in LaTeX⁴

⁴ <http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/squeeze.html>

43. Text Formatting

This section will guide you through the various text, paragraph, and page formatting techniques. *Formatting* tends to refer to most things to do with appearance, so it makes the list of possible topics quite eclectic: text style, font, size, etc.

A lot of the formatting techniques are required to differentiate certain elements from the rest of the text. It is often necessary to add emphasis to key words or phrases. Footnotes are useful for providing extra information or clarification without interrupting the main flow of text. So, for these reasons, formatting is very important. However, it is also very easy to abuse, and a document that has been over-done can look and read worse than one with none at all.

LaTeX is so flexible that we will actually only skim the surface, as you can have much more control over the presentation of your document if you wish. Having said that, one of the purposes of LaTeX is to take away the stress of having to deal with the physical presentation yourself, so you need not get too carried away!

43.1. Spacing

43.1.1. Line Spacing

If you want to use larger inter-line spacing in a document, you can change its value by putting the

```
\linespread{factor}
```

command into the preamble of your document. Use `\linespread{1.3}` for "one and a half" line spacing, and `\linespread{1.6}` for "double"

line spacing. Normally the lines are not spread, so the default line spread factor is 1.

The `setspace` package allows more fine-grained control over line spacing. To set "one and a half" line spacing document-wide, but not where it is usually unnecessary (e.g. footnotes, captions):

```
\usepackage{setspace}  
%\singlespacing  
\onehalfspacing  
%\doublespacing  
%\setstretch{1.1}
```

To change line spacing within the document, the `setspace` package provides the environments `singlespace`, `onehalfspace`, `doublespace` and `spacing`:

This paragraph has \\ default \\ line spacing.

```
\begin{doublespace}
```

```
This paragraph has \\ double \\ line spacing.  
\end{doublespace}  
  
\begin{spacing}{2.5}  
This paragraph has \\ huge gaps \\ between lines.  
\end{spacing}
```

43.1.2. Horizontal Space

LaTeX determines the spaces between words and sentences automatically. To add horizontal space, use:

```
\hspace{length}
```

If such a space should be kept even if it falls at the end or the start of a line, use `\hspace*` instead of `\hspace`. The length in the simplest case is just a

number plus a unit, e.g. `\hspace{1.5 cm}`. For a list of the possible units, see the Useful Measurement Macros¹ appendix.

The command:

```
\stretch{n}
```

generates a special rubber space. It stretches until all the remaining space on a line is filled up. If two `\hspace{\stretch{n}}` commands are issued on the same line, they grow according to the stretch factor.

x x x

```
x\hspace{\stretch{1}}
x\hspace{\stretch{3}} x
```

43.1.1.3. Vertical Space

The space between paragraphs, sections, subsections, etc. is determined automatically by LaTeX. If you want to customize the default paragraph spacing, it can be achieved with the following command in the preamble of your document:

```
\parskip 7.2pt
```

If necessary, additional vertical space *between two paragraphs* can be added with the command:

```
\vspace{length}
```

This command should normally be used between two empty lines. If the space should be preserved at the top or at the bottom of a page, use the starred version of the command, `\vspace*`, instead of `\vspace`. The `\stretch`

command, in connection with `\pagebreak`, can be used to typeset text on the last line of a page, or to center text vertically on a page.

Additional space between two lines of the same paragraph or within a table is specified with the

```
\\[length]
```

command.

If you want to add space at the beginning of the document, without anything else written before, then you may use

```
{ \vspace*{length} }
```

It's important you use the `\vspace*` command instead of `\vspace`, otherwise LaTeX can silently ignore the extra space.

43.1.4. Fill the rest of the line

Several macros allow filling the rest of the line `--` or stretching parts of the line `--` in different manners:

- `\hfill` will produce spaces.
- `\dotfill` will produce dots.
- `\hrulefill` will produce a rule.

43.2. Hyphenation

LaTeX hyphenates words whenever necessary. If the hyphenation algorithm does not find the correct hyphenation points, you can remedy the situation by using the following commands to tell TeX about the exception. The command

```
\hyphenation{word list}
```

causes the words listed in the argument to be hyphenated only at the points marked by “-”. The argument of the command should only contain words

built from normal letters, or rather characters that are considered to be normal letters by LaTeX. It is known that the hyphenation algorithm does not find all correct American English hyphenation points for several words. A log of known exceptions is published periodically in the *TUGboat* journal. (See a 2008 list: <http://www.tug.org/TUGboat/Articles/tb29-2/tb92hyf.pdf>)

The hyphenation hints are stored for the language that is active when the hyphenation command occurs. This means that if you place a hyphenation command into the preamble of your document it will influence the English language hyphenation. If you place the command after the `\begin{document}` and you are using some package for national language support like babel, then the hyphenation hints will be active in the language activated through babel. The example below will allow “hyphenation” to be hyphenated as well as “Hyphenation”, and it prevents “FORTRAN”, “Fortran” and “fortran” from being hyphenated at all. No special characters or symbols are allowed in the argument. Example:

```
\hyphenation{FORTRAN Hy-phen-a-tion}
```

The command `\-` inserts a discretionary hyphen into a word. This also becomes the only point where hyphenation is allowed in this word. This

command is especially useful for words containing special characters (e.g., accented characters), because LaTeX does not automatically hyphenate words containing special characters.

```
\begin{minipage}{2in}  
I think this is: su\per\cal\-%  
i\frag\i\lis\tic\ex\pi\-%  
al\i\do\cious  
\end{minipage}
```

**I think this is: supercalifragi-
listicexpialidocious**

This can be quite cumbersome if one has many words that contain a dash like electromagnetic-endoscopy. One alternative to this is using the `\hyp` command of the `hyphenat` package. This command typesets a hyphen and

allows full automatic hyphenation of the other words forming the compound word. One would thus write

```
electromagnetic\hyp{}endioscopy
```

Several words can be kept together on one line with the command

```
\mbox{text}
```

It causes its argument to be kept together under all circumstances. Example:

```
My phone number will change soon. It will be \mbox{01116 291 2319}.
```

`\fbox` is similar to `\mbox`, but in addition there will be a visible box drawn around the content.

To avoid hyphenation altogether, the penalty for hyphenation can be set to an extreme value:

```
\hyphenpenalty=100000
```

You can change the degree to which LaTeX will hyphenate by changing the value of `\tolerance=1000` and `\hyphenalty=1000`. You'll have to experiment with the values to achieve the desired effect. A document which has a low tolerance value will cause LaTeX not to tolerate uneven spacing between words, hyphenating words more frequently than in documents with higher tolerances. Also note that using a higher text width will decrease the probability of encountering badly hyphenated word. For example adding

```
\usepackage{geometry}
```

will widen the text width and reduce the amount of margin overruns.

43.3. Quote-marks

LaTeX treats left and right quotes as different entities. For single quotes, `'` (on American keyboards, this symbol is found on the tilde key (adjacent to the number 1 key on most)) gives a left quote mark, and `'` is the right. For double quotes, simply double the symbols, and LaTeX will interpret them

accordingly. (Although, you can use the " for right double quotes if you wish). On British keyboards, ' ' ' is left of the ' 1 ' key and shares the key with ' ¬ ' , and sometimes ' ! ' or ' | ' . The apostrophe (' ' ') key is to the right of the colon/semicolon key and shares it with the ' @ ' symbol.

To 'quote' in Latex

To 'quote' in Latex.

Figure 116

To ``quote'' in Latex

To "quote" in Latex.

Figure 117

To ``quote" in Latex

To "quote" in Latex.

Figure 118

To „quote” in Latex.

To ,,quote'' in Latex

Figure 119

Figure 119

```
``Please press the `x' key.''
```

„Proszę, naciśnij klawisz «x»”.

```
,,Proszę, naciśnij klawisz <<x>>''.
```

Figure 121

The right quote is also used for apostrophe in LaTeX without trouble.

For left bottom quote and European quoting style you need to use T1 font encoding enabled by:

```
\usepackage[T1]{fontenc}
```

The package `csquotes` offers a multi-lingual solution to quotations, with integration to citation mechanisms offered by `BibTeX`. This package allows one for example to switch languages and quotation styles according to babel language selections.

43.4. Diacritics and accents

Diacritics may be added to letters by placing special escaped metacharacters before the letter that requires the diacritic. For a list of diacritic metacharacters, see `LaTeX/Accents2`.

43.5. Space between Words and Sentences

To get a straight right margin in the output, LaTeX inserts varying amounts of space between the words. By default, It also inserts slightly more space at the end of a sentence. However, the extra space added at the end of sentences is generally considered typographically old-fashioned in English language printing. (The practice is found in nineteenth century design and in twentieth century typewriter styles.) Most modern typesetters treat the end of sentence space the same as the interword space. (See for example, Bringhurst's *Elements of Typographic Style*.) The additional space after periods can be disabled with the command

```
\frenchspacing
```

which tells LaTeX not to insert more space after a period than after ordinary character. Frenchspacing can be turned off later in your document via the

```
\nonfrenchspacing
```

command.

If an author wishes to use the wider end-of-sentence spacing, care must be exercised so that punctuation marks are not misinterpreted as ends of sentences. TeX assumes that sentences end with periods, question marks or exclamation marks. Although if a period follows an uppercase letter, this is not taken as a sentence ending, since periods after uppercase letters normally occur in abbreviations. Any exception from these assumptions has to be specified by the author. A backslash in front of a space generates a space that will not be enlarged. A tilde ‘

~

’ character generates a space that cannot be enlarged and additionally prohibits a line break. The command

`\@`

in front of a period specifies that this period terminates a sentence even when it follows an uppercase letter. (If you are using

`\frenchspacing`

, then none of these exceptions need be specified.)

43.5.1. Margin misalignment and interword spacing

Some very long words, numbers or URLs may not be hyphenated properly and move far beyond the side margin. One solution for this problem is to use `sloppypar` environment, which tells LaTeX to adjust word spacing less strictly. As a result, some spaces between words may be a bit too large, but long words will be placed properly.

This is a paragraph with a very long word ABCDEFGHIJKLMNOPRST; then we have an another bad thing — a long number 1234567890123456789.
This is a paragraph with a very long word ABCDEFGHIJKLMNOPRST; then we have an another bad thing — a long number 1234567890123456789.

```
This is a paragraph with
a very long word ABCDEFGHIJKLMNOPRST
then we have another bad thing
--- a long number 1234567890123456789.

\begin{sloppypar}
This is a paragraph with
a very long word ABCDEFGHIJKLMNOPRST;
then we have an another bad thing
--- a long number 1234567890123456789.
\end{sloppypar}
```

Figure 122 border

43.6. Ligatures

Some letter combinations are typeset not just by setting the different letters one after the other, but by actually using special symbols (like "ff"), called ligatures³. Ligatures can be prohibited by inserting

```
{ }
```

or, if this does not work,

```
{\kern0pt }
```

between the two letters in question. This might be necessary with words built from two words. Here is an example:

³ <http://en.wikipedia.org/wiki/Typographyical%20ligature>

Not shelfful but shelfful

```
\Large Not shelfful\\  
but shelf{}ful
```

Figure 123

Some tools are unable to perform search in documents that contain ligatures (a search for "**finally**" wouldn't find the string "`finally`"). If one desires, for greater accessibility, to disable ligatures altogether in the whole document, the

```
\DisableLigatures
```

from the `microtype` package⁴ can be used:

4 <http://www.ctan.org/tex-archive/macros/latex/contrib/microtype/>

```
\usepackage{microtype}
\DisableLigatures{encoding = *, family = *}
```

Note that this will also disable ligatures such as `--` \rightarrow `-`, `---` \rightarrow `---`, etc.

If you are using XeLaTeX and OpenType fonts, the fontspec package allows for standard ligatures to be turned off as well as fancy swash ligatures to be turned on.

Another solution is to use the cmap package, which will help the reader to interpret the ligatures:

```
\usepackage[resetfonts]{cmap}
```

43.7. Slash marks

The normal typesetting of the

/

character in LaTeX does not allow following characters to be "broken" on to new lines, which often create "overflow" errors in output (where letters push off the margin). Words that use slash marks, such as "input/output" should be typeset as "

`input\slash output`

", which allow the line to "break" after the slash mark (if needed). The use of the

/

character in LaTeX should be restricted to units, such as "

`mm/year`

", which should not be broken over multiple lines.

43.8. Emphasizing Text

In order to add some emphasis to a word or phrase, the simplest way is to use the

```
\emph{text}
```

command.

I want to *emphasize* a word.

I want to `\emph{emphasize}` a word. **Figure 124**

43.9. Fonts

See also: `LaTeX/Fonts`⁵.

In LaTeX, there are many ways to specify and control font, and this section is only intended to serve as a brief overview of the topic.

43.9.1. Font Styles

There are three main font families: roman (e.g., Times), sans serif (e.g., Arial) and monospace (e.g., Courier). You can also specify styles such as italic and bold. To change the default family for the rest of the document, change the

```
\familydefault
```

command, like in the following example:


```
\renewcommand{\familydefault}{\sfdefault}
```

The following table lists the commands you will need to access the typical font styles:

LaTeX command	Equivalent to	Output style	Remarks
<code>\textnormal{...}</code>	<code>{\normalfont ...}</code>	document font family	this is the default or normal font
<code>\emph{...}</code>	<code>{\em ...}</code>	<i>emphasis</i>	typically italics
<code>\textrm{...}</code>	<code>{\rmfamily ...}</code>	roman font family	
<code>\textsf{...}</code>	<code>{\sffamily ...}</code>	sans serif font family	

LaTeX command	Equivalent to	Output style	Remarks
<code>\texttt{...}</code>	<code>{\ttfamily ...}</code>	teletypefont family	this is a fixed-width or monospace font
<code>\textup{...}</code>	<code>{\upshape ...}</code>	upright shape	the same as the normal typeface
<code>\textit{...}</code>	<code>{\itshape ...}</code>	italic shape	
<code>\textsl{...}</code>	<code>{\slshape ...}</code>	slanted shape	a skewed version of the normal typeface (similar to, but slightly different from, italics)

LaTeX command	Equivalent to	Output style	Remarks
<code>\textsc{...}</code>	<code>{\scshape ...}</code>	Small Capitals	
<code>\uppercase{...}</code>	<code>{...}</code>	uppercase (all caps)	Also <code>\lowercase</code> . There are some caveats, though; see here ⁶ .
<code>\textbf{...}</code>	<code>{\bfseries ...}</code>	bold	
<code>\textmd{...}</code>	<code>{\mdseries ...}</code>	medium weight	a font weight in between normal and bold

6 <http://www.tex.ac.uk/cgi-bin/texfaq2html?label=caschange>

The commands in column two are not entirely equivalent to the commands in column one: They do not correct spacing after the selected font style has ended. The commands in column one are therefore in general recommended.

You may have noticed the absence of underline. This is because underlining is not recommended for typographic reasons (it weighs the text down). You should use *emph* instead. Although this is available via the

```
\underline{...}
```

command, text underlined in this way will not break properly. This functionality has to be added with the `ulem` (underline emphasis) package. Stick

```
\usepackage{ulem}
```

in your preamble. By default, this overrides the

```
\emph
```

command with the underline rather than the italic style. It is unlikely that you wish this to be the desired effect, so it is better to stop `ulem` taking over

```
\emph
```

and simply call the underline command as and when it is needed.

- To restore the usual em formatting, add

```
\normalem
```

straight after the document environment begins. Alternatively, use

```
\usepackage[normalem]{ulem}
```

- To underline, use

```
\uline{...}
```

- To add a wavy underline, use

```
\uwave{...}
```

- And for a strike-out

```
\sout{...}
```

.

Some font styles are not compatible one with the other. But some extra packages will fill this hole. For bold small capitals, you might want to use:

```
\usepackage{bold-extra}
...
\textsc{\textbf{This is bold small capitals}}
```

43.9.2. Sizing text

To apply different font sizes, simply follow the commands on this table:

Command	Output
<code>\tiny</code>	<small>sample text</small>
<code>\scriptsize</code>	sample text
<code>\footnotesize</code>	sample text
<code>\small</code>	sample text
<code>\normalsize</code>	sample text
<code>\large</code>	sample text
<code>\Large</code>	sample text
<code>\LARGE</code>	sample text
<code>\huge</code>	sample text
<code>\Huge</code>	sample text

Figure 125

Note that the font size definitions are set by the document class. Depending on the document style the actual font size may differ from that listed above. And not every document class has unique sizes for all 10 size commands.

Absolute Point Sizes, [10pt] being default size	standard classes, <i>proc</i>			AMS classes, <i>mem-oir</i>			<i>slides</i>			<i>beamer</i>		
	[10pt]	[11pt]	[12pt]	[10pt]	[11pt]	[12pt]	[10pt]	[11pt]	[12pt]	[10pt]	[11pt]	[12pt]
<code>\tiny</code>	6.80565	7.33325	7.33325	7.33325	7.97224	8.50012	17.27505	5.31258	6.37509	6.37509	6.37509	6.37509
<code>\scriptsize</code>	7.97224	8.50012	8.50012	7.97224	8.50012	9.24994	20.73755	7.43760	8.50012	8.50012	8.50012	8.50012
<code>\footnotesize</code>	8.50012	9.24994	10.00002	8.50012	9.24994	10.00002	20.73755	8.50012	9.24994	9.24994	9.24994	9.24994
<code>\small</code>	9.24994	10.00002	10.95003	9.24994	10.00002	10.95003	20.73755	9.24994	10.00002	10.00002	10.00002	10.00002
<code>\normalsize</code>	10.95003	11.74988	11.74988	10.00002	10.95003	11.74988	24.88382	10.00002	10.95003	10.95003	10.95003	10.95003
<code>\large</code>	11.74988	11.74988	14.09984	10.95003	11.74988	14.09984	29.86258	11.74988	11.74988	11.74988	11.74988	11.74988
<code>\Large</code>	14.09984	14.09984	15.84985	11.74988	14.09984	15.84985	35.82510	14.09984	14.09984	14.09984	14.09984	14.09984
<code>\LARGE</code>	15.84985	15.84985	19.02350	14.09984	15.84985	19.02350	43.00012	16.24988	16.24988	16.24988	16.24988	16.24988
<code>\huge</code>	19.02350	19.02350	22.82086	15.84985	19.02350	22.82086	51.60014	19.50362	19.50362	19.50362	19.50362	19.50362
<code>\Huge</code>	22.82086	22.82086	22.82086	19.02350	22.82086	22.82086	51.60014	23.39682	23.39682	23.39682	23.39682	23.39682

As a technical note, points in TeX follow the standard American point size in which 1 pt is approximately 0.3513_6 mm. The standard point size used in most modern computer programs (known as the *desktop publishing point* or *PostScript point*) has 1 pt equal to approximately 0.352_7 mm while the

standard European point size (known as the *Didot point*) had 1 pt equal to approximately 0.37597151 mm. (See: `w:Point_(typography)`⁷.)

The default for `\normalsize` is 10-point, but it may differ for some Document Styles or their options. The actual size produced by these commands also depends on the Document Style and, in some styles, more than one of these size commands may produce the same actual size.

The new size takes effect immediately after the size command; if an entire paragraph or unit is set in a certain size, the size command should include the blank line or the `\end{}` which delimites the unit. (See *Declarations*.)

These commands change the size within a given scope, so for instance `{\Large some words}` will change the size of only some words, and does not affect the font in the rest of the document. These commands cannot be used in math mode. However, part of a formula may be set in a different size by using an `\mbox` command containing the size command.

⁷ http://en.wikipedia.org/wiki/Point_%28typography%29

Another use for size commands is to embed a single command inside the size-syntax, which affects all things within this command, e.g. `\Large{\tableofcontents}`.

43.9.3. Formatting Macros

Even if you can easily change the output of your fonts using those commands, you're better off not using explicit commands like this, because they work in opposition to the basic idea of LaTeX, which is to separate the logical and visual markup of your document. This means that if you use the same font changing command in several places in order to typeset a special kind of information, you should use

```
\newcommand
```

to define a "logical wrapper command" for the font changing command.

Do not **enter** this room, it's occupied by **machines** of unknown origin and purpose.

```
\newcommand{\oops}[1]{\textbf{#1}}
```

```
Do not \oops{enter} this room,  
it's occupied by \oops{machines}  
of unknown origin and purpose.
```

This approach has the advantage that you can decide at some later stage that you want to use some visual representation of danger other than

```
\textbf
```

, without having to wade through your document, identifying all the occurrences of

```
\textbf
```

and then figuring out for each one whether it was used for pointing out danger or for some other reason.

43.9.4. Text mode superscript and subscript

To superscript text in text-mode, you can use the

```
\textsuperscript{}
```

command. This allows you to, for instance, typeset 6th as 6th:

```
Michelangelo was born on March 6\textsuperscript{th}, 1475.
```

A very common use of subscripts within the text environment is to typeset chemical formulae. For this purposes, a highly recommended package is `mhchem`⁸. This package is easy to use, and works with your text fonts (rather than math fonts). To insert a chemical formula, use

```
\ce{}
```

with the text-equivalent formula, for example:

8 <http://www.ctan.org/tex-archive/macros/latex/contrib/mhchem/>

Ammonium sulphate is $(\text{NH}_4)_2\text{SO}_4$.

Figure 126

```
% In your preamble, add:  
\usepackage[version=3]{mhchem}  
...  
% In your document:  
Ammonium sulphate is \ce{(NH4)2SO4}.
```

Subscripting in text-mode is not supported by LaTeX alone; however, several packages allow the use of the

```
\textsubscript{}
```

command. For instance, `bpchem`⁹, `KOMA-Script2`¹⁰, and `fixltx2e`¹¹ all support this command. Of these, `fixltx2e`¹² is perhaps the most universal option since it is distributed with LaTeX and requires no additional packages to be implemented.

9 <http://www.ctan.org/tex-archive/macros/latex/contrib/bpchem/>

10 <http://www.ctan.org/tex-archive/macros/latex/contrib/koma-script/>

11 <http://tug.ctan.org/pkg/fixltx2e>

12 <http://tug.ctan.org/pkg/fixltx2e>

It is found that $\text{height}_{\text{apple tree}}$ is different

```
% In your preamble, add:  
\usepackage{fixltx2e}  
...
```

```
% In your document:  
It is found that  $\text{height}\text{\textsubscript{apple tree}}$  is  
different than  $\text{height}\text{\textsubscript{orange tree}}$ .
```

If you do not load a package that supports

```
\textsubscript{}
```

, the math mode must be used. This is easily accomplished in running text by bracketing your text with the

§

symbol. In math mode subscripting is done using the underscore:

$_{} \{ \}$

For example, the formula for water is written as:

H_2O is the formula for water

```
H\$_2\$O is the formula for water
```

See also the above mentioned package `mhchem` for chemical symbols and formulas.

Note that in math mode text will appear in a font suitable for mathematical variables. In math mode, to generate roman text, for example, one would use the

```
\mathrm
```

command:

This is normal roman and_{subscript} roman

```
This is  $\mathrm{normal}$  and  $\mathrm{subscript}$  roman
```

Note the use of `\<space>` to insert a space in math mode.

Similarly, you can superscript using:

This is normal roman and^{superscript} roman

```
This is  $\mathrm{normal}$  and  $\mathrm{superscript}$  roman
```

43.9.5. Text figures ("old style" numerals)

Many typographers prefer to use titling figures, sometimes called lining figures, when numerals are interspersed with full caps, when they appear in tables, and when they appear in equations, using text figures¹³ elsewhere. LaTeX allows this usage through the

```
\oldstylenums{}
```

command:

```
\oldstylenums{1234567890}
```

Some fonts do not have text figures built in; the `textcomp` package attempts to remedy this by effectively generating text figures from the currently-selected font. Put

```
\usepackage{textcomp}
```

in your preamble. `textcomp` also allows you to use decimal points, properly formatted dollar signs, etc. within

¹³ <http://en.wikipedia.org/wiki/Text%20figures>

```
\oldstylenums{}
```

.

One common use for text figures is in section, paragraph, and page numbers. These can be set to use text figures by placing some code in your preamble:

```
\usepackage{textcomp}

% Enclose everything in an \AtBeginDocument{}
\AtBeginDocument{%
  % Make \section{} use text figures
  \let\myTheSection\thesection
  \renewcommand{\thesection}{\oldstylenums{\myTheSection}}

  % Make \paragraph{} use text figures
  \let\myTheParagraph\theparagraph
  \renewcommand{\theparagraph}{\oldstylenums{\myTheParagraph}}

  % Make the page numbers in text figures
  \let\myThePage\thepage
  \renewcommand{\thepage}{\oldstylenums{\myThePage}}
}
```

Should you use additional sectioning or paragraphing commands, you may adapt the previous code listing to include them as well.

NOTE: A subsequent use of the

```
\pagenumbering  
command, e.g.,  
\pagenumbering{arabic}
```

, will reset the

```
\thepage
```

command back to the original. Thus, if you use the

```
\pagenumbering
```

command in your document, be sure to reinstate your

```
\myThePage definition
```

from the code above:

...

```
\tableofcontents
\pagenumbering{roman}
\chapter{Preface}
...
\chapter{Introduction}
...
\pagenumbering{arabic}
% without this, the \thepage command will not be in oldstyle (e.g.,
in your Table of Contents}
\renewcommand{\thepage}{\oldstylenums{\myThePage}}
\Chapter{Foo}
...
```

43.10. Symbols and special characters

43.10.1. Dashes and Hyphens

LaTeX knows four kinds of dashes: a hyphen¹⁴ (-), en dash¹⁵ (–), em dash¹⁶ (—), or a minus sign¹⁷ (−). You can access three of them with different numbers of consecutive dashes. The fourth sign is actually not a dash at all—it is the mathematical minus sign:

14 <http://en.wikipedia.org/wiki/hyphen>

15 <http://en.wikipedia.org/wiki/Dash%23En%20dash>

16 <http://en.wikipedia.org/wiki/Dash%23Em%20dash>

17 <http://en.wikipedia.org/wiki/Plus%20and%20minus%20signs%23Minus%20sign>

daughter-in-law, X-rated

pages 13–67

yes—or no?

0, 1 and –1

Hyphen: daughter-in-law, X-rated\\

En dash: pages 13--67\\

Em dash: yes---or no? \\

Minus sign: \$0\$, \$1\$ and \$-1\$

Figure 127

The names for these dashes are: ‘-’(-) hyphen, ‘--’(–) en-dash, ‘---’(—) em-dash and ‘—’(—) minus sign. They have different purposes:

Input	Output	Purpose
-	-	inter-word
--	-	page range, 1–10
---	—	punctuation dash — like this

Figure 128

Use

`\hyp{ }`

macro from `hyphenat` package instead of hyphen if you want LaTeX to break compound words between lines.

43.10.2. Euro € currency symbol

When writing about money these days, you need the euro sign¹⁸. You have several choices. If the fonts you are using have a euro symbol and you want to use that one, first you have to load the *textcomp* package in the preamble:

```
\usepackage{textcomp}
```

then you can insert the euro symbol with the command

```
\texteuro
```

. If you want to use the official version of the euro symbol, then you have to use `eurosym`, load it with the

```
official
```

option in the preamble:

```
\usepackage[official]{eurosym}
```

¹⁸ <http://en.wikipedia.org/wiki/euro%20sign>

then you can insert it with the

```
\euro
```

command. Finally, if you want a euro symbol that matches with the current font style (e.g., bold, italics, etc.) but your current font does not provide it, you can use the `eurosym` package again but with a different option:

```
\usepackage[gen]{eurosym}
```

again you can insert the euro symbol with

```
\euro
```

43.10.3. Ellipsis (...)

A sequence of three dots is known as an *ellipsis*¹⁹, which is commonly used to indicate omitted text. On a typewriter, a comma or a period takes the same amount of space as any other letter. In book printing, these characters occupy

19 <http://en.wikipedia.org/wiki/Ellipsis>

only a little space and are set very close to the preceding letter. Therefore, you cannot enter ‘ellipsis’ by just typing three dots, as the spacing would be wrong. Instead, there is a special command for these dots. It is called

```
\ldots
```

```
:
```

Not like this ... but like this:
New York, Tokyo, Budapest, ...

```
Not like this ... but like this:\\  
New York, Tokyo, Budapest, \ldots
```

Figure 129

Alternatively, you can use the

```
\textellipsis
```

command which allows the spacing between the dots to vary.

43.10.4. Ready-made strings

There are some very simple LaTeX commands for typesetting special text strings:

Command	Example	Description
<code>\today</code>	May 31, 2006	Current date
<code>\TeX</code>	\TeX	Your favorite typesetter
<code>\LaTeX</code>	\LaTeX	The Name of the Game
<code>\LaTeXe</code>	$\LaTeX_{2\epsilon}$	The current incarnation

Figure 130

43.10.5. Other symbols

LaTeX has *lots* of symbols at its disposal. The majority of them are within the mathematical domain, and later chapters will cover how to get access to them. For the more common text symbols, use the following commands:

Command	Symbol	Command	Symbol
<code>\%</code>	%	<code>\#</code>	#
<code>\\$</code>	\$	<code>\&</code>	&
<code>\{</code>	{	<code>\}</code>	}
<code>_</code>	-	<code>\\$</code>	§
<code>\P</code>	¶	<code>\dagger</code>	†
<code>\ddagger</code>	‡	<code>\backslash</code>	\
<code>\textbar</code>		<code>\less</code>	<
<code>\textgreater</code>	>	<code>\endash</code>	—
<code>\textendash</code>	—	<code>\textregistered</code>	®
<code>\texttrademark</code>	™	<code>\textquestiondown</code>	¿
<code>\textexclamdown</code>	!	<code>\textcircled{a}</code>	Ⓐ
<code>a</code>	^a	<code>\copyright</code>	©
<code>\pounds</code>	£		

Figure 131

Not mentioned in above table, tilde (~) is used in LaTeX code to produce non-breakable space²⁰. To get printed tilde sign, either make it verbatim text²¹ or write

```
\~{ }
```

. And a visible space `_` can be created with

```
\textvisiblespace
```

.

Of course, these are rather boring. For some more interesting symbols, the Postscript ZipfDingbats font is available thanks to the `pifont` package. Hopefully you are beginning to notice now that when you want to use a package, you need to add the declaration to your preamble; in this instance:

```
\usepackage{pifont}
```

. **Next, the command**

```
\ding{number}
```

20 Chapter 43.5 on page 1084

21 Chapter 10.4.1 on page 333

, will print the specified symbol. Here is a table of the available symbols:

32	→	33	↗	34	↘	35	↖	36	↙	37	↔	38	↞	39	↠
40	↡	41	↢	42	↣	43	↤	44	↥	45	↦	46	↧	47	↨
48	↩	49	↪	50	↫	51	↬	52	↭	53	↮	54	↯	55	↰
56	↱	57	↲	58	↳	59	↴	60	↵	61	↶	62	↷	63	↸
64	↹	65	↺	66	↻	67	↼	68	↽	69	↾	70	↿	71	⇀
72	★	73	☆	74	☉	75	☆	76	★	77	★	78	★	79	★
80	✱	81	✱	82	✱	83	✱	84	✱	85	✱	86	✱	87	✱
88	✱	89	✱	90	✱	91	✱	92	✱	93	✱	94	✱	95	✱
96	✱	97	✱	98	✱	99	✱	100	✱	101	✱	102	✱	103	✱
104	✱	105	✱	106	✱	107	✱	108	●	109	○	110	■	111	□
112	□	113	□	114	□	115	▲	116	▼	117	◆	118	◇	119	♠
120	♠	121	♠	122	♠	123	♠	124	♠	125	♠	126	♠		
168	♣	169	♣	170	♣	171	♣	172	♣	173	♣	174	♣	175	♣
176	♣	177	♣	178	♣	179	♣	180	♣	181	♣	182	♣	183	♣
184	♣	185	♣	186	♣	187	♣	188	♣	189	♣	190	♣	191	♣
192	♣	193	♣	194	♣	195	♣	196	♣	197	♣	198	♣	199	♣
200	♣	201	♣	202	♣	203	♣	204	♣	205	♣	206	♣	207	♣
208	♣	209	♣	210	♣	211	♣	212	→	213	→	214	↔	215	↑
216	↗	217	↘	218	↙	219	↖	220	↗	221	↘	222	↙	223	↖
224	↗	225	↘	226	↙	227	↖	228	↗	229	↘	230	↙	231	↖
232	↗	233	↘	234	↙	235	↖	236	↗	237	↘	238	↙	239	↖
241	↗	242	↘	243	↙	244	↖	245	↗	246	↘	247	↙	248	↖
249	↗	250	↘	251	↙	252	↖	253	↗	254	↘				

Figure 132 ZapfDingbats symbols

43.11. Notes and References

This page uses material from Andy Roberts' Getting to grips with LaTeX with permission from the author.

44. Sample LaTeX documents

The easiest way to learn how to use latex is to look at how other people use it. Here is a list of *real world* latex sources that are freely available on the internet. The information here is sorted by application area, so that it is grouped by the scientific communities that use similar notation and LaTeX constructs.

44.1. General examples

Tutorial examples, books, and real world uses of LaTeX.

- `caption.tex`¹, `simple.tex`², `wrapped.tex`³
- `[ftp://tug.ctan.org/tex-archive/macros/latex/base/small2e.tex small2e.tex]`
and `[ftp://tug.ctan.org/tex-archive/macros/latex/base/sample2e.tex sample2e.tex]`
- A short example of how to use LaTeX for scientific reports⁴ by Stephen J. Eglen.
- The not so Short Introduction to LaTeX⁵ by Tobias Oetiker is distributed with full latex sources.

1 <http://en.wikibooks.org/wiki/LaTeX%2Fcaption.tex>

2 <http://en.wikibooks.org/wiki/LaTeX%2Fsimple.tex>

3 <http://en.wikibooks.org/wiki/LaTeX%2Fwrapped.tex>

4 <http://www.tug.org/pracjourn/2006-2/eglen/>

5 <http://www.ctan.org/tex-archive/info/lshort/english/>

44.2. Semantics of Programming Languages

Articles on programming language research, from syntax to semantics, including source code listings, type rules, proof trees, and even some category theory. A good place to start is Mitchell Wand's *Latex Resources*⁶, including a sample file that also demonstrates Didier Remy's `mathpartir`⁷ package. The following are latex sources of some articles, books, or presentations from this field:

- **Pugs: Bootstrapping Perl 6 with Haskell**⁸. This paper by Audrey Tang contains nice examples on configuring the `listings` package⁹ to format source code.

6 <http://www.ccs.neu.edu/course/csg264/latex/>

7 <http://cristal.inria.fr/~remy/latex/>

8 <http://svn.openfoundry.org/pugs/docs/talks/hw2005.tex>

9 <http://en.wikibooks.org/wiki/LaTeX%2FPackages%2FListings>

45. Command Glossary

This is a glossary of LaTeX commands—an alphabetical listing of LaTeX commands with the summaries of their effects. (Brackets "[]" are optional arguments and braces "{ }" are required arguments.)

45.1.

/

see slash marks¹

`\@`

following period ends sentence

`\[*][extra-space]`

new line. See Page Layout².

`\,`

thin space, math and text mode

`\;`

thick space, math mode

`\:`

1 <http://en.wikibooks.org/wiki/LaTeX%2FFormatting%23Slash%20marks>

medium space, math mode

`\!`

negative thin space, math mode

`\-`

hyphenation; tabbing

`\=`

set tab, see tabbing

`\>`

tab, see tabbing

`\<`

back tab, see tabbing

`\+`

see tabbing

`\'`

accent or tabbing

`\'`

accent or tabbing

`\l`

double vertical lines, math mode

`\l`

start math environment³

`\l`

end math environment

\M

begin displaymath environment

\M

end displaymath environment

45.2. A

\addcontentsline{file}{sec_unit}{entry}

adds an entry to the specified list or table

\addtocontents{file}{text}

adds text (or formatting commands) directly to the file that generates the specified list or table

\addtocounter{counter}{value}

increments the counter

\address{Return address}

\addtolength{len-cmd}{len}

increments a length command, see Useful Measurement Macros⁴

\addvspace

adds a vertical space of a specified height

\alph

causes the current value of a specified counter to be printed in alphabetic characters

\appendix

changes the way sectional units are numbered so that information after the command is considered part of the appendix

\arabic

causes the current value of a specified counter to be printed in Arabic numbers

\author

declares the author(s). See Document Structure⁵

45.3. B

\backslashslash

prints a backslash

\baselineskip

a length command (see Useful Measurement Macros⁶), which specifies the minimum space between the bottom of two successive lines in a paragraph

`\baselinestretch`

scales the value of `\baselineskip`

`\bf`

Boldface typeface

`\bibitem`

generates a labeled entry for the bibliography⁷

`\bigskipamount`

`\bigskip`

equivalent to `\vspace{\bigskipamount}`

6 Chapter 42 on page 1059

7 Chapter 3.1.8 on page 98

`\boldmath`

bold font in math mode

45.4. C

`\cal`

Calligraphic style in math mode

`\caption`

generate caption for figures and tables

`\cdots`

Centered dots

`\centering`

Used to center align LaTeX environments

\chapter

Starts a new chapter. See Document Structure⁸.

\circle

\cite

Used to make citations⁹ from the provided bibliography

\cleardoublepage

\clearpage

Ends the current page and causes any floats to be printed. See Page Layout¹⁰.

\cline

8 Chapter 3.1.4 on page 85

9 Chapter 5.2 on page 121

10 Chapter 9.5 on page 310

Adds horizontal line in a table that spans only to a range of cells. See `\hline`¹¹ and `../Tables/`¹² chapter.

`\closing`

Inserts a closing phrase (e.g. `\closing{yours sincerely}`), leaves space for a handwritten signature and inserts a signature specified by `\signature{}`. Used in the *Letter* class.

`\color`

Specifies color of the text. `../Colors`¹³

`\copyright`

makes © sign. See `Formatting`¹⁴.

11 Chapter 45.9 on page 1146

12 Chapter 7 on page 219

13 Chapter 15 on page 471

14 <http://en.wikibooks.org/wiki/...%2FFormatting%23Other%20symbols>

45.5. D

`\dashbox`

`\date`

`\ddots`

Inserts a diagonal ellipsis (3 diagonal dots) in math mode

`\documentclass[options]{style}`

Used to begin a latex document

`\dotfill`

45.6. E

`\em`

Toggles italics on/off for the text inside curly braces with the command. Such as `\em` This is in italics `\em` but this isn't `\em` and this is again}. This command allows nesting.

`\emph`

Toggles italics on/off for the text in curly braces following the command e.g. `\emph{This is in italics \emph{but this isn't} and this is again}`.

`\ensurermath (LaTeX2e)`

`\euro`

Prints euro € symbol. Requires `eurosym` package.

45.7. F

`\fbox`

\flushbottom

\fnsymbol

\footnote

Creates a footnote¹⁵.

\footnotemark

\footnotesize

Sets font size. See Formatting¹⁶.

\footnotetext

\frac

15 <http://en.wikibooks.org/wiki/LaTeX%2FFormatting%23Footnotes>

16 <http://en.wikibooks.org/wiki/...%2FFormatting%23Font%20Styles%20and%20size>

inserts a fraction in mathematics mode. The usage is `\frac{numerator}{denominator}`.

\frame

\framebox

Like `\makebox` but creates a frame around the box. See `LaTeX/Advanced Topics#Boxes`¹⁷.

\frenchspacing

Instructs LaTeX to abstain from inserting more space after a period (‘.’) than is the case for an ordinary character. In order to untoggle this functionality resort to the command `\nonfrenchspacing`¹⁸.

17 Chapter 33.2 on page 892

18 Chapter 45.14 on page 1155

45.8. G

45.9. H

\hfill

Abbreviation for `\hspace{\hfill}`.

\hline

adds a horizontal line in a tabular environment. See also `\cline19`, `Tables20` chapter.

\href

Add a link, or an anchor. See `Hyperlinks21`

19 Chapter 45.4 on page 1139

20 Chapter 7 on page 219

21 Chapter 14 on page 437

\hrulefill

\hspace

Produces horizontal space.

\huge

Sets font size. See Formatting²².

\Huge

Sets font size. See Formatting²³.

\hyphenation{word list}

22 <http://en.wikibooks.org/wiki/..%2FFormating%23Font%20styles%20and%20size>

23 <http://en.wikibooks.org/wiki/..%2FFormating%23Font%20styles%20and%20size>

Overrides default hyphenation algorithm for specified words. See Hyphenation²⁴

45.10. I

\include

This command is different from `\input` in that it's the output that is added instead of the commands from the other files. For more see [LaTeX/Basics](#)²⁵

\includegraphics

Inserts an image²⁶. Requires `graphicx` package.

\includeonly

²⁴ <http://en.wikibooks.org/wiki/..%2FFormatting%23Hyphenation>

²⁵ Chapter 2.4.2 on page 71

²⁶ Chapter 11.1.9 on page 377

\indent**\input**

Used to read in LaTeX files. For more see LaTeX/Basics²⁷.

\it

Italicizes the text which is inside curly braces with the command. Such as `{\it This is in italics}`. `\em` is generally preferred since this allows nesting.

\item

Creates an item in a list. Used in list structures²⁸.

27 Chapter 2.4.2 on page 71

28 <http://en.wikibooks.org/wiki/LaTeX%2FFormattin%23List%20Structures>

45.11. K

\kill

45.12. L

\label

Used to create label which can be later referenced with `\ref`. See Labels and Cross-referencing²⁹.

\large

Sets font size. See Formatting³⁰.

²⁹ Chapter 19 on page 647

³⁰ <http://en.wikibooks.org/wiki/...%2FFormatting%23Font%20styles%20and%20size>

\Large

Sets font size. See Formatting³¹.

\LARGE

Sets font size. See Formatting³².

\LaTeX

Prints LaTeX logo. See Formatting³³.

\LaTeXe

-
- 31 <http://en.wikibooks.org/wiki/..%2FFormatting%23Font%20styles%20and%20size>
- 32 <http://en.wikibooks.org/wiki/..%2FFormatting%23Font%20styles%20and%20size>
- 33 <http://en.wikibooks.org/wiki/..%2FFormatting%23Ready-made%20strings>

Prints current LaTeX version logo. See Formatting³⁴.

\ldots

Prints sequence of three dots. See Formatting³⁵.

\left

\lefteqn

\line

\linebreak

Suggests LaTeX to break line in this place. See Page Layout³⁶.

\linethickness

34 <http://en.wikibooks.org/wiki/..%2FFormatting%23Ready-made%20strings>

35 <http://en.wikibooks.org/wiki/..%2FFormatting%23Ellipsis%20%28...%29>

36 Chapter 9.5 on page 310

\linewidth

\listoffigures

Inserts a list of the figures in the document. Similar to TOC³⁷

\listoftables

Inserts a list of the tables in the document. Similar to TOC³⁸

\location

45.13. M

\makebox

37 Chapter 3.1.7 on page 93

38 Chapter 3.1.7 on page 93

Defines a box that has a specified width, independent from its content. See LaTeX/Advanced Topics#Boxes³⁹.

`\maketitle`

Causes the title page to be typeset, using information provided by commands such as `\title{}` and `\author{}`.

`\markboth \markright`

`\mathcal`

`\mathop`

`\mbox`

Write a text in roman font inside a math part

`\medskip`

`\multicolumn`

\multiput

45.14. N

\newcommand

Defines a new command. See New Commands⁴⁰.

\newcounter

\newenvironment

Defines a new environment. See New Environments⁴¹.

\newfont

\newlength

40 Chapter 31.1 on page 844

41 Chapter 31.2 on page 849

`\newline`

Ends current line and starts a new one. See Page Layout⁴².

`\newpage`

Ends current page and starts a new one. See Page Layout⁴³.

`\newsavebox`**`\newtheorem`****`\nocite`**

Adds a reference to the bibliography without an inline citation. `\nocite{*}` causes all entries in a bibtex database to be added to the bibliography.

`\noindent`**`\nolinebreak`**

42 Chapter 9.5 on page 310

43 Chapter 9.5 on page 310

\nonfrenchspacing

Setting the command `\nonfrenchspacing` toggles the command `\frenchspacing`⁴⁴ and activates LaTeX standards to insert more space after a period (‘.’) than after an ordinary character.

\normalsize

Sets default font size. See [Formatting](#)⁴⁵.

\nopagebreak

Suggests LaTeX not to break page in this place. See [Page Layout](#)⁴⁶.

\not

44 [Chapter 45.7](#) on page 1143

45 <http://en.wikibooks.org/wiki/...%2FFormatting%23Font%20Style%20and%20size>

46 [Chapter 9.5](#) on page 310

45.15. O

\onecolumn

\opening

Inserts an opening phrase when using the *letter* class, for example `\opening{Dear Sir}`

\oval

\overbrace

Draws a brace over the argument. Can be used in `displaystyle` with superscript to label formulae. See *Advanced Mathematics*⁴⁷.

\overline

Draws a line over the argument.

45.16. P

\pagebreak

Suggests LaTeX breaking page in this place. See Page Layout⁴⁸.

\pagenumbering

Defines the type of characters used for the page numbers. Options : arabic, roman, Roman, alph, Alph, gobble (invisible).

\pageref

Used to reference to number of page where a previously declared \label is located. See Floats, Figures and Captions⁴⁹.

\pagestyle

48 Chapter 9.5 on page 310

49 Chapter 13.2.3 on page 413

See Page Layout⁵⁰.

\par

Starts a new paragraph

\paragraph

Starts a new paragraph. See Document Structure⁵¹.

\parbox

Defines a box whose contents are created in paragraph mode. See Advanced Topics⁵².

\parindent

Normal paragraph indentation. See Useful Measurement Macros⁵³.

50 Chapter 9.3.1 on page 288

51 Chapter 3.1.4 on page 85

52 Chapter 33.2 on page 892

53 Chapter 42 on page 1059

\parskip

\part

Starts a new part of a book. See Document Structure⁵⁴.

\protect

\providecommand (LaTeX2e)

See Customizing LaTeX⁵⁵.

\put

45.17. R

\raggedbottom

54 Chapter 3.1.4 on page 85

55 Chapter 31.1 on page 844

Command used for top justified within other environments.

`\raggedleft`

Command used for right justified within other environments.

`\raggedright`

Command used for left justified within other environments.

`\raisebox`

Creates a box and raises its content. See `LaTeX/Advanced Topics#Boxes`⁵⁶.

`\ref`

Used to reference to number of previously declared `\label`. See `Labels and Cross-referencing`⁵⁷.

`\renewcommand`

56 Chapter 33.2 on page 892

57 Chapter 19 on page 647

\right

\rm

\roman

\rule

Creates a line of specified width and height. See LaTeX/Advanced Topics#Rules and Struts⁵⁸.

45.18. S

\savebox

Makes a box and saves it in a named storage bin.

\sbox

The short form of `\savebox` with no optional arguments.

`\sc`

`\scriptsize`

Sets font size. See Formatting⁵⁹.

`\section`

Starts a new section. See Document Structure⁶⁰.

`\setcounter`

`\setlength`

`\settowidth`

`\sf`

⁵⁹ <http://en.wikibooks.org/wiki/..%2FFormating%23Font%20styles%20and%20size>

\shortstack

\signature

In the *Letter* class, specifies a signature for later insertion by `\closing`.

\sl

\slash

See slash marks⁶¹

\small

Sets font size. See Formatting⁶².

\smallskip

61 <http://en.wikibooks.org/wiki/LaTeX%2FFormatting%23Slash%20marks>

62 <http://en.wikibooks.org/wiki/..%2FFormatting%23Font%20Styles%20and%20size>

\sout

Strikes out text. Requires `ulem` package. See Formatting⁶³.

\space

force ordinary space

\sqrt

Creates a root⁶⁴ (default square, but magnitude can be given as an optional parameter).

\stackrel

Takes two arguments and stacks the first on top of the second.

\subparagraph

63 <http://en.wikibooks.org/wiki/..%2FFormating%23Font%20Styles%20and%20size>

Starts a new subparagraph. See Document Structure⁶⁵.

\subsection

Starts a new subsection. See Document Structure⁶⁶.

\subsubsection

Starts a new sub-subsection. See Document Structure⁶⁷.

45.19. T

\tableofcontents

Inserts a table of contents (based on section headings) at the point where the command appears.

65 Chapter 3.1.4 on page 85

66 Chapter 3.1.4 on page 85

67 Chapter 3.1.4 on page 85

\telephone

In the *letter* class, specifies the sender's telephone number.

\TeX

Prints TeX logo. See Formatting⁶⁸.

\textbf{}

Sets bold font style. See Formatting⁶⁹.

\textcolor{}}}

Creates colored text. See Entering colored text⁷⁰.

\textit{}

68 <http://en.wikibooks.org/wiki/...%2FFormatting%23Ready-made%20strings>

69 <http://en.wikibooks.org/wiki/...%2FFormatting%23Font%20styles%20and%20size>

70 Chapter 15.2 on page 473

Sets italic font style. See Formatting⁷¹.

`\textmd{}`

Sets medium weight of a font. See Formatting⁷².

`\textnormal{}`

Sets normal font. See Formatting⁷³.

`\textrm{}`

Sets roman font family. See Formatting⁷⁴.

-
- 71 `http://en.wikibooks.org/wiki/..%2FFormatting%23Font%20Styles%20and%20size`
- 72 `http://en.wikibooks.org/wiki/..%2FFormatting%23Font%20Styles%20and%20size`
- 73 `http://en.wikibooks.org/wiki/..%2FFormatting%23Font%20Styles%20and%20size`
- 74 `http://en.wikibooks.org/wiki/..%2FFormatting%23Font%20Styles%20and%20size`

`\textsc{}`

Sets font style to small caps. See Formatting⁷⁵.

`\textsf{}`

Sets sans serif font family. See Formatting⁷⁶.

`\textsl{}`

Sets slanted font style. See Formatting⁷⁷.

`\texttt{}`

75 <http://en.wikibooks.org/wiki/...%2FFormatting%23Font%20styles%20and%20size>

76 <http://en.wikibooks.org/wiki/...%2FFormatting%23Font%20styles%20and%20size>

77 <http://en.wikibooks.org/wiki/...%2FFormatting%23Font%20styles%20and%20size>

Sets typewriter font family. See Formatting⁷⁸.

\textup{}

Sets upright shape of a font. See Formatting⁷⁹.

\textwidth

\textheight

\thanks

\thispagestyle

\tiny

78 <http://en.wikibooks.org/wiki/..%2FFormatting%23Font%20styles%20and%20size>

79 <http://en.wikibooks.org/wiki/..%2FFormatting%23Font%20styles%20and%20size>

Sets font size. See Formatting⁸⁰.

\title

\today

Writes current day. See Formatting⁸¹.

\tt

\twocolumn

\typeout

\typein

80 <http://en.wikibooks.org/wiki/..%2FFormatting%23Font%20Styles%20and%20size>

81 <http://en.wikibooks.org/wiki/..%2FFormatting%23Ready-made%20strings>

45.20. U

\uline

Underlines text. Requires `ulem` package. See Formatting⁸².

\underbrace

\underline

\unitlength

\usebox

\usecounter

\uwave

⁸² <http://en.wikibooks.org/wiki/..%2FFormatting%23Font%20Styles%20and%20size>

Creates wavy underline. Requires `ulem` package. See Formatting⁸³.

45.21. V

`\value`

`\vbox{text}`

Encloses a paragraph's text to prevent it from running over a page break

`\vdots`

Creates vertical dots. See Mathematics⁸⁴.

`\vector`

83 <http://en.wikibooks.org/wiki/..%2FFormating%23Font%20Styles%20and%20size>

`\verb`

Creates inline verbatim text. See Formatting⁸⁵.

`\vfill`

`\vline`

`\vphantom`

`\vspace`

This page uses material from Dr. Sheldon Green's Hypertext Help with LaTeX.

⁸⁵ <http://en.wikibooks.org/wiki/LaTeX%2FFormatting%23Verbatim%20Text>

46. More Bibliographies

This is a gentle introduction to using some of the bibliography functionality available to LaTeX users beyond the BibTeX basics. This introduction won't be discussing how to create new styles or packages but rather how to use some existing ones. It is worth noting that *Harvard*, for example, is a *citation* style. It is associated with an alphabetical reference list secondarily ordered on date, but the only strictly defined element of *Harvard* style is the citation in *author-date* format.

46.1. The example data

The database used for my examples contains just the following

```
@article{erdos65,
  title = {Some very hard sums},
  journal={Difficult Maths Today},
  author={Paul Erd\H{o}s and Arend Heyting and Luitzen Egbertus
Brouwer}
  year={1930}
  pages={30}
}
```

46.2. The limits of BibTeX styles

Using `cite.sty` and BibTeX makes it very easy to produce **some** bibliography styles. But *author-date* styles - for example the often mentioned, never defined "Harvard" - are not so easy. It's true that you can download some *.bst* files

from CTAN that will handle some variants but using them is not always straightforward. This guide deals with *Natbib* a supplementary package that can access *.bib* files and has sophisticated functionality for producing custom or default author-year format citations and bibliographies as well as the numerical styles handled by BibTeX.

46.3. Natbib

Natbib is a package created by Patrick Daly as a replacement for the *cite.sty* package when *author-date* citation styles are required. Natbib provides three associated bibliography styles:

- plainnat
- abbrvnat
- unsrtnat

which correspond to the three styles available by default in BibTeX where you have a plain numbered style, an abbreviated numbered style and an unsorted

numbered style. Alongside these new styles is an extended set of citation commands to provide flexible citation formats. These are

```
\citet[]{} 
```

and

```
\citep[]{} 
```

each of which has a number of variants.

46.3.1. The Preamble

All Natbib styles require that you load the package in your document preamble. So, a skeleton LaTeX file with Natbib might look like this:

```
\documentclass[] {article}
\usepackage [round] {natbib}

\begin{document} 
```

Document body text with citations.

```
\bibliographystyle{plainnat}  
\bibliography{myrefs}  
  
\end{document}
```

Options

Options available with Natbib can be specified in the brackets on the `\usepackage` command. Among them are:

Option	Effect
round	()
square	[]
curly	{}
angle	<>
semicolon	separate citations with ;
colon	as semicolon

Option	Effect
comma	separate with commas
authoryear	author-year citations
numbers	numeric citations
super	superscript citations
sort	multiple citations are ordered as in bibliography
sort&compress	as <i>sort</i> but number ranges are compressed and hyphenated
compress	number ranges are compressed and hyphenated but only where the 'natural' sort produces a continuous range
longnamesfirst	first citation is full author list and subsequent citations are abbreviated
sectionbib	allows multiple bibliographies in the same document

Option	Effect
nonamebreak	forces all author names onto one line
merge	merges a citation with a previous citation
elide	elides any repeated elements in merged references
mcite	ignore merge

Clearly some of these options require explanation but that will be achieved via examples below. For now, we just note that they can be passed through `\usepackage[]{} in the preamble of your LaTeX file.`

46.4. Citation

46.4.1. Basic Citation Commands

To cite with Natbib, use the commands `\citet` or `\citep` in your document. The "plain" versions of these commands produced abbreviated lists in the case of multiple authors but both have * variants which result in full author listings. I have assumed the use of the *round* option in these examples.

`\citet` and `\cite*`

The `\citet` command is used for *textual* citations, that is to say that author names appear in the text outside of the parenthetical reference to the date of publication. This command can take options for chapter, page numbers etc. Here are examples

```
\citet{Erdos65}
```

produces

Erdős et al. (1965)

<code>\citet[chapter 2]{Erdos65}</code>	produces	Erdős et al. (1965, chapter 2)
<code>\citet[pp. 10-12]{Erdos65}</code>	produces	Erdős et al. (1965, pp. 10-12)
<code>\citet[see][chapter 2]{Erdos65}</code>	produces	Erdős et al. (see 1965, chap. 2)

Here are the `\citet*` versions

<code>\citet*{Erdos65}</code>	produces	Erdős, Heyting and Brouwer (1965)
<code>\citet*[chapter 2]{Erdos65}</code>	produces	Erdős, Heyting and Brouwer (1965, chapter 2)
<code>\citet*[pp. 10-12]{Erdos65}</code>	produces	Erdős, Heyting and Brouwer (1965, pp. 10-12)

```
\citet*[see][chap.  
2]{Erdos65}
```

produces

Erdős , Heyting and
Brouwer (see 1965,
chap. 2)

\citep and \citep*

The `\citep` command is used where the author name is to appear inside the parentheses alongside the date.

```
\citep{Erdos65}  
\citep[chapter  
2]{Erdos65}  
\citep[pp. 10-  
12]{Erdos65}
```

produces

produces

produces

(Erdős et al. 1965)
(Erdős et al. 1965,
chapter 2)
(Erdős et al. 1965, pp.
10-12)

`\citep[see][chap.
2]{Erdos65}`

produces

(Erdős et al. see 1965,
chap. 2)

Here are the `\citep*` versions

`\citep*{Erdos65}`

produces

(Erdős, Heyting and
Brouwer 1965)

`\citep*[chapter
2]{Erdos65}`

produces

(Erdős, Heyting and
Brouwer 1965, chap-
ter 2)

`\citep*[pp. 10-
12]{Erdos65}`

produces

(Erdős, Heyting and
Brouwer 1965, pp.
10-12)

`\citep*[see][chap.
2]{Erdos65}`

produces

(Erdős, Heyting and
Brouwer see 1965,
chap. 2)

46.4.2. The Reference List

Having dealt with basic varieties of citation, we turn to the creation of the bibliography or reference list. Inserting a correct and correctly formatted bibliography when using Natbib is no different than when using plain BibTeX. There are two essential commands -

```
\bibliography{mybibliographydatabase}
```

which LaTeX interprets as an instruction to read a bibliographic database file (eg myrefs.bib) and insert the relevant data here, and

```
\bibliographystyle{plainnat}
```

which specifies how the data are to be presented. Above the three basic Natbib styles were mentioned as analogues of the partially homonymous styles in BibTeX. Let us imagine documents bearing citations as in the section about citation above¹. Here is, approximately, how these citations would appear in plainnat.

Paul Erdos, Arend Heyting, and Luitzen Egbertus Brouwer. *Some difficult sums*. Kluwer, Dordmund, 1930.

Figure 133

46.4.3. What more is there?

This covers the basic functionality provided by the package `Natbib`. It may not, of course, provide what you are looking for. If you don't find what you want here then you should probably next investigate `harvard.sty` which provides a slightly different set of author-date citation functions. Providing a gentle guide to `harvard.sty` is my next rainy day project.

47. Index

This is an alphabetical index of the book.

47.1. A

- Absolute Beginners¹
- Abstract²

1 <http://en.wikibooks.org/wiki/LaTeX%2FAbsolute%20Beginners>
2 <http://en.wikibooks.org/wiki/LaTeX%2FFormatting>
23Abstracts

- Accents³
- Advanced Topics⁴
- Algorithms⁵
- Arrays⁶
- Authors⁷

47.2. B

- babel⁸
- Basics⁹

3 <http://en.wikibooks.org/wiki/LaTeX%2FFormatting%23Accents>

4 Chapter 33 on page 887

5 Chapter 21 on page 679

6 Chapter 17.10 on page 538

7 Chapter 39 on page 1035

8 Chapter 36 on page 953

9 Chapter 2 on page 35

- **beamer package**¹⁰
- **Bibliography Management**¹¹
- **BibTeX**¹²
- **Bold**¹³
- **Bullets**¹⁴
- **Bullet points**¹⁵

-
- 10 **Chapter 28 on page 779**
 - 11 **Chapter 5 on page 117**
 - 12 **Chapter 5 on page 117**
 - 13 **<http://en.wikibooks.org/wiki/LaTeX%2FFormatting%23Font%20styles%20and%20size>**
 - 14 **Chapter 6 on page 193**
 - 15 **Chapter 6 on page 193**

47.3. C

- Captions¹⁶
- Collaborative Writing of LaTeX Documents¹⁷
- Color¹⁸
- `color` package¹⁹
- Columns, see Multi-column Pages²⁰
- Cross-referencing²¹
- Customizing LaTeX²²

-
- 16 Chapter 13 on page 399
 - 17 Chapter 34 on page 899
 - 18 Chapter 15 on page 471
 - 19 Chapter 15 on page 471
 - 20 Chapter 9.4 on page 307
 - 21 Chapter 19 on page 647
 - 22 Chapter 31 on page 843

47.4. D

- Dashes²³
- description environment²⁴
- Diacritical marks²⁵
- Document Classes²⁶
- Document Structure²⁷
- Drawings²⁸

23 <http://en.wikibooks.org/wiki/LaTeX%2FFormatting%23Dashes%20and%20Hyphens>

24 <http://en.wikibooks.org/wiki/LaTeX%2FFormatting%23Description>

25 <http://en.wikibooks.org/wiki/LaTeX%2FFormatting%23Accents>

26 Chapter 2.2.1 on page 46

27 Chapter 3 on page 77

28 Chapter 12 on page 387

47.5. E

- e.g. (exempli gratia)²⁹
- Ellipsis³⁰
- em-dash³¹
- en-dash³²
- enumerate³³
- Errors and Warnings³⁴

29 Chapter 38.2 on page 1011

30 http://en.wikibooks.org/wiki/LaTeX%2FFormatting%23Ellipsis_.28.E2.80.A6.29

31 http://en.wikibooks.org/wiki/LaTeX%2FFormatting%23Dashes_and_Hyphens

32 http://en.wikibooks.org/wiki/LaTeX%2FFormatting%23Dashes_and_Hyphens

33 <http://en.wikibooks.org/wiki/LaTeX%2FFormatting%23Enumerate>

34 Chapter 4 on page 101

- Euro currency symbol³⁵
- Export To Other Formats³⁶

47.6. F

- Figures³⁷
- Floats³⁸
- Fonts³⁹
- Footer, Page⁴⁰

35 http://en.wikibooks.org/wiki/LaTeX%2FFormatting%23Euro_.E2.

82.AC_currency_symbol

36 Chapter 35 on page 937

37 Chapter 13 on page 399

38 Chapter 13 on page 399

39 Chapter 16 on page 491

40 Chapter 9 on page 273

- Footnotes⁴¹
- Formatting⁴²

47.7. G

- General Guidelines⁴³
- Graphics
- Creating⁴⁴
- Embedding⁴⁵
- Importing⁴⁶

41 <http://en.wikibooks.org/wiki/LaTeX%2FFormatting%23Footnotes>

42 <http://en.wikibooks.org/wiki/LaTeX%2FFormatting>

43 Chapter 30 on page 819

44 Chapter 12 on page 387

45 Chapter 11 on page 343

46 Chapter 11 on page 343

- `graphicx` package⁴⁷

47.8. H

- Header, Page⁴⁸
- HTML output⁴⁹
- Hyperlinks⁵⁰
- `hyperref` package⁵¹
- `hyphen`⁵²

-
- 47 Chapter 11 on page 343
 - 48 Chapter 9 on page 273
 - 49 Chapter 35.4 on page 946
 - 50 Chapter 14 on page 437
 - 51 Chapter 14 on page 437
 - 52 http://en.wikibooks.org/wiki/LaTeX%2FFormatting%23Dashes_and_Hyphens

- Hyphenation⁵³

47.9. I

- i.e. (id est)⁵⁴
- Images⁵⁵
- Importing Graphics⁵⁶
- Indexing⁵⁷
- Internationalization⁵⁸

53 <http://en.wikibooks.org/wiki/LaTeX%2FFormatting%23Hyphenation>

54 Chapter 38.2 on page 1011

55 Chapter 11 on page 343

56 Chapter 11 on page 343

57 Chapter 27 on page 753

58 Chapter 36 on page 953

- Introduction⁵⁹
- Italics⁶⁰
- itemize⁶¹

47.10. L

- Labels⁶²
- Letters⁶³
- Links⁶⁴

59 Chapter 1 on page 5

60 http://en.wikibooks.org/wiki/LaTeX%2FFormatting%23Font_Styles_and_size

61 <http://en.wikibooks.org/wiki/LaTeX%2FFormatting%23Itemize>

62 Chapter 19 on page 647

63 Chapter 27.4 on page 761

64 Chapter 40 on page 1039

- Lists⁶⁵

47.11. M

- `makeidx` package⁶⁶
 - `\maketitle`⁶⁷
 - Margin Notes⁶⁸
 - Creating Graphics⁶⁹
 - Mathematics⁷⁰
-
- 65 Chapter 6 on page 193
 - 66 Chapter 27 on page 753
 - 67 Chapter 3.1.2 on page 81
 - 68 <http://en.wikibooks.org/wiki/LaTeX%2FFormatting%23Margin%20Notes>
 - 69 <http://en.wikibooks.org/wiki/LaTeX%2FMathematical%20Graphics>
 - 70 Chapter 17 on page 501

- Matrices⁷¹
- Minipage environment example⁷²
- Multi-column Pages⁷³

47.12. P

- Packages⁷⁴
 - Creating 1⁷⁵, Creating 2⁷⁶
- Page Layout⁷⁷

71	Chapter 17.10 on page 538
72	Chapter 7.3.3 on page 259
73	Chapter 9.4 on page 307
74	Chapter 32 on page 859
75	Chapter 31.4 on page 854
76	Chapter 33.3 on page 897
77	Chapter 9 on page 273

- PDF output⁷⁸
- picture⁷⁹
- Pictures⁸⁰
- PNG output⁸¹
- Presentations⁸²
- Pseudocode⁸³

47.13. Q

- Quotes⁸⁴

78	Chapter 35.1.4 on page 942
79	Chapter 12 on page 387
80	Chapter 11 on page 343
81	Chapter 35.1.4 on page 942
82	Chapter 28 on page 779
83	Chapter 21 on page 679

84 <http://en.wikibooks.org/wiki/LaTeX%2FFormattin%23Quotes>

47.14. R

- References⁸⁵
- RTF output⁸⁶

47.15. S

- Small Capitals⁸⁷
- Source Code Listings⁸⁸

85 Chapter 19 on page 647

86 Chapter 35.1.4 on page 942

87 <http://en.wikibooks.org/wiki/LaTeX%2FFormatting%23Font%20Styles%20and%20size>

88 <http://en.wikibooks.org/wiki/LaTeX%2FFormatting%23Listing%20Environment>

- Space Between Words⁸⁹
- Spell-checking⁹⁰
- Superscript and subscript: powers and indices⁹¹
- Superscript and subscript: text mode⁹²
- SVG output⁹³

47.16. T

- Table of contents⁹⁴

89	http://en.wikibooks.org/wiki/LaTeX%2FFormattting%23The%20Space%20Between%20Words
90	Chapter 38.6 on page 1026
91	Chapter 17.5 on page 512
92	http://en.wikibooks.org/wiki/LaTeX%2FFormattting%23Text%20mode%20superscript%20and%20subscript
93	Chapter 35.1.4 on page 942
94	Chapter 3.1.7 on page 93

- Tables⁹⁵
- Teletype text⁹⁶
- Text Size⁹⁷
- Theorems⁹⁸
- Tips and Tricks⁹⁹
- Title Creation¹⁰⁰

95 Chapter 7 on page 219

96 <http://en.wikibooks.org/wiki/LaTeX%2FFormatting%23Font%20Styles%20and%20size>

97 Chapter 7.3.3 on page 259

98 Chapter 20 on page 667

99 Chapter 38 on page 1009

100 Chapter 7.3.3 on page 259

47.17. U

- URLs¹⁰¹

47.18. V

- Verbatim Text¹⁰²

101 <http://en.wikibooks.org/wiki/LaTeX%2FFormatting%23Typesetting%20URLs>

102 <http://en.wikibooks.org/wiki/LaTeX%2FFormatting%23Verbatim%20Text>

47.19. W

- Word Counting¹⁰³

47.20. X

- XeTeX¹⁰⁴
- XY-pic package¹⁰⁵
- xy package¹⁰⁶

103 Chapter 38.6 on page 1026

104 Chapter 16.1 on page 494

105 Chapter 12.2.1 on page 394

106 Chapter 12.2.1 on page 394

48. Contributors

Edits	User
140	3mta3 ¹
2	ABCD ²
1	ATC2 ³
1	Aadornelles ⁴

-
- 1 <http://en.wikibooks.org/w/index.php?title=User:3mta3>
 - 2 <http://en.wikibooks.org/w/index.php?title=User:ABCD>
 - 3 <http://en.wikibooks.org/w/index.php?title=User:ATC2>
 - 4 <http://en.wikibooks.org/w/index.php?title=User:Aadornelles>

- 2 Abonnema⁵
- 1 Adam majewski⁶
- 1 Adamcrume⁷
- 1 Adouglass⁸
- 47 Adrignola⁹
- 1 Akim Demaille¹⁰
- 122 Alejo2083¹¹
- 2 AllenZh¹²

5 <http://en.wikibooks.org/w/index.php?title=User:Abonnema>
6 http://en.wikibooks.org/w/index.php?title=User:Adam_
majewski

7 <http://en.wikibooks.org/w/index.php?title=User:Adamcrume>
8 <http://en.wikibooks.org/w/index.php?title=User:Adouglass>
9 <http://en.wikibooks.org/w/index.php?title=User:Adrignola>
10 http://en.wikibooks.org/w/index.php?title=User:Akim_
Demaille

11 <http://en.wikibooks.org/w/index.php?title=User:Alejo2083>
12 <http://en.wikibooks.org/w/index.php?title=User:AllenZh>

- 9 Alzahrawi¹³
- 1 Amamory¹⁴
- 59 Ambrevar¹⁵
- 2 Anarchyboy¹⁶
- 6 Andyr¹⁷
- 5 Ans¹⁸
- 2 Arided¹⁹
- 25 Arnehe²⁰
- 1 Arthurvogel²¹

13 <http://en.wikibooks.org/w/index.php?title=User:Alzahrawi>

14 <http://en.wikibooks.org/w/index.php?title=User:Amamory>

15 <http://en.wikibooks.org/w/index.php?title=User:Ambrevar>

16 <http://en.wikibooks.org/w/index.php?title=User:Anarchyboy>

17 <http://en.wikibooks.org/w/index.php?title=User:Andyr>

18 <http://en.wikibooks.org/w/index.php?title=User:Ans>

19 <http://en.wikibooks.org/w/index.php?title=User:Arided>

20 <http://en.wikibooks.org/w/index.php?title=User:Arnehe>

21 <http://en.wikibooks.org/w/index.php?title=User:Arthurvogel>

- 1 Asmeurer²²
1 Astrophizz²³
1 Atallcostsky²⁴
1 Atiq ur Rehman²⁵
1 Atulya1988²⁶
13 Avila.gas²⁷
1 Bajrangkhichi96²⁸
1 Bakken²⁹
-
- 22 <http://en.wikibooks.org/w/index.php?title=User:Asmeurer>
23 <http://en.wikibooks.org/w/index.php?title=User:Astrophizz>
24 <http://en.wikibooks.org/w/index.php?title=User:Atallcostsky>
25 http://en.wikibooks.org/w/index.php?title=User:Atiq_ur_Rehman
26 <http://en.wikibooks.org/w/index.php?title=User:Atulya1988>
27 <http://en.wikibooks.org/w/index.php?title=User:Avila.gas>
28 [http://en.wikibooks.org/w/index.php?title=User:Atiq_ur_Rehman](http://en.wikibooks.org/w/index.php?title=User:Bajrangkhichi96)
29 <http://en.wikibooks.org/w/index.php?title=User:Bakken>

- 1 Bamgooly³⁰
- 1 Barakafrit³¹
- 4 Basenga³²
- 1 Bcmpinc³³
- 1 Belteshazzar³⁴
- 2 Benjaminevans82³⁵
- 1 Benregn³⁶
- 1 Bhanuvrat³⁷

-
- 30 <http://en.wikibooks.org/w/index.php?title=User:Bamgooly>
 - 31 <http://en.wikibooks.org/w/index.php?title=User:Barakafrit>
 - 32 <http://en.wikibooks.org/w/index.php?title=User:Basenga>
 - 33 <http://en.wikibooks.org/w/index.php?title=User:Bcmpinc>
 - 34 <http://en.wikibooks.org/w/index.php?title=User:Belteshazzar>
 - 35 <http://en.wikibooks.org/w/index.php?title=User:Benjaminevans82>
 - 36 <http://en.wikibooks.org/w/index.php?title=User:Benregn>
 - 37 <http://en.wikibooks.org/w/index.php?title=User:Bhanuvrat>

- 26 **BiT**³⁸
- 1 **Bianbum**³⁹
- 2 **Bilbo1507**⁴⁰
- 1 **Billy the Goat II**⁴¹
- 1 **Blacktrumpeter**⁴²
- 1 **Blaisorblade**⁴³
- 3 **Bonuama**⁴⁴
- 1 **Borgg**⁴⁵
-
- 38 <http://en.wikibooks.org/w/index.php?title=User:BiT>
- 39 <http://en.wikibooks.org/w/index.php?title=User:Bianbum>
- 40 <http://en.wikibooks.org/w/index.php?title=User:Bilbo1507>
- 41 [http://en.wikibooks.org/w/index.php?title=User:Goat_II](http://en.wikibooks.org/w/index.php?title=User:Billy_the_Goat_II)
- 42 <http://en.wikibooks.org/w/index.php?title=User:Blacktrumpeter>
- 43 <http://en.wikibooks.org/w/index.php?title=User:Blaisorblade>
- 44 <http://en.wikibooks.org/w/index.php?title=User:Bonuama>
- 45 <http://en.wikibooks.org/w/index.php?title=User:Borgg>

- 1 Bpsullivan⁴⁶
- 1 Brainrain0000⁴⁷
- 1 Brammers⁴⁸
- 1 Brendanarnold⁴⁹
- 1 BrettMontgomery⁵⁰
- 1 Brevity⁵¹
- 1 Bsander⁵²
- 8 Bumbulski⁵³

-
- 46 <http://en.wikibooks.org/w/index.php?title=User:Bpsullivan>
 - 47 <http://en.wikibooks.org/w/index.php?title=User:Brainrain0000>
 - 48 <http://en.wikibooks.org/w/index.php?title=User:Brammers>
 - 49 <http://en.wikibooks.org/w/index.php?title=User:Brendanarnold>
 - 50 <http://en.wikibooks.org/w/index.php?title=User:BrettMontgomery>
 - 51 <http://en.wikibooks.org/w/index.php?title=User:Brevity>
 - 52 <http://en.wikibooks.org/w/index.php?title=User:Bsander>
 - 53 <http://en.wikibooks.org/w/index.php?title=User:Bumbulski>

- 1 **Byassine**⁵⁴
- 1 **Bytecrook**⁵⁵
- 1 **C3I**⁵⁶
- 7 **CD-Stevens**⁵⁷
- 1 **Caesura**⁵⁸
- 1 **Calimo**⁵⁹
- 3 **Cameronc**⁶⁰
- 1 **Canageek**⁶¹
- 2 **CarsracBot**⁶²

-
- 54 <http://en.wikibooks.org/w/index.php?title=User:Byassine52>
 - 55 <http://en.wikibooks.org/w/index.php?title=User:Bytecrook>
 - 56 <http://en.wikibooks.org/w/index.php?title=User:C3I>
 - 57 <http://en.wikibooks.org/w/index.php?title=User:CD-Stevens>
 - 58 <http://en.wikibooks.org/w/index.php?title=User:Caesura>
 - 59 <http://en.wikibooks.org/w/index.php?title=User:Calimo>
 - 60 <http://en.wikibooks.org/w/index.php?title=User:Cameronc>
 - 61 <http://en.wikibooks.org/w/index.php?title=User:Canageek>
 - 62 <http://en.wikibooks.org/w/index.php?title=User:CarsracBot>

- 2 Cdecoro⁶³
- 3 Cengique⁶⁴
- 5 Cerniagigante⁶⁵
- 1 Cfailde⁶⁶
- 1 Chafe66⁶⁷
- 1 Chazz⁶⁸
- 2 Chisophugis⁶⁹
- 61 ChrisHodgesUK⁷⁰

-
- 63 <http://en.wikibooks.org/w/index.php?title=User:Cdecoro>
 - 64 <http://en.wikibooks.org/w/index.php?title=User:Cengique>
 - 65 <http://en.wikibooks.org/w/index.php?title=User:Cerniagigante>
 - 66 <http://en.wikibooks.org/w/index.php?title=User:Cfailde>
 - 67 <http://en.wikibooks.org/w/index.php?title=User:Chafe66>
 - 68 <http://en.wikibooks.org/w/index.php?title=User:Chazz>
 - 69 <http://en.wikibooks.org/w/index.php?title=User:Chisophugis>
 - 70 <http://en.wikibooks.org/w/index.php?title=User:ChrisHodgesUK>

- 1 ChristianGruen⁷¹
- 1 Chuaprap⁷²
- 1 Chuckhoffmann⁷³
- 1 Clebell⁷⁴
- 1 Codairem⁷⁵
- 1 Collinpark⁷⁶
- 1 Comput2h⁷⁷
- 5 ConditionalZenith⁷⁸

71 <http://en.wikibooks.org/w/index.php?title=User:ChristianGruen>

72 <http://en.wikibooks.org/w/index.php?title=User:Chuaprap>

73 <http://en.wikibooks.org/w/index.php?title=User:Chuckhoffmann>

74 <http://en.wikibooks.org/w/index.php?title=User:Clebell>

75 <http://en.wikibooks.org/w/index.php?title=User:Codairem>

76 <http://en.wikibooks.org/w/index.php?title=User:Collinpark>

77 <http://en.wikibooks.org/w/index.php?title=User:Comput2h>

78 <http://en.wikibooks.org/w/index.php?title=User:ConditionalZenith>

- 1 Conighion⁷⁹
- 2 Conrad.Irwin⁸⁰
- 1 Courcelles⁸¹
- 1 Crasic⁸²
- 2 Crasshopper⁸³
- 2 CrazyTerabyte⁸⁴
- 2 Crissov⁸⁵
- 2 Cícero⁸⁶

79 <http://en.wikibooks.org/w/index.php?title=User:Conighion>

80 <http://en.wikibooks.org/w/index.php?title=User:Conrad.Irwin>

81 <http://en.wikibooks.org/w/index.php?title=User:Courcelles>

82 <http://en.wikibooks.org/w/index.php?title=User:Crasic>

83 <http://en.wikibooks.org/w/index.php?title=User:Crasshopper>

84 <http://en.wikibooks.org/w/index.php?title=User:CrazyTerabyte>

85 <http://en.wikibooks.org/w/index.php?title=User:Crissov>

86 <http://en.wikibooks.org/w/index.php?title=User:C%C3%ADcero>

71	Dan Polansky ⁸⁷
1	Danielstrong ⁵² ⁸⁸
1	David.s.hollman ⁸⁹
6	DavidMcKenzie ⁹⁰
1	Debejyo ⁹¹
1	Dendik ⁹²
103	Derbeth ⁹³

87 http://en.wikibooks.org/w/index.php?title=User:Dan_Polansky

88 <http://en.wikibooks.org/w/index.php?title=User:Danielstrong52>

89 <http://en.wikibooks.org/w/index.php?title=User:David.s.hollman>

90 <http://en.wikibooks.org/w/index.php?title=User:DavidMcKenzie>

91 <http://en.wikibooks.org/w/index.php?title=User:Debejyo>

92 <http://en.wikibooks.org/w/index.php?title=User:Dendik>

93 <http://en.wikibooks.org/w/index.php?title=User:Derbeth>

- 1 Derwaldrandfoerster⁹⁴
- 6 Dilaudid⁹⁵
- 66 Dirk Hünninger⁹⁶
- 1 Dlituiev⁹⁷
- 3 Dmb⁹⁸
- 1 DmitriyZotikov⁹⁹
- 3 Dncarley¹⁰⁰
- 1 Dporter¹⁰¹

94 [http://en.wikibooks.org/w/index.php?title=User:](http://en.wikibooks.org/w/index.php?title=User:Derwaldrandfoerster)

Derwaldrandfoerster

95 <http://en.wikibooks.org/w/index.php?title=User:Dilaudid>

96 http://en.wikibooks.org/w/index.php?title=User:Dirk_H%C3%BCnniger

97 <http://en.wikibooks.org/w/index.php?title=User:Dlituiev>

98 <http://en.wikibooks.org/w/index.php?title=User:Dmb>

99 [http://en.wikibooks.org/w/index.php?title=User:](http://en.wikibooks.org/w/index.php?title=User:DmitriyZotikov)

DmitriyZotikov

100 <http://en.wikibooks.org/w/index.php?title=User:Dncarley>

101 <http://en.wikibooks.org/w/index.php?title=User:Dporter>

- 1 DrOpi¹⁰²
- 1 Dreaven3¹⁰³
- 4 Drevicko¹⁰⁴
- 3 Drewbie¹⁰⁵
- 1 Dubbaluga¹⁰⁶
- 3 E.lewis1¹⁰⁷
- 1 Echeban¹⁰⁸
- 6 Ediahist¹⁰⁹
- 1 Edudobay¹¹⁰

-
- 102 <http://en.wikibooks.org/w/index.php?title=User:Dr0pi>
 - 103 <http://en.wikibooks.org/w/index.php?title=User:Dreaven3>
 - 104 <http://en.wikibooks.org/w/index.php?title=User:Drevicko>
 - 105 <http://en.wikibooks.org/w/index.php?title=User:Drewbie>
 - 106 <http://en.wikibooks.org/w/index.php?title=User:Dubbaluga>
 - 107 <http://en.wikibooks.org/w/index.php?title=User:E.lewis1>
 - 108 <http://en.wikibooks.org/w/index.php?title=User:Echeban>
 - 109 <http://en.wikibooks.org/w/index.php?title=User:Ediahist>
 - 110 <http://en.wikibooks.org/w/index.php?title=User:Edudobay>

- 3 Elliptic1¹¹¹
- 1 Elwikipedista¹¹²
- 1 Emreg00¹¹³
- 16 Erylaos¹¹⁴
- 1 Eudoxos¹¹⁵
- 2 EvanKroske¹¹⁶
- 1 Everlong¹¹⁷
- 5 Eyliu¹¹⁸
- 3 Ffangs¹¹⁹

111 <http://en.wikibooks.org/w/index.php?title=User:Elliptic1>
112 <http://en.wikibooks.org/w/index.php?title=User:>

Elwikipedista

113 <http://en.wikibooks.org/w/index.php?title=User:Emreg00>
114 <http://en.wikibooks.org/w/index.php?title=User:Erylaos>
115 <http://en.wikibooks.org/w/index.php?title=User:Eudoxos>
116 <http://en.wikibooks.org/w/index.php?title=User:EvanKroske>
117 <http://en.wikibooks.org/w/index.php?title=User:Everlong>
118 <http://en.wikibooks.org/w/index.php?title=User:Eyliu>
119 <http://en.wikibooks.org/w/index.php?title=User:Ffangs>

- 2 Filip Dominec¹²⁰
- 1 Fishix¹²¹
- 2 Fishpi¹²²
- 1 Flal¹²³
- 1 FlashSheridan¹²⁴
- 1 Fmccown¹²⁵
- 1 Fongs¹²⁶
- 4 Franzl aus tirol¹²⁷

120 http://en.wikibooks.org/w/index.php?title=User:Filip_

Dominec

121 <http://en.wikibooks.org/w/index.php?title=User:Fishix>

122 <http://en.wikibooks.org/w/index.php?title=User:Fishpi>

123 <http://en.wikibooks.org/w/index.php?title=User:Flal>

124 <http://en.wikibooks.org/w/index.php?title=User:>

FlashSheridan

125 <http://en.wikibooks.org/w/index.php?title=User:Fmccown>

126 <http://en.wikibooks.org/w/index.php?title=User:Fongs>

127 http://en.wikibooks.org/w/index.php?title=User:Franzl_

tirol

- 1 Frap¹²⁸
- 1 Fredmaranhao¹²⁹
- 1 Fsart¹³⁰
- 1 Ftravers¹³¹
- 1 GPHemsley¹³²
- 1 Gallen01¹³³
- 1 Garoth¹³⁴
- 1 GavinMcGimpsey¹³⁵

128 <http://en.wikibooks.org/w/index.php?title=User:Frap>
129 <http://en.wikibooks.org/w/index.php?title=User:>

Fredmaranhao

130 <http://en.wikibooks.org/w/index.php?title=User:Fstart>
131 <http://en.wikibooks.org/w/index.php?title=User:Ftravers>
132 <http://en.wikibooks.org/w/index.php?title=User:GPHemsley>
133 <http://en.wikibooks.org/w/index.php?title=User:Gallen01>
134 <http://en.wikibooks.org/w/index.php?title=User:Garoth>
135 <http://en.wikibooks.org/w/index.php?title=User:>
GavinMcGimpsey

- 3 Geetha nitc¹³⁶
- 2 Gelbukh¹³⁷
- 1 Geminatea¹³⁸
- 1 Germanzs¹³⁹
- 2 Ghoti¹⁴⁰
- 4 Gkc¹⁴¹
- 1 Glad¹⁴²
- 3 Glosser.ca¹⁴³
- 1 Gmh04¹⁴⁴

-
- 136 http://en.wikibooks.org/w/index.php?title=User:Geetha_nitc
 - 137 <http://en.wikibooks.org/w/index.php?title=User:Gelbukh>
 - 138 <http://en.wikibooks.org/w/index.php?title=User:Geminatea>
 - 139 <http://en.wikibooks.org/w/index.php?title=User:Germanzs>
 - 140 <http://en.wikibooks.org/w/index.php?title=User:Ghoti>
 - 141 <http://en.wikibooks.org/w/index.php?title=User:Gkc>
 - 142 <http://en.wikibooks.org/w/index.php?title=User:Glad>
 - 143 <http://en.wikibooks.org/w/index.php?title=User:Glosser.ca>
 - 144 <http://en.wikibooks.org/w/index.php?title=User:Gmh04>

- 7 Gms¹⁴⁵
- 1 Go.pbam.¹⁴⁶
- 1 Goodgerster¹⁴⁷
- 1 Graemeg¹⁴⁸
- 1 Greenbreen¹⁴⁹
- 1 Grenouille¹⁵⁰
- 1 Grj23¹⁵¹
- 2 Gronau¹⁵²
- 1 Guyrobbie¹⁵³

-
- 145 <http://en.wikibooks.org/w/index.php?title=User:Gms>
 - 146 <http://en.wikibooks.org/w/index.php?title=User:Go.pbam.>
 - 147 <http://en.wikibooks.org/w/index.php?title=User:Goodgerster>
 - 148 <http://en.wikibooks.org/w/index.php?title=User:Graemeg>
 - 149 <http://en.wikibooks.org/w/index.php?title=User:Greenbreen>
 - 150 <http://en.wikibooks.org/w/index.php?title=User:Grenouille>
 - 151 <http://en.wikibooks.org/w/index.php?title=User:Grj23>
 - 152 <http://en.wikibooks.org/w/index.php?title=User:Gronau>
 - 153 <http://en.wikibooks.org/w/index.php?title=User:Guyrobbie>

- 1 Gwpl¹⁵⁴
- 1 Gyro Copter¹⁵⁵
- 1 Götz¹⁵⁶
- 2 HJMills¹⁵⁷
- 1 Hagindaz¹⁵⁸
- 7 Hankjones¹⁵⁹
- 1 Hankwang¹⁶⁰
- 1 Hannes Röst¹⁶¹
- 2 Hansfn¹⁶²

-
- 154 <http://en.wikibooks.org/w/index.php?title=User:Gwpl>
 - 155 http://en.wikibooks.org/w/index.php?title=User:Gyro_Copter
 - 156 <http://en.wikibooks.org/w/index.php?title=User:G%C3%B6tz>
 - 157 <http://en.wikibooks.org/w/index.php?title=User:HJMills>
 - 158 <http://en.wikibooks.org/w/index.php?title=User:Hagindaz>
 - 159 <http://en.wikibooks.org/w/index.php?title=User:Hankjones>
 - 160 <http://en.wikibooks.org/w/index.php?title=User:Hankwang>
 - 161 http://en.wikibooks.org/w/index.php?title=User:Hannes_R%C3%B6st
 - 162 <http://en.wikibooks.org/w/index.php?title=User:Hansfn>

- 3 Harrikoo¹⁶³
- 1 Harrywt¹⁶⁴
- 1 Helder.wiki¹⁶⁵
- 4 Helptry¹⁶⁶
- 3 HenrikMidtiby¹⁶⁷
- 4 Henry Tallboys¹⁶⁸
- 5 Henrybissonnette¹⁶⁹
- 1 Herbythyme¹⁷⁰

-
- 163 <http://en.wikibooks.org/w/index.php?title=User:Harrikoo>
 - 164 <http://en.wikibooks.org/w/index.php?title=User:Harrywt>
 - 165 <http://en.wikibooks.org/w/index.php?title=User:Helder.wiki>
 - 166 <http://en.wikibooks.org/w/index.php?title=User:Helptry>
 - 167 <http://en.wikibooks.org/w/index.php?title=User:HenrikMidtiby>
 - 168 http://en.wikibooks.org/w/index.php?title=User:Henry_Tallboys
 - 169 <http://en.wikibooks.org/w/index.php?title=User:Henrybissonnette>
 - 170 <http://en.wikibooks.org/w/index.php?title=User:Herbythyme>

- 2 Hermine potter¹⁷¹
- 1 Hippasus¹⁷²
- 1 Hjsb¹⁷³
- 1 Hokiehead¹⁷⁴
- 6 Hosszuka¹⁷⁵
- 3 Hroobjartr¹⁷⁶
- 1 ILubeMyCucumbers20¹⁷⁷
- 1 Iamunknown¹⁷⁸

-
- 171 http://en.wikibooks.org/w/index.php?title=User:Hermine_potter
 - 172 <http://en.wikibooks.org/w/index.php?title=User:Hippasus>
 - 173 <http://en.wikibooks.org/w/index.php?title=User:Hjsb>
 - 174 <http://en.wikibooks.org/w/index.php?title=User:Hokiehead>
 - 175 <http://en.wikibooks.org/w/index.php?title=User:Hosszuka>
 - 176 <http://en.wikibooks.org/w/index.php?title=User:Hroobjartr>
 - 177 <http://en.wikibooks.org/w/index.php?title=User:ILubeMyCucumbers20>
 - 178 <http://en.wikibooks.org/w/index.php?title=User:Iamunknown>

- 1 Icc97¹⁷⁹
- 26 Igjimh¹⁸⁰
- 1 Immae¹⁸¹
- 1 Inductiveload¹⁸²
- 1 Infenwe¹⁸³
- 2 Insaneinside¹⁸⁴
- 5 InverseHypercube¹⁸⁵
- 1 IrfanAli¹⁸⁶

-
- 179 <http://en.wikibooks.org/w/index.php?title=User:Icc97>
 - 180 <http://en.wikibooks.org/w/index.php?title=User:Igjimh>
 - 181 <http://en.wikibooks.org/w/index.php?title=User:Immae>
 - 182 <http://en.wikibooks.org/w/index.php?title=User:Inductiveload>
 - 183 <http://en.wikibooks.org/w/index.php?title=User:Infenwe>
 - 184 <http://en.wikibooks.org/w/index.php?title=User:Insaneinside>
 - 185 <http://en.wikibooks.org/w/index.php?title=User:InverseHypercube>
 - 186 <http://en.wikibooks.org/w/index.php?title=User:IrfanAli>

- 23 Ish ishwar¹⁸⁷
- 1 Itai¹⁸⁸
- 4 JECompton¹⁸⁹
- 5 JV¹⁹⁰
- 6 Jacho¹⁹¹
- 1 JackPotte¹⁹²
- 2 Jacobrothstein¹⁹³
- 2 Jamoroch¹⁹⁴

187 http://en.wikibooks.org/w/index.php?title=User:Ish_ishwar

188 <http://en.wikibooks.org/w/index.php?title=User:Itai>

189 <http://en.wikibooks.org/w/index.php?title=User:JECompton>

190 <http://en.wikibooks.org/w/index.php?title=User:JV>

191 <http://en.wikibooks.org/w/index.php?title=User:Jacho>

192 <http://en.wikibooks.org/w/index.php?title=User:JackPotte>

193 <http://en.wikibooks.org/w/index.php?title=User:Jacobrothstein>

194 <http://en.wikibooks.org/w/index.php?title=User:Jamoroch>

- 1 Jan Winnicki¹⁹⁵
- 9 Janltx¹⁹⁶
- 1 Janskalicky¹⁹⁷
- 2 Jason barrington¹⁹⁸
- 1 Jasu¹⁹⁹
- 1 Javalenok²⁰⁰
- 1 Jayk²⁰¹
- 1 Je ne détiens pas la vérité universelle²⁰²

195 http://en.wikibooks.org/w/index.php?title=User:Jan_Winnicki

196 <http://en.wikibooks.org/w/index.php?title=User:Janltx>

197 <http://en.wikibooks.org/w/index.php?title=User:Janskalicky>

198 http://en.wikibooks.org/w/index.php?title=User:Jason_barrington

199 <http://en.wikibooks.org/w/index.php?title=User:Jasu>

200 <http://en.wikibooks.org/w/index.php?title=User:Javalenok>

201 <http://en.wikibooks.org/w/index.php?title=User:Jayk>

202 http://en.wikibooks.org/w/index.php?title=User:Je_ne_d%C3%A9tiens_pas_la_v%C3%A9rit%C3%A9_universelle

- 2 JenVan²⁰³
- 1 Jer789²⁰⁴
- 3 Jevon²⁰⁵
- 5 Jflycn²⁰⁶
- 3 Jguk²⁰⁷
- 51 Jimbotyson²⁰⁸
- 1 Jld²⁰⁹
- 1 Jlrn²¹⁰
- 1 Jmcdon10²¹¹

-
- 203 <http://en.wikibooks.org/w/index.php?title=User:JenVan>
 - 204 <http://en.wikibooks.org/w/index.php?title=User:Jer789>
 - 205 <http://en.wikibooks.org/w/index.php?title=User:Jevon>
 - 206 <http://en.wikibooks.org/w/index.php?title=User:Jflycn>
 - 207 <http://en.wikibooks.org/w/index.php?title=User:Jguk>
 - 208 <http://en.wikibooks.org/w/index.php?title=User:Jimbotyson>
 - 209 <http://en.wikibooks.org/w/index.php?title=User:Jld>
 - 210 <http://en.wikibooks.org/w/index.php?title=User:Jlrn>
 - 211 <http://en.wikibooks.org/w/index.php?title=User:Jmcdon10>

- 1 Joaspam²¹²
- 1 Jodi.a.schneider²¹³
- 2 Joe Schmedley²¹⁴
- 1 Joeyboi²¹⁵
- 1 John1923²¹⁶
- 12 Jomegat²¹⁷
- 30 Jonathan Webley²¹⁸
- 1 Jotomicron²¹⁹

-
- 212 <http://en.wikibooks.org/w/index.php?title=User:Joaspam>
 - 213 <http://en.wikibooks.org/w/index.php?title=User:Jodi.a.schneider>
 - 214 http://en.wikibooks.org/w/index.php?title=User:Joe_Schmedley
 - 215 <http://en.wikibooks.org/w/index.php?title=User:Joeyboi>
 - 216 <http://en.wikibooks.org/w/index.php?title=User:John1923>
 - 217 <http://en.wikibooks.org/w/index.php?title=User:Jomegat>
 - 218 http://en.wikibooks.org/w/index.php?title=User:Jonathan_Webley
 - 219 <http://en.wikibooks.org/w/index.php?title=User:Jotomicron>

- 1 Jraregris²²⁰
28 Jtwdog²²¹
2 Juliabackhausen²²²
5 Juliusross²²³
1 Justin W Smith²²⁴
1 Jwchong²²⁵
2 Karcih²²⁶
2 Karper²²⁷

-
- 220 <http://en.wikibooks.org/w/index.php?title=User:Jraregris>
221 <http://en.wikibooks.org/w/index.php?title=User:Jtwdog>
222 <http://en.wikibooks.org/w/index.php?title=User:Juliabackhausen>
223 <http://en.wikibooks.org/w/index.php?title=User:Juliusross>
224 http://en.wikibooks.org/w/index.php?title=User:Justin_W_Smith
225 <http://en.wikibooks.org/w/index.php?title=User:Jwchong>
226 <http://en.wikibooks.org/w/index.php?title=User:Karcih>
227 <http://en.wikibooks.org/w/index.php?title=User:Karper>

- 1 Karthicknainar²²⁸
- 7 Kazkaskazkasako²²⁹
- 1 Kcho²³⁰
- 4 Kejia²³¹
- 1 Kenyon²³²
- 7 Keplerspeed²³³
- 1 Kernigh²³⁴
- 2 Kevang²³⁵

228 <http://en.wikibooks.org/w/index.php?title=User:Karthicknainar>

229 <http://en.wikibooks.org/w/index.php?title=User:Kazkaskazkasako>

230 <http://en.wikibooks.org/w/index.php?title=User:Kcho>

231 <http://en.wikibooks.org/w/index.php?title=User:Kejia>

232 <http://en.wikibooks.org/w/index.php?title=User:Kenyon>

233 <http://en.wikibooks.org/w/index.php?title=User:Keplerspeed>

234 <http://en.wikibooks.org/w/index.php?title=User:Kernigh>

235 <http://en.wikibooks.org/w/index.php?title=User:Kevang>

- 1 Kevinfiesta²³⁶
- 1 KlausFoehl²³⁷
- 1 Klusinyan²³⁸
- 1 Konteki²³⁹
- 14 KovianyO²⁴⁰
- 1 Kpym²⁴¹
- 14 Kri²⁴²
- 1 Krischik²⁴³
- 1 Krishnavedala²⁴⁴

236 <http://en.wikibooks.org/w/index.php?title=User:Kevinfiesta>

237 <http://en.wikibooks.org/w/index.php?title=User:KlausFoehl>

238 <http://en.wikibooks.org/w/index.php?title=User:Klusinyan>

239 <http://en.wikibooks.org/w/index.php?title=User:Konteki>

240 <http://en.wikibooks.org/w/index.php?title=User:KovianyO>

241 <http://en.wikibooks.org/w/index.php?title=User:Kpym>

242 <http://en.wikibooks.org/w/index.php?title=User:Kri>

243 <http://en.wikibooks.org/w/index.php?title=User:Krischik>

244 <http://en.wikibooks.org/w/index.php?title=User:>

- 3 Krisrose²⁴⁵
- 2 Kroolik²⁴⁶
- 4 Krst²⁴⁷
- 1 Kubieziel²⁴⁸
- 2 Kwetal²⁴⁹
- 1 LQST²⁵⁰
- 8 LR²⁵¹
- 1 LaTeX²⁵²
- 2 Latexing²⁵³

245 <http://en.wikibooks.org/w/index.php?title=User:Krisrose>

246 <http://en.wikibooks.org/w/index.php?title=User:Kroolik>

247 <http://en.wikibooks.org/w/index.php?title=User:Krst>

248 <http://en.wikibooks.org/w/index.php?title=User:Kubieziel>

249 <http://en.wikibooks.org/w/index.php?title=User:Kwetal>

250 <http://en.wikibooks.org/w/index.php?title=User:LQST>

251 <http://en.wikibooks.org/w/index.php?title=User:LR>

252 <http://en.wikibooks.org/w/index.php?title=User:LaTeX>

253 <http://en.wikibooks.org/w/index.php?title=User:Latexing>

- 1 Lavaka²⁵⁴
- 1 Lbailey45²⁵⁵
- 1 Leyo²⁵⁶
- 1 LinuxChristian²⁵⁷
- 3 Listdata²⁵⁸
- 1 Literaturgenerator²⁵⁹
- 1 Lnkbuidingservices4u²⁶⁰
- 1 Louabill²⁶¹

-
- 254 <http://en.wikibooks.org/w/index.php?title=User:Lavaka>
 - 255 <http://en.wikibooks.org/w/index.php?title=User:Lbailey45>
 - 256 <http://en.wikibooks.org/w/index.php?title=User:Leyo>
 - 257 <http://en.wikibooks.org/w/index.php?title=User:LinuxChristian>
 - 258 <http://en.wikibooks.org/w/index.php?title=User:Listdata>
 - 259 <http://en.wikibooks.org/w/index.php?title=User:Literaturgenerator>
 - 260 <http://en.wikibooks.org/w/index.php?title=User:Lnkbuidingservices4u>
 - 261 <http://en.wikibooks.org/w/index.php?title=User:Louabill>

- 3 **Louisix**²⁶²
- 3 **Lucasreddinger**²⁶³
- 1 **MER-C**²⁶⁴
- 1 **MaBoehm**²⁶⁵
- 3 **Maartenweyn**²⁶⁶
- 1 **Madskaddie**²⁶⁷
- 2 **MagnusPI**²⁶⁸
- 1 **MarSraM**²⁶⁹
- 1 **Maratonda**²⁷⁰

262 <http://en.wikibooks.org/w/index.php?title=User:Louisix>

263 <http://en.wikibooks.org/w/index.php?title=User:>

Lucasreddinger

- 264 <http://en.wikibooks.org/w/index.php?title=User:MER-C>
- 265 <http://en.wikibooks.org/w/index.php?title=User:MaBoehm>
- 266 <http://en.wikibooks.org/w/index.php?title=User:Maartenweyn>
- 267 <http://en.wikibooks.org/w/index.php?title=User:Madskaddie>
- 268 <http://en.wikibooks.org/w/index.php?title=User:MagnusPI>
- 269 <http://en.wikibooks.org/w/index.php?title=User:MarSraM>
- 270 <http://en.wikibooks.org/w/index.php?title=User:Maratonda>

- 4 Marzols²⁷¹
- 2 Marra²⁷²
- 1 Martin scharrer²⁷³
- 1 Martin von Wittich²⁷⁴
- 6 MartinSpace²⁷⁵
- 1 Martinkunev²⁷⁶
- 2 Masterpiga²⁷⁷

-
- 271 <http://en.wikibooks.org/w/index.php?title=User:Marzols>
 - 272 <http://en.wikibooks.org/w/index.php?title=User:Marra>
 - 273 http://en.wikibooks.org/w/index.php?title=User:Martin_scharrer
 - 274 http://en.wikibooks.org/w/index.php?title=User:Martin_von_Wittich
 - 275 <http://en.wikibooks.org/w/index.php?title=User:MartinSpace>
 - 276 <http://en.wikibooks.org/w/index.php?title=User:Martinkunev>
 - 277 <http://en.wikibooks.org/w/index.php?title=User:Masterpiga>

- 1 Matej.korvas²⁷⁸
2 Mathieu Perrin²⁷⁹
1 Matthias M.²⁸⁰
1 Mckay²⁸¹
19 Mcld²⁸²
1 Mecanismo²⁸³
1 Merciadriluca²⁸⁴
4 Mhue²⁸⁵
-
- 278 <http://en.wikibooks.org/w/index.php?title=User:Matej.korvas>
279 http://en.wikibooks.org/w/index.php?title=User:Mathieu_Perrin
280 http://en.wikibooks.org/w/index.php?title=User:Matthias_M.
281 <http://en.wikibooks.org/w/index.php?title=User:Mckay>
282 <http://en.wikibooks.org/w/index.php?title=User:Mcld>
283 <http://en.wikibooks.org/w/index.php?title=User:Mecanismo>
284 <http://en.wikibooks.org/w/index.php?title=User:Merciadriluca>
285 <http://en.wikibooks.org/w/index.php?title=User:Mhue>

- 1 MichaelBueker²⁸⁶
- 1 Mietchen²⁸⁷
- 1 Mijikenda²⁸⁸
- 1 Mike.lifeguard²⁸⁹
- 2 Mikhail Ryazanov²⁹⁰
- 2 Mimo²⁹¹
- 1 MoMaT²⁹²

286 [http://en.wikibooks.org/w/index.php?title=User:](http://en.wikibooks.org/w/index.php?title=User:MichaelBueker)

MichaelBueker

287 <http://en.wikibooks.org/w/index.php?title=User:Mietchen>

288 <http://en.wikibooks.org/w/index.php?title=User:Mijikenda>

289 <http://en.wikibooks.org/w/index.php?title=User:Mike.lifeguard>

290 <http://en.wikibooks.org/w/index.php?title=User:MikhailRyazanov>

291 <http://en.wikibooks.org/w/index.php?title=User:Mimo>

292 <http://en.wikibooks.org/w/index.php?title=User:MoMaT>

- 1 Modest Genius²⁹³
- 3 Morelight²⁹⁴
- 1 Mouselb²⁹⁵
- 1 Mrt doulaty²⁹⁶
- 1 Ms2ger²⁹⁷
- 84 Mwtoews²⁹⁸
- 2 Naught101²⁹⁹
- 1 NavarroJ³⁰⁰
- 3 Nbrouard³⁰¹

293 http://en.wikibooks.org/w/index.php?title=User:Modest_

Genius

- 294 <http://en.wikibooks.org/w/index.php?title=User:Morelight>
- 295 <http://en.wikibooks.org/w/index.php?title=User:Mouselb>
- 296 http://en.wikibooks.org/w/index.php?title=User:Mrt_doulaty
- 297 <http://en.wikibooks.org/w/index.php?title=User:Ms2ger>
- 298 <http://en.wikibooks.org/w/index.php?title=User:Mwtoews>
- 299 <http://en.wikibooks.org/w/index.php?title=User:Naught101>
- 300 <http://en.wikibooks.org/w/index.php?title=User:NavarroJ>
- 301 <http://en.wikibooks.org/w/index.php?title=User:Nbrouard>

- 21 Neet³⁰²
- 1 Nemoniac³⁰³
- 1 Nemti³⁰⁴
- 2 Neoptolemus³⁰⁵
- 5 Neoriddle³⁰⁶
- 1 Netheril96³⁰⁷
- 1 Niel.Bowerman³⁰⁸
- 1 Nigels³⁰⁹
- 3 Nixphoeni³¹⁰

302 <http://en.wikibooks.org/w/index.php?title=User:Neet>

303 <http://en.wikibooks.org/w/index.php?title=User:Nemoniac>

304 <http://en.wikibooks.org/w/index.php?title=User:Nemti>

305 <http://en.wikibooks.org/w/index.php?title=User:Neoptolemus>

306 <http://en.wikibooks.org/w/index.php?title=User:Neoriddle>

307 <http://en.wikibooks.org/w/index.php?title=User:Netheril96>

308 <http://en.wikibooks.org/w/index.php?title=User:Niel>.

Bowerman

309 <http://en.wikibooks.org/w/index.php?title=User:Nigels>

310 <http://en.wikibooks.org/w/index.php?title=User:Nixphoeni>

- 1 Nkour³¹¹
- 1 Norbert.beckers³¹²
- 1 Nothing1212³¹³
- 1 NqpZ³¹⁴
- 4 Nsuwan³¹⁵
- 1 Nlux³¹⁶
- 1 Obelyaev³¹⁷
- 1 Oderbolz³¹⁸
- 1 Ojan³¹⁹

311 <http://en.wikibooks.org/w/index.php?title=User:Nkour>
312 <http://en.wikibooks.org/w/index.php?title=User:Norbert.beckers>

313 <http://en.wikibooks.org/w/index.php?title=User:Nothing1212>
314 <http://en.wikibooks.org/w/index.php?title=User:NqpZ>
315 <http://en.wikibooks.org/w/index.php?title=User:Nsuwan>
316 <http://en.wikibooks.org/w/index.php?title=User:Nux>
317 <http://en.wikibooks.org/w/index.php?title=User:Obelyaev>
318 <http://en.wikibooks.org/w/index.php?title=User:Oderbolz>
319 <http://en.wikibooks.org/w/index.php?title=User:Ojan>

- 2 Olesh³²⁰
- 1 Olivier.descout³²¹
- 22 Orderud³²²
- 1 Otec Stochastik³²³
- 29 PAC³²⁴
- 25 PAC2³²⁵
- 1 Pamputt³²⁶
- 1 Pandora85³²⁷

-
- 320 <http://en.wikibooks.org/w/index.php?title=User:Olesh>
 - 321 <http://en.wikibooks.org/w/index.php?title=User:Olivier.descout>
 - 322 <http://en.wikibooks.org/w/index.php?title=User:Orderud>
 - 323 http://en.wikibooks.org/w/index.php?title=User:Otec_Stochastik
 - 324 <http://en.wikibooks.org/w/index.php?title=User:PAC>
 - 325 <http://en.wikibooks.org/w/index.php?title=User:PAC2>
 - 326 <http://en.wikibooks.org/w/index.php?title=User:Pamputt>
 - 327 <http://en.wikibooks.org/w/index.php?title=User: Pandora85>

- 3 Panic2k4³²⁸
- 3 PatrickDevlin21³²⁹
- 1 PatrickGalyon³³⁰
- 4 Paxinum³³¹
- 2 Pdelong³³²
- 1 PeterAllen³³³
- 2 Petter Strandmark³³⁴
- 2 PhilJohnG³³⁵

-
- 328 <http://en.wikibooks.org/w/index.php?title=User:Panic2k4>
 - 329 <http://en.wikibooks.org/w/index.php?title=User:PatrickDevlin21>
 - 330 <http://en.wikibooks.org/w/index.php?title=User:PatrickGalyon>
 - 331 <http://en.wikibooks.org/w/index.php?title=User:Paxinum>
 - 332 <http://en.wikibooks.org/w/index.php?title=User:Pdelong>
 - 333 <http://en.wikibooks.org/w/index.php?title=User:PeterAllen>
 - 334 http://en.wikibooks.org/w/index.php?title=User:Petter_Strandmark
 - 335 <http://en.wikibooks.org/w/index.php?title=User:PhilJohnG>

- 82 Pi zero³³⁶
- 11 Piksi³³⁷
- 13 Pmillerrhodes³³⁸
- 1 Pmlineditor³³⁹
- 1 Polytropos Technikos³⁴⁰
- 1 Ppadmapriya³⁴¹
- 1 Prawojazdy³⁴²
- 1 Prispartmentlow³⁴³

336 http://en.wikibooks.org/w/index.php?title=User:Pi_zero
337 <http://en.wikibooks.org/w/index.php?title=User:Piksi>
338 [http://en.wikibooks.org/w/index.php?title=User:](http://en.wikibooks.org/w/index.php?title=User:Pmillerrhodes)

Pmillerrhodes

339 <http://en.wikibooks.org/w/index.php?title=User:Pmlineditor>
340 [http://en.wikibooks.org/w/index.php?title=User:Polytropos_](http://en.wikibooks.org/w/index.php?title=User:Polytropos_Technikos)
Technikos

341 <http://en.wikibooks.org/w/index.php?title=User:Ppadmapriya>
342 <http://en.wikibooks.org/w/index.php?title=User:Prawojazdy>
343 <http://en.wikibooks.org/w/index.php?title=User:Prispartmentlow>

- 7 Pstar³⁴⁴
- 1 PsyberS³⁴⁵
- 2 QUBot³⁴⁶
- 1 Qeny³⁴⁷
- 1 QuantumEleven³⁴⁸
- 1 Quaristice³⁴⁹
- 8 QuiteUnusual³⁵⁰
- 3 Qwertyus³⁵¹

344 <http://en.wikibooks.org/w/index.php?title=User:Pstar>
345 <http://en.wikibooks.org/w/index.php?title=User:Psybers>
346 <http://en.wikibooks.org/w/index.php?title=User:QUBot>
347 <http://en.wikibooks.org/w/index.php?title=User:Qeny>
348 [http://en.wikibooks.org/w/index.php?title=User:](http://en.wikibooks.org/w/index.php?title=User:QuantumEleven)

QuantumEleven

349 <http://en.wikibooks.org/w/index.php?title=User:Quaristice>
350 [http://en.wikibooks.org/w/index.php?title=User:](http://en.wikibooks.org/w/index.php?title=User:QuiteUnusual)
QuiteUnusual
351 <http://en.wikibooks.org/w/index.php?title=User:Qwertyus>

- 1 Rafaelgr³⁵²
- 1 Rafopar³⁵³
- 3 Rajkiran g³⁵⁴
- 3 Ramac³⁵⁵
- 1 Raphael Ackermann³⁵⁶
- 1 Raylu³⁵⁷
- 1 RaymondSutanto³⁵⁸
- 3 Rbonvall³⁵⁹

-
- 352 <http://en.wikibooks.org/w/index.php?title=User:Rafaelgr>
 - 353 <http://en.wikibooks.org/w/index.php?title=User:Rafopar>
 - 354 http://en.wikibooks.org/w/index.php?title=User:Rajkiran_g
 - 355 <http://en.wikibooks.org/w/index.php?title=User:Ramac>
 - 356 http://en.wikibooks.org/w/index.php?title=User:Raphael_Ackermann
 - 357 <http://en.wikibooks.org/w/index.php?title=User:Raylu>
 - 358 <http://en.wikibooks.org/w/index.php?title=User:RaymondSutanto>
 - 359 <http://en.wikibooks.org/w/index.php?title=User:Rbonvall>

- 1 Rdg nz³⁶⁰
- 17 RealSebix³⁶¹
- 8 Recent Runes³⁶²
- 1 Redirect fixer³⁶³
- 4 Rehoot³⁶⁴
- 1 Ricordisamoa³⁶⁵
- 1 Risk³⁶⁶
- 1 Rnddim³⁶⁷

-
- 360 http://en.wikibooks.org/w/index.php?title=User:Rdg_nz
 - 361 <http://en.wikibooks.org/w/index.php?title=User:RealSebix>
 - 362 http://en.wikibooks.org/w/index.php?title=User:Recent_Runes
 - 363 http://en.wikibooks.org/w/index.php?title=User:Redirect_fixer
 - 364 <http://en.wikibooks.org/w/index.php?title=User:Rehoot>
 - 365 <http://en.wikibooks.org/w/index.php?title=User:Ricordisamoa>
 - 366 <http://en.wikibooks.org/w/index.php?title=User:Risk>
 - 367 <http://en.wikibooks.org/w/index.php?title=User:Rnddim>

- 1 Roarbakk³⁶⁸
- 5 Robbiemorrison³⁶⁹
- 1 Robert Borkowski³⁷⁰
- 4 Robert Horning³⁷¹
- 3 Robin³⁷²
- 1 Rogal³⁷³
- 1 Rogerbrent³⁷⁴

-
- 368 <http://en.wikibooks.org/w/index.php?title=User:Roarbakk>
 - 369 <http://en.wikibooks.org/w/index.php?title=User:Robbiemorrison>
 - 370 http://en.wikibooks.org/w/index.php?title=User:Robert_Borkowski
 - 371 http://en.wikibooks.org/w/index.php?title=User:Robert_Horning
 - 372 <http://en.wikibooks.org/w/index.php?title=User:Robin>
 - 373 <http://en.wikibooks.org/w/index.php?title=User:Rogal>
 - 374 <http://en.wikibooks.org/w/index.php?title=User:Rogerbrent>

- 4 Rondenaranja³⁷⁵
- 2 Rossdub³⁷⁶
- 1 Rror³⁷⁷
- 4 Sabalka³⁷⁸
- 35 SamuelLB³⁷⁹
- 4 Sanderd17³⁸⁰
- 1 Sandman10000³⁸¹
- 1 Sandrobt³⁸²

375 <http://en.wikibooks.org/w/index.php?title=User:Rondenaranja>

376 <http://en.wikibooks.org/w/index.php?title=User:Rossdub>

377 <http://en.wikibooks.org/w/index.php?title=User:Rror>

378 <http://en.wikibooks.org/w/index.php?title=User:Sabalka>

379 <http://en.wikibooks.org/w/index.php?title=User:SamuelLB>

380 <http://en.wikibooks.org/w/index.php?title=User:Sanderd17>

381 <http://en.wikibooks.org/w/index.php?title=User:Sandman10000>

382 <http://en.wikibooks.org/w/index.php?title=User:Sandrobt>

- 14 Sargas³⁸³
- 1 Sbeyer³⁸⁴
- 2 Schaber³⁸⁵
- 1 SciYann³⁸⁶
- 1 Scientific29³⁸⁷
- 1 Scorwin³⁸⁸
- 1 Scrus³⁸⁹
- 2 Semperos³⁹⁰
- 1 Sgenier³⁹¹

383 <http://en.wikibooks.org/w/index.php?title=User:Sargas>
384 <http://en.wikibooks.org/w/index.php?title=User:Sbeyer>
385 <http://en.wikibooks.org/w/index.php?title=User:Schaber>
386 <http://en.wikibooks.org/w/index.php?title=User:SciYann>
387 <http://en.wikibooks.org/w/index.php?title=User:Scientific29>

Scientific29

388 <http://en.wikibooks.org/w/index.php?title=User:Scorwin>
389 <http://en.wikibooks.org/w/index.php?title=User:Scrus>
390 <http://en.wikibooks.org/w/index.php?title=User:Semperos>
391 <http://en.wikibooks.org/w/index.php?title=User:Sgenier>

- 1 **Shahbaz Youssefi**³⁹²
- 4 **Silca678**³⁹³
- 1 **Silverpie**³⁹⁴
- 1 **Simeon**³⁹⁵
- 2 **Simonjtyler**³⁹⁶
- 1 **SiriusB**³⁹⁷
- 2 **Sjlegg**³⁹⁸
- 1 **Skarakoleva**³⁹⁹
- 1 **Skou**⁴⁰⁰

392 http://en.wikibooks.org/w/index.php?title=User:Shahbaz_Youssefi

393 <http://en.wikibooks.org/w/index.php?title=User:Silca678>

394 <http://en.wikibooks.org/w/index.php?title=User:Silverpie>

395 <http://en.wikibooks.org/w/index.php?title=User:Simeon>

396 <http://en.wikibooks.org/w/index.php?title=User:Simonjtyler>

397 <http://en.wikibooks.org/w/index.php?title=User:SiriusB>

398 <http://en.wikibooks.org/w/index.php?title=User:Sjlegg>

399 <http://en.wikibooks.org/w/index.php?title=User:Skarakoleva>

400 <http://en.wikibooks.org/w/index.php?title=User:Skou>

- 1 Smobl Bobbl⁴⁰¹
- 5 Snaxe920⁴⁰²
- 1 Snoopy67⁴⁰³
- 4 Spag85⁴⁰⁴
- 11 Spelemann⁴⁰⁵
- 4 Spirosdenaxas⁴⁰⁶
- 1 Spook⁴⁰⁷
- 1 Steindani⁴⁰⁸

-
- 401 http://en.wikibooks.org/w/index.php?title=User:Smobl_Bobl
 - 402 <http://en.wikibooks.org/w/index.php?title=User:Snaxe920>
 - 403 <http://en.wikibooks.org/w/index.php?title=User:Snoopy67>
 - 404 <http://en.wikibooks.org/w/index.php?title=User:Spag85>
 - 405 <http://en.wikibooks.org/w/index.php?title=User:Spelemann>
 - 406 <http://en.wikibooks.org/w/index.php?title=User:Spirosdenaxas>
 - 407 <http://en.wikibooks.org/w/index.php?title=User:Spook>
 - 408 <http://en.wikibooks.org/w/index.php?title=User:Steindani>

- 3 Stephan Schneider⁴⁰⁹
- 1 SteveM82⁴¹⁰
- 1 StevenJohnston⁴¹¹
- 7 Stoettner⁴¹²
- 1 Stuples⁴¹³
- 1 Svick⁴¹⁴
- 1 Swift⁴¹⁵
- 1 TFTD⁴¹⁶

409 http://en.wikibooks.org/w/index.php?title=User:Stephan_Schneider

410 <http://en.wikibooks.org/w/index.php?title=User:SteveM82>

411 <http://en.wikibooks.org/w/index.php?title=User:StevenJohnston>

412 <http://en.wikibooks.org/w/index.php?title=User:Stoettner>

413 <http://en.wikibooks.org/w/index.php?title=User:Stuples>

414 <http://en.wikibooks.org/w/index.php?title=User:Svick>

415 <http://en.wikibooks.org/w/index.php?title=User:Swift>

416 <http://en.wikibooks.org/w/index.php?title=User:TFTD>

- 1 **TWiStErRob**⁴¹⁷
- 1 **Tauriel-1**⁴¹⁸
- 3 **Tazquebec**⁴¹⁹
- 1 **Tdomhan**⁴²⁰
- 1 **Teles**⁴²¹
- 1 **Tgwizard**⁴²²
- 5 **Thefrankinator**⁴²³
- 21 **Thenub314**⁴²⁴
- 2 **Thietkeweb**⁴²⁵

417 <http://en.wikibooks.org/w/index.php?title=User:TWiStErRob>
418 <http://en.wikibooks.org/w/index.php?title=User:Tauriel-1>
419 <http://en.wikibooks.org/w/index.php?title=User:Tazquebec>
420 <http://en.wikibooks.org/w/index.php?title=User:Tdomhan>
421 <http://en.wikibooks.org/w/index.php?title=User:Teles>
422 <http://en.wikibooks.org/w/index.php?title=User:Tgwizard>
423 <http://en.wikibooks.org/w/index.php?title=User:Thefrankinator>

424 <http://en.wikibooks.org/w/index.php?title=User:Thenub314>
425 <http://en.wikibooks.org/w/index.php?title=User:Thietkeweb>

- 2 TinyTimZamboni⁴²⁶
- 1 ToematoeAdmn⁴²⁷
- 1 Tom Morris⁴²⁸
- 3 Tom.marcik⁴²⁹
- 124 Tomato86⁴³⁰
- 6 Tomxlawson⁴³¹
- 11 TomyDuby⁴³²
- 2 Topodelapradera⁴³³

426 <http://en.wikibooks.org/w/index.php?title=User:>

TinyTimZamboni

427 <http://en.wikibooks.org/w/index.php?title=User:>

ToematoeAdmn

428 http://en.wikibooks.org/w/index.php?title=User:Tom_Morris

429 <http://en.wikibooks.org/w/index.php?title=User:Tom.marcik>

430 <http://en.wikibooks.org/w/index.php?title=User:Tomato86>

431 <http://en.wikibooks.org/w/index.php?title=User:Tomxlawson>

432 <http://en.wikibooks.org/w/index.php?title=User:TomyDuby>

433 <http://en.wikibooks.org/w/index.php?title=User:>

Topodelapradera

- 5 TorfusPolymorphus⁴³⁴
- 14 Tasha⁴³⁵
- 2 Towsonu2003⁴³⁶
- 1 Tpr⁴³⁷
- 1 Tualha⁴³⁸
- 1 Tuetschek⁴³⁹
- 1 Tuka⁴⁴⁰
- 8 Tully⁴⁴¹
- 1 Tweenk⁴⁴²

434 [http://en.wikibooks.org/w/index.php?title=User:](http://en.wikibooks.org/w/index.php?title=User:TorfusPolymorphus)

TorfusPolymorphus

435 <http://en.wikibooks.org/w/index.php?title=User:Tosha>

436 <http://en.wikibooks.org/w/index.php?title=User:Towsonu2003>

437 <http://en.wikibooks.org/w/index.php?title=User:Tpr>

438 <http://en.wikibooks.org/w/index.php?title=User:Tualha>

439 <http://en.wikibooks.org/w/index.php?title=User:Tuetschek>

440 <http://en.wikibooks.org/w/index.php?title=User:Tuka>

441 <http://en.wikibooks.org/w/index.php?title=User:Tully>

442 <http://en.wikibooks.org/w/index.php?title=User:Tweenk>

- 1 Uluboz⁴⁴³
- 1 Unbitwise⁴⁴⁴
- 1 Unco⁴⁴⁵
- 3 Urhixidur⁴⁴⁶
- 6 Vadik wiki⁴⁴⁷
- 3 Vaffelkake⁴⁴⁸
- 3 Velociostrich⁴⁴⁹
- 10 Vesal⁴⁵⁰
- 70 Waldir⁴⁵¹

-
- 443 <http://en.wikibooks.org/w/index.php?title=User:Uluboz>
 - 444 <http://en.wikibooks.org/w/index.php?title=User:Unbitwise>
 - 445 <http://en.wikibooks.org/w/index.php?title=User:Unco>
 - 446 <http://en.wikibooks.org/w/index.php?title=User:Urhixidur>
 - 447 http://en.wikibooks.org/w/index.php?title=User:Vadik_wiki
 - 448 <http://en.wikibooks.org/w/index.php?title=User:Vaffelkake>
 - 449 <http://en.wikibooks.org/w/index.php?title=User:Velociostrich>
 - 450 <http://en.wikibooks.org/w/index.php?title=User:Vesal>
 - 451 <http://en.wikibooks.org/w/index.php?title=User:Waldir>

- 4 Wdcf⁴⁵²
- 1 Webinn⁴⁵³
- 1 Wenzeslaus⁴⁵⁴
- 1 White gecko⁴⁵⁵
- 1 Whiteknight⁴⁵⁶
- 9 Wikieditoroftoday⁴⁵⁷
- 1 Willy james⁴⁵⁸
- 1 Winfree⁴⁵⁹
- 1 Winniehell⁴⁶⁰

-
- 452 <http://en.wikibooks.org/w/index.php?title=User:Wdcf>
 - 453 <http://en.wikibooks.org/w/index.php?title=User:Webinn>
 - 454 <http://en.wikibooks.org/w/index.php?title=User:Wenzeslaus>
 - 455 http://en.wikibooks.org/w/index.php?title=User:White_gecko
 - 456 <http://en.wikibooks.org/w/index.php?title=User:Whiteknight>
 - 457 <http://en.wikibooks.org/w/index.php?title=User:Wikieditoroftoday>
 - 458 http://en.wikibooks.org/w/index.php?title=User:Willy_james
 - 459 <http://en.wikibooks.org/w/index.php?title=User:Winfree>
 - 460 <http://en.wikibooks.org/w/index.php?title=User:Winniehell>

- 28 **Withinfocus**⁴⁶¹
- 1 **Wkdurfee**⁴⁶²
- 1 **Wmheric**⁴⁶³
- 1 **Wn202**⁴⁶⁴
- 2 **Writalnaie**⁴⁶⁵
- 2 **Wxm29**⁴⁶⁶
- 1 **Wysinwygaa**⁴⁶⁷
- 5 **Xania**⁴⁶⁸
- 1 **Xeracles**⁴⁶⁹

-
- 461 <http://en.wikibooks.org/w/index.php?title=User:Withinfocus>
 - 462 <http://en.wikibooks.org/w/index.php?title=User:Wkdurfee>
 - 463 <http://en.wikibooks.org/w/index.php?title=User:Wmheric>
 - 464 <http://en.wikibooks.org/w/index.php?title=User:Wn202>
 - 465 <http://en.wikibooks.org/w/index.php?title=User:Writalnaie>
 - 466 <http://en.wikibooks.org/w/index.php?title=User:Wxm29>
 - 467 <http://en.wikibooks.org/w/index.php?title=User:Wysinwygaa>
 - 468 <http://en.wikibooks.org/w/index.php?title=User:Xania>
 - 469 <http://en.wikibooks.org/w/index.php?title=User:Xeracles>

- 1 Xnn⁴⁷⁰
- 12 Xonqno⁴⁷¹
- 1 Yanuzz⁴⁷²
- 1 Yez⁴⁷³
- 1 Ynhockey⁴⁷⁴
- 1 Ypey⁴⁷⁵
- 1 Ysnikraz⁴⁷⁶
- 3 ZeroOne⁴⁷⁷
- 1 Zrisher⁴⁷⁸

-
- 470 <http://en.wikibooks.org/w/index.php?title=User:Xnn>
 - 471 <http://en.wikibooks.org/w/index.php?title=User:Xonqno>
 - 472 <http://en.wikibooks.org/w/index.php?title=User:Yanuzz>
 - 473 <http://en.wikibooks.org/w/index.php?title=User:Yez>
 - 474 <http://en.wikibooks.org/w/index.php?title=User:Ynhockey>
 - 475 <http://en.wikibooks.org/w/index.php?title=User:Ypey>
 - 476 <http://en.wikibooks.org/w/index.php?title=User:Ysnikraz>
 - 477 <http://en.wikibooks.org/w/index.php?title=User:ZeroOne>
 - 478 <http://en.wikibooks.org/w/index.php?title=User:Zrisher>

- 3 Zvika⁴⁷⁹
- 1 Zwiebelleder⁴⁸⁰
- 1 Zxx117⁴⁸¹
- 2 Zyqqh⁴⁸²
- 1 Zzo38⁴⁸³
- 1 Ævar Arnfjörð Bjarmason⁴⁸⁴
- 1 Пика Пика⁴⁸⁵

479 <http://en.wikibooks.org/w/index.php?title=User:Zvika>
480 <http://en.wikibooks.org/w/index.php?title=User:Zwiebelleder>

481 <http://en.wikibooks.org/w/index.php?title=User:Zxx117>
482 <http://en.wikibooks.org/w/index.php?title=User:Zyqqh>
483 <http://en.wikibooks.org/w/index.php?title=User:Zzo38>
484 http://en.wikibooks.org/w/index.php?title=User:Arnrfj%C3%B6r%C3%B0_Bjarmason
485 http://en.wikibooks.org/w/index.php?title=User:B8%D0%9F%D0%BA%D0%B0_%D0%9F%D0%B8%D0%BA%D0%B0

1 گن اوع رنام⁴⁸⁶

486 http://en.wikibooks.org/w/index.php?title=User:%D8%A7%D9%85%DB%8C%D8%B1_%D8%A7%D8%B9%D9%88%D8%A7%D9%86%DB%8C

List of Figures

- **GFDL: Gnu Free Documentation License.** <http://www.gnu.org/licenses/fdl.html>
- **cc-by-sa-3.0: Creative Commons Attribution ShareAlike 3.0 License.**
<http://creativecommons.org/licenses/by-sa/3.0/>
- **cc-by-sa-2.5: Creative Commons Attribution ShareAlike 2.5 License.**
<http://creativecommons.org/licenses/by-sa/2.5/>
- **cc-by-sa-2.0: Creative Commons Attribution ShareAlike 2.0 License.**
<http://creativecommons.org/licenses/by-sa/2.0/>

- **cc-by-sa-1.0: Creative Commons Attribution ShareAlike 1.0 License.** <http://creativecommons.org/licenses/by-sa/1.0/>
- **cc-by-2.0: Creative Commons Attribution 2.0 License.** <http://creativecommons.org/licenses/by/2.0/>
- **cc-by-2.0: Creative Commons Attribution 2.0 License.** <http://creativecommons.org/licenses/by/2.0/deed.en>
- **cc-by-2.5: Creative Commons Attribution 2.5 License.** <http://creativecommons.org/licenses/by/2.5/deed.en>
- **cc-by-3.0: Creative Commons Attribution 3.0 License.** <http://creativecommons.org/licenses/by/3.0/deed.en>
- **GPL: GNU General Public License.** <http://www.gnu.org/licenses/gpl-2.0.txt>
- **LGPL: GNU Lesser General Public License.** <http://www.gnu.org/licenses/lgpl.html>

- **PD:** This image is in the public domain.
- **ATTR:** The copyright holder of this file allows anyone to use it for any purpose, provided that the copyright holder is properly attributed. Redistribution, derivative work, commercial use, and all other use is permitted.
- **EURO:** This is the common (reverse) face of a euro coin. The copyright on the design of the common face of the euro coins belongs to the European Commission. Authorised is reproduction in a format without relief (drawings, paintings, films) provided they are not detrimental to the image of the euro.
- **LFK:** Lizenz Freie Kunst. <http://artlibre.org/licence/lal/de>
- **CFR:** Copyright free use.
- **EPL:** Eclipse Public License. <http://www.eclipse.org/org/documents/epl-v10.php>

Copies of the GPL, the LGPL as well as a GFDL are included in chapter Licenses⁴⁸⁷. Please note that images in the public domain do not require attribution. You may click on the image numbers in the following table to open the webpage of the images in your webbrowser.

1	Alessio Damato	GFDL
2	Dirk Hünninger ⁴⁸⁸	cc-by-sa-3.0
3	Dirk Hünninger ⁴⁸⁹	cc-by-sa-3.0
4	Dirk Hünninger ⁴⁹⁰	cc-by-sa-3.0
5	Literaturgenerator ⁴⁹¹	GFDL
6		GFDL
7	Mij	PD
8	Dirk Hünninger ⁴⁹²	cc-by-sa-3.0

488 <http://en.wikibooks.org/wiki/User%3ADirk%20H%Cnniger>

489 <http://en.wikibooks.org/wiki/User%3ADirk%20H%Cnniger>

490 <http://en.wikibooks.org/wiki/User%3ADirk%20H%Cnniger>

491 <http://en.wikibooks.org/wiki/User%3ALiteraturgenerator>

492 <http://en.wikibooks.org/wiki/User%3ADirk%20H%Cnniger>

9	Dirk Hünninger ⁴⁹³	cc-by-sa-3.0
10	Dirk Hünninger ⁴⁹⁴	cc-by-sa-3.0
11	Dirk Hünninger ⁴⁹⁵	cc-by-sa-3.0
12	Dirk Hünninger ⁴⁹⁶	cc-by-sa-3.0
13	Dirk Hünninger ⁴⁹⁷	cc-by-sa-3.0
14	Dirk Hünninger ⁴⁹⁸	cc-by-sa-3.0

493 <http://en.wikibooks.org/wiki/User%3ADirk%20H%FCnniger>

494 <http://en.wikibooks.org/wiki/User%3ADirk%20H%FCnniger>

495 <http://en.wikibooks.org/wiki/User%3ADirk%20H%FCnniger>

496 <http://en.wikibooks.org/wiki/User%3ADirk%20H%FCnniger>

497 <http://en.wikibooks.org/wiki/User%3ADirk%20H%FCnniger>

498 <http://en.wikibooks.org/wiki/User%3ADirk%20H%FCnniger>

15	Dirk Hünninger ⁴⁹⁹	cc-by-sa-3.0
16	Dirk Hünninger ⁵⁰⁰	cc-by-sa-3.0
17	Dirk Hünninger ⁵⁰¹	cc-by-sa-3.0
18	Dirk Hünninger ⁵⁰²	cc-by-sa-3.0
19	Dirk Hünninger ⁵⁰³	cc-by-sa-3.0
20	Dirk Hünninger ⁵⁰⁴	cc-by-sa-3.0

499 <http://en.wikibooks.org/wiki/User%3ADirk%20H%Cnniger>

500 <http://en.wikibooks.org/wiki/User%3ADirk%20H%Cnniger>

501 <http://en.wikibooks.org/wiki/User%3ADirk%20H%Cnniger>

502 <http://en.wikibooks.org/wiki/User%3ADirk%20H%Cnniger>

503 <http://en.wikibooks.org/wiki/User%3ADirk%20H%Cnniger>

504 <http://en.wikibooks.org/wiki/User%3ADirk%20H%Cnniger>

21	Dirk Hünninger ⁵⁰⁵	cc-by-sa-3.0
22	Dirk Hünninger ⁵⁰⁶	cc-by-sa-3.0
23	Dirk Hünninger ⁵⁰⁷	cc-by-sa-3.0
24	Dirk Hünninger ⁵⁰⁸	cc-by-sa-3.0
25	Dirk Hünninger ⁵⁰⁹	cc-by-sa-3.0
26	Dirk Hünninger ⁵¹⁰	cc-by-sa-3.0

505 <http://en.wikibooks.org/wiki/User%3ADirk%20H%FCnniger>

506 <http://en.wikibooks.org/wiki/User%3ADirk%20H%FCnniger>

507 <http://en.wikibooks.org/wiki/User%3ADirk%20H%FCnniger>

508 <http://en.wikibooks.org/wiki/User%3ADirk%20H%FCnniger>

509 <http://en.wikibooks.org/wiki/User%3ADirk%20H%FCnniger>

510 <http://en.wikibooks.org/wiki/User%3ADirk%20H%FCnniger>

27	Dirk Hünninger ⁵¹¹	cc-by-sa-3.0
28	Dirk Hünninger ⁵¹²	cc-by-sa-3.0
29	Dirk Hünninger ⁵¹³	cc-by-sa-3.0
30	Dirk Hünninger ⁵¹⁴	cc-by-sa-3.0
31	Dirk Hünninger ⁵¹⁵	cc-by-sa-3.0
32	Dirk Hünninger ⁵¹⁶	cc-by-sa-3.0

511 <http://en.wikibooks.org/wiki/User%3ADirk%20H%C3nninger>

512 <http://en.wikibooks.org/wiki/User%3ADirk%20H%C3nninger>

513 <http://en.wikibooks.org/wiki/User%3ADirk%20H%C3nninger>

514 <http://en.wikibooks.org/wiki/User%3ADirk%20H%C3nninger>

515 <http://en.wikibooks.org/wiki/User%3ADirk%20H%C3nninger>

516 <http://en.wikibooks.org/wiki/User%3ADirk%20H%C3nninger>

33	Dirk Hünninger ⁵¹⁷		cc-by-sa-3.0
34	Dirk Hünninger ⁵¹⁸		cc-by-sa-3.0
35	Dirk Hünninger ⁵¹⁹		cc-by-sa-3.0
36	Alessio Damato ⁵²⁰		GFDL
37	Dirk Hünninger ⁵²¹		cc-by-sa-3.0
38	Dirk Hünninger ⁵²²		cc-by-sa-3.0

517 <http://en.wikibooks.org/wiki/User%3ADirk%20H%FCnniger>

518 <http://en.wikibooks.org/wiki/User%3ADirk%20H%FCnniger>

519 <http://en.wikibooks.org/wiki/User%3ADirk%20H%FCnniger>

520 <http://en.wikibooks.org/wiki/User%3AAlejo2083>

521 <http://en.wikibooks.org/wiki/User%3ADirk%20H%FCnniger>

522 <http://en.wikibooks.org/wiki/User%3ADirk%20H%FCnniger>

39	Original uploader was Jtwdog ⁵²³ at en.wikibooks ⁵²⁴	GFDL
40	Original uploader was Jtwdog ⁵²⁵ at en.wikibooks ⁵²⁶	GFDL
41	Original uploader was Jtwdog ⁵²⁷ at en.wikibooks ⁵²⁸	GFDL

523 <http://en.wikibooks.org/wiki/%3Awikibooks%3Aen%3AUser%3AJtwdog>

524 <http://en.wikibooks.org>

525 <http://en.wikibooks.org/wiki/%3Awikibooks%3Aen%3AUser%3AJtwdog>

526 <http://en.wikibooks.org>

527 <http://en.wikibooks.org/wiki/%3Awikibooks%3Aen%3AUser%3AJtwdog>

528 <http://en.wikibooks.org>

42	Original uploader was Jtwdog ⁵²⁹ at en.wikibooks ⁵³⁰	GFDL
43	ChrisHodgesUK ⁵³¹	
44	ChrisHodgesUK ⁵³²	
45	Alessio Damato ⁵³³	GFDL
46	Mwtoews ⁵³⁴	GFDL
47	User:Mwtoews ⁵³⁵	GFDL
48	Alessio Damato ⁵³⁶	GFDL
49	Alessio Damato ⁵³⁷	GFDL

529 <http://en.wikibooks.org/wiki/%3Awikibooks%3Aen%3AUser%3AJtwdog>

530 <http://en.wikibooks.org>

531 <http://en.wikibooks.org/wiki/User%3AChrisHodgesUK>

532 <http://en.wikibooks.org/wiki/User%3AChrisHodgesUK>

533 <http://en.wikibooks.org/wiki/User%3AAlejo2083>

534 <http://en.wikibooks.org/wiki/User%3AMwtoews>

535 <http://en.wikibooks.org/wiki/User%3AMwtoews>

536 <http://en.wikibooks.org/wiki/User%3AAlejo2083>

537 <http://en.wikibooks.org/wiki/User%3AAlejo2083>

50	Alessio Damato ⁵³⁸	GFDL
51	ChrisHodgesUK ⁵³⁹	
52	ChrisHodgesUK ⁵⁴⁰	
53	Tomato86 ⁵⁴¹	GFDL
54	Alessio Damato ⁵⁴²	GFDL
55	ChrisHodgesUK ⁵⁴³	
56	Winfree ⁵⁴⁴	PD
57	Tomato86 ⁵⁴⁵	GFDL
58	Neet ⁵⁴⁶	PD

538 <http://en.wikibooks.org/wiki/User%3AAAlejo2083>

539 <http://en.wikibooks.org/wiki/User%3AChrisHodgesUK>

540 <http://en.wikibooks.org/wiki/User%3AChrisHodgesUK>

541 <http://en.wikibooks.org/wiki/User%3ATomato86>

542 <http://en.wikibooks.org/wiki/User%3AAAlejo2083>

543 <http://en.wikibooks.org/wiki/User%3AChrisHodgesUK>

544 <http://en.wikibooks.org/wiki/User%3AWinfree>

545 <http://en.wikibooks.org/wiki/User%3ATomato86>

546 <http://en.wikibooks.org/wiki/User%3ANeet>

59	Neet ⁵⁴⁷	PD
60	Neet ⁵⁴⁸	PD
61	Neet ⁵⁴⁹	PD
62	Neet ⁵⁵⁰	PD
63	Tomato86 ⁵⁵¹	GFDL
64	Tomato86 ⁵⁵²	GFDL
65	Tomato86 ⁵⁵³	GFDL
66	Tomato86 ⁵⁵⁴	GFDL
67	Inductiveload ⁵⁵⁵	PD

⁵⁴⁷ <http://en.wikibooks.org/wiki/User%3ANeet>

⁵⁴⁸ <http://en.wikibooks.org/wiki/User%3ANeet>

⁵⁴⁹ <http://en.wikibooks.org/wiki/User%3ANeet>

⁵⁵⁰ <http://en.wikibooks.org/wiki/User%3ANeet>

⁵⁵¹ <http://en.wikibooks.org/wiki/User%3ATomato86>

⁵⁵² <http://en.wikibooks.org/wiki/User%3ATomato86>

⁵⁵³ <http://en.wikibooks.org/wiki/User%3ATomato86>

⁵⁵⁴ <http://en.wikibooks.org/wiki/User%3ATomato86>

⁵⁵⁵ <http://en.wikibooks.org/wiki/User%3AInductiveload>

68	Tomato86 ⁵⁵⁶	GFDL
69	Tomato86 ⁵⁵⁷	GFDL
70	Tomato86 ⁵⁵⁸	GFDL
71	ChrisHodgesUK ⁵⁵⁹	
72	Tomato86 ⁵⁶⁰	GFDL
73	Alessio Damato ⁵⁶¹	GFDL
74	Alessio Damato ⁵⁶² 13:31, 12 January 2007 (UTC)	GFDL
75	Alessio Damato ⁵⁶³	GFDL
76	Nemti ⁵⁶⁴	PD

⁵⁵⁶ <http://en.wikibooks.org/wiki/User%3ATomato86>

⁵⁵⁷ <http://en.wikibooks.org/wiki/User%3ATomato86>

⁵⁵⁸ <http://en.wikibooks.org/wiki/User%3ATomato86>

⁵⁵⁹ <http://en.wikibooks.org/wiki/User%3AChrisHodgesUK>

⁵⁶⁰ <http://en.wikibooks.org/wiki/User%3ATomato86>

⁵⁶¹ <http://en.wikibooks.org/wiki/User%3AAlejo2083>

⁵⁶² <http://en.wikibooks.org/wiki/User%3AAlejo2083>

⁵⁶³ <http://en.wikibooks.org/wiki/User%3AAlejo2083>

⁵⁶⁴ <http://en.wikibooks.org/wiki/User%3ANemti>

77	MyName (Gkc ⁵⁶⁵ (talk ⁵⁶⁶))	GFDL
78	LaTeX	GPL
79	jeg	
80	Hankjones ⁵⁶⁷	GFDL
81	Hankjones ⁵⁶⁸	cc-by-sa-3.0
82	Hankjones ⁵⁶⁹	cc-by-sa-3.0
83	jeg	
84	Hankjones ⁵⁷⁰	GFDL

⁵⁶⁵ <http://en.wikibooks.org/wiki/User%3AGkc>

⁵⁶⁶ <http://en.wikibooks.org/wiki/User%20talk%3AGkc>

⁵⁶⁷ <http://en.wikibooks.org/wiki/User%3AHankjones>

⁵⁶⁸ <http://en.wikibooks.org/wiki/User%3AHankjones>

⁵⁶⁹ <http://en.wikibooks.org/wiki/User%3AHankjones>

⁵⁷⁰ <http://en.wikibooks.org/wiki/User%3AHankjones>

85	Hankjones ⁵⁷¹	cc-by-sa-3.0
86	Philip John Gorinski ⁵⁷²	
87	Philip John Gorinski ⁵⁷³	
88	Philip John Gorinski ⁵⁷⁴	PD
89	Matěj Korvas	PD
90	Matej.korvas ⁵⁷⁵	
91	Matej.korvas ⁵⁷⁶	
92	Daniele Pighin	GFDL
93	Olesh ⁵⁷⁷	cc-by-sa-3.0

⁵⁷¹ <http://en.wikibooks.org/wiki/User%3AHankjones>

⁵⁷² <http://en.wikibooks.org/wiki/User%3APhilJohnG>

⁵⁷³ <http://en.wikibooks.org/wiki/User%3APhilJohnG>

⁵⁷⁴ <http://en.wikibooks.org/wiki/User%3APhilJohnG>

⁵⁷⁵ <http://en.wikibooks.org/wiki/User%3AMatej.korvas>

⁵⁷⁶ <http://en.wikibooks.org/wiki/User%3AMatej.korvas>

⁵⁷⁷ <http://en.wikibooks.org/wiki/User%3AOlesh>

94	Hankjones ⁵⁷⁸		cc-by-sa-3.0
95	Hankjones ⁵⁷⁹		cc-by-sa-3.0
96	ChrisHodgesUK ⁵⁸⁰		
97	Hankjones ⁵⁸¹		cc-by-sa-3.0
98	Hankjones ⁵⁸²		cc-by-sa-3.0
99	Hankjones ⁵⁸³		cc-by-sa-3.0

578 <http://en.wikibooks.org/wiki/User%3AHankjones>

579 <http://en.wikibooks.org/wiki/User%3AHankjones>

580 <http://en.wikibooks.org/wiki/User%3AChrisHodgesUK>

581 <http://en.wikibooks.org/wiki/User%3AHankjones>

582 <http://en.wikibooks.org/wiki/User%3AHankjones>

583 <http://en.wikibooks.org/wiki/User%3AHankjones>

100	Hankjones ⁵⁸⁴	cc-by-sa-3.0
101	Derbeth ⁵⁸⁵	GFDL
102		PD
103	gms	GFDL
104	Israel Buitron ⁵⁸⁶	GFDL
105	Alessio Damato ⁵⁸⁷	GFDL
106	Alessio Damato	GFDL
107	Alessio Damato ⁵⁸⁸	GFDL
108	Alessio Damato ⁵⁸⁹	GFDL
109	Alessio Damato ⁵⁹⁰	GFDL

584 <http://en.wikibooks.org/wiki/User%3AHankjones>

585 <http://en.wikibooks.org/wiki/User%3ADerbeth>

586 <http://en.wikibooks.org/wiki/User%3ANeoriddle>

587 <http://en.wikibooks.org/wiki/User%3AAAlejo2083>

588 <http://en.wikibooks.org/wiki/User%3AAAlejo2083>

589 <http://en.wikibooks.org/wiki/User%3AAAlejo2083>

590 <http://en.wikibooks.org/wiki/User%3AAAlejo2083>

110	Original uploader was Arnehe ⁵⁹¹ at en.wikibooks ⁵⁹²	
111	Original uploader was Arnehe ⁵⁹³ at en.wikibooks ⁵⁹⁴	
112	Original uploader was Arnehe ⁵⁹⁵ at en.wikibooks ⁵⁹⁶	

591 <http://en.wikibooks.org/wiki/%3Awikibooks%3Aen%3AUser%3AArnehe>

592 <http://en.wikibooks.org>

593 <http://en.wikibooks.org/wiki/%3Awikibooks%3Aen%3AUser%3AArnehe>

594 <http://en.wikibooks.org>

595 <http://en.wikibooks.org/wiki/%3Awikibooks%3Aen%3AUser%3AArnehe>

596 <http://en.wikibooks.org>

113	Original uploader was Arnehe ⁵⁹⁷ at en.wikibooks ⁵⁹⁸	
114	Original uploader was Arnehe ⁵⁹⁹ at en.wikibooks ⁶⁰⁰	
115	Alessio Damato ⁶⁰¹	GFDL
116		GFDL
117		GFDL
118		GFDL
119	Tomato86 ⁶⁰²	GFDL
120	Thenub314 ⁶⁰³	GFDL

⁵⁹⁷ <http://en.wikibooks.org/wiki/%3Aawikibooks%3Aen%3AUser%3AArnehe>

⁵⁹⁸ <http://en.wikibooks.org>

⁵⁹⁹ <http://en.wikibooks.org/wiki/%3Aawikibooks%3Aen%3AUser%3AArnehe>

⁶⁰⁰ <http://en.wikibooks.org>

⁶⁰¹ <http://en.wikibooks.org/wiki/User%3AAlejo2083>

⁶⁰² <http://en.wikibooks.org/wiki/User%3ATomato86>

⁶⁰³ <http://en.wikibooks.org/wiki/User%3Athenub314>

121	Tomato86 ⁶⁰⁴		GFDL
122	Derbeth ⁶⁰⁵		PD
123	Tobias Oetiker		GFDL
124			GFDL
125	Jtwdog ⁶⁰⁶ at en.wikibooks ⁶⁰⁷		GFDL
126			
127	Tobias Oetiker		GFDL
128			GFDL
129	Tobias Oetiker		GFDL
130	Tobias Oetiker		GFDL
131			GFDL
132	Andrew Roberts		GFDL

⁶⁰⁴ <http://en.wikibooks.org/wiki/User%3ATomato86>

⁶⁰⁵ <http://en.wikibooks.org/wiki/User%3ADerbeth>

⁶⁰⁶ <http://en.wikibooks.org/wiki/%3Awikibooks%3Aen%3AUser%3AJtwdog>

⁶⁰⁷ <http://en.wikibooks.org>

49. Licenses

49.1. GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc.
<<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program—to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this

way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure

that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the

area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary.

To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow. TERMS AND CONDITIONS 0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licenseses" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for

infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a promi-

ment item in the list meets this criterion. 1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essen-

subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work. 2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions

so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary. 3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obliga-

tions under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures. 4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with sec-

tion 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee. 5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

* a) The work must carry prominent notices stating that you modified it, and giving a relevant date. * b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices". * c) You must license the entire work, as a whole, under this License to anyone who

comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it. * d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individ-

ual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate. 6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Copying the Source under the terms of this License, in one of these ways:

* a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange. * b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either

ferred with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Conveying Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Ac-

tion into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or inter-

ent server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements. * e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorpora-

(1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge. * c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accordance with subsection 6b. * d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a differ-

cess to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those

permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

* a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or * b) Requiring preservation of specified legal notices

or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or * c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or * d) Limiting the use for publicity purposes of names of licensors or authors of the material; or * e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or * f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this

License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way. 8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise

to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation from this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An “entity transaction” is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party’s predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For ex-

ample, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it. 11. Patents.

A “contributor” is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor’s “contributor version”.

A contributor’s “essential patent claims” are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For

purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not avail-

able for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, prop-

agate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you enti-

tered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law. 12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Pro-

gram. 13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such. 14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If

the Program specifies that a certain numbered version of the GNU General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version. 15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION. 16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER,

OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. 17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an abso-

lute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.> Copyright (C) <year>
<name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with

this program. If not, see <<http://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C)
<year> <name of author> This
program comes with ABSO-
LUTELY NO WARRANTY;
for details type 'show w'. This
is free software, and you are
welcome to redistribute it under
certain conditions; type 'show
c' for details.
```

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a

GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <<http://www.gnu.org/licenses/>>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <<http://www.gnu.org/philosophy/not-lgpl.html>>.

49.2. GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002,
2007, 2008 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. 0.
PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which

is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference. 1. **APPLICABILITY AND DEFINITIONS**

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or

work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philo-

sophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general pub-

lic, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG,

XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either

is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License. 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this Li-

cence, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies. 3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover,

and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using pub-

lication filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

* A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission. * B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement. * C. State on the Title page the name of the publisher of the Modified Version, as the publisher. * D. Preserve all the copyright notices of the Document. * E. Add

an appropriate copyright notice for your modifications adjacent to the other copyright notices. * F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License.

in the form shown in the Addendum below. * G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice. * H. Include an unaltered copy of this License. * I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence. * J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the

network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

* K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein. * L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles. * M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version. * N. Do not retitile any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section. * O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made

by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version. 5.

COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements". 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the in-

dividual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document. 7.

8. TRANSLATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this

License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices

in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title. 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) pro-

visionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it. 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for

the Document. 11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copy/left versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing. ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (C) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any

later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with ... Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains non-trivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

49.3. GNU Lesser General Public License

GNU LESSER GENERAL
PUBLIC LICENSE

Version 3, 29 June 2007

Copyright © 2007 Free
Software Foundation, Inc.
<<http://fsf.org/>>

Everyone is permitted to copy
and distribute verbatim copies
of this license document, but
changing it is not allowed.

This version of the GNU Lesser
General Public License incorpo-
rates the terms and conditions of
version 3 of the GNU General
Public License, supplemented
by the additional permissions
listed below. 0. Additional Defi-
nitions.

As used herein, “this License”
refers to version 3 of the GNU
Lesser General Public License,
and the “GNU GPL” refers to
version 3 of the GNU General
Public License.

“The Library” refers to a cov-
ered work governed by this Li-
cense, other than an Application
or a Combined Work as defined
below.

An “Application” is any work
that makes use of an interface
provided by the Library, but
which is not otherwise based on
the Library. Defining a subclass
of a class defined by the Library
is deemed a mode of using an in-
terface provided by the Library.

A “Combined Work” is a work
produced by combining or link-
ing an Application with the Li-
brary. The particular version of
the Library with which the Com-
bined Work was made is also
called the “Linked Version”.

The “Minimal Correspond-
ing Source” for a Combined
Work means the Corresponding
Source for the Combined Work,
excluding any source code for
portions of the Combined Work
that, considered in isolation, are

based on the Application, and
not on the Linked Version.

The “Corresponding Applica-
tion Code” for a Combined
Work means the object code
and/or source code for the Ap-
plication, including any data
and utility programs needed
for reproducing the Combined
Work from the Application, but
excluding the System Libraries
of the Combined Work. 1. Ex-
ception to Section 3 of the GNU
GPL.

You may convey a covered work
under sections 3 and 4 of this
License without being bound by
section 3 of the GNU GPL. 2.
Conveying Modified Versions.

If you modify a copy of the
Library, and, in your modifica-
tions, a facility refers to a func-
tion or data to be supplied by an
Application that uses the facil-
ity (other than as an argument
passed when the facility is in-
voked), then you may convey a
copy of the modified version:

* a) under this License, provided
that you make a good faith ef-
fort to ensure that, in the event
an Application does not supply
the function or data, the facil-
ity still operates, and performs
whatever part of its purpose re-
mains meaningful, or * b) un-
der the GNU GPL, with none
of the additional permissions of
this License applicable to that
copy.

3. Object Code Incorporating
Material from Library Header
Files.

The object code form of an Ap-
plication may incorporate mate-
rial from a header file that is
part of the Library. You may
convey such object code under
terms of your choice, provided
that, if the incorporated material
is not limited to numerical pa-
rameters, data structure layouts
and accessors, or small macros,
inline functions and templates
(ten or fewer lines in length),
you do both of the following:

* a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License. * b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

* a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License. * b) Accompany the Combined Work with a copy of the GNU GPL and this license document. * c) For a Combined Work that displays copyright notices during execution, include the copyright notice for

the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document. * d) Do one of the following: o 0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source. o 1)

Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version. * e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the

GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

* a) Accompany the combined library with a copy of the same

work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License. * b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later

version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public Li-

cence, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's

public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.