

GHCi: Getting started (1A)

Copyright (c) 2016 - 2017 Young W. Lim.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 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".

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice.

Based on

[Haskell in 5 steps](https://wiki.haskell.org/Haskell_in_5_steps)

https://wiki.haskell.org/Haskell_in_5_steps

Interpreter GHCi

```
young@MNTSys-BB1 ~ $ ghci
```

```
GHCi, version 7.10.3: http://www.haskell.org/ghc/ :? for help
```

```
Prelude> "hello, world!"
```

```
"hello, world!"
```

```
Prelude> putStrLn "hello, world!"
```

```
hello, world!
```

https://wiki.haskell.org/Learn_Haskell_in_10_minutes

Function

```
Prelude> let fac n = if n == 0 then 1 else n * fac (n-1)
```

```
Prelude> fac 5
```

```
120
```

```
Prelude> fac 2
```

```
2
```

```
Prelude> fac 3
```

```
6
```

```
Prelude> fac 4
```

```
24
```

```
Prelude>
```

https://wiki.haskell.org/Learn_Haskell_in_10_minutes

Compiler GHC

```
young@MNTSys-BB1 ~ $ ghc -o hello hello.hs
[1 of 1] Compiling Main      ( hello.hs, hello.o )
Linking hello ...
young@MNTSys-BB1 ~ $ ./hello
hello, world!
```

```
young@MNTSys-BB1 ~ $ cat hello.hs
main = putStrLn "hello, world!"
```

https://wiki.haskell.org/Learn_Haskell_in_10_minutes

Layout

t.hs

```
main = do putStrLn "Type an integer : ?"
         x <- readLn
         if even x
           then putStrLn "even number"
           else putStrLn "odd number"
```

the first non-space character after **do**.

every line that starts in the same column as that p is in the **do** block

If you indent more, it is the nested block in **do**

If you indent less, it is an end of the **do** block.

ghc t.hs

Ghc -o run t.hs

./t

./t

Multi-line in GHCi

ghci multi-line

Prelude> **{**

Prelude| **main = do** { putStrLn "Type an integer: "; x<-readLn;

Prelude| **if** even x **then** putStrLn "even" **else** putStrLn "odd"; }

Prelude| **}**

https://wiki.haskell.org/Learn_Haskell_in_10_minutes

Types

Int	an integer with at least <u>30 bits</u> of precision.
Integer	an integer with <u>unlimited</u> precision.
Float	a <u>single precision</u> floating point number.
Double	a <u>double precision</u> floating point number.
Rational	a <u>fraction</u> type, with no rounding error.

Types and Class Types start with capital letters

Variables start with lower case letters

Declaring a type :: type

Asking which type :t something

https://wiki.haskell.org/Learn_Haskell_in_10_minutes

Types

```
Prelude> 3 :: Int
3
Prelude> 3 :: Float
3.0
Prelude> 4 :: Double
4.0
Prelude> 2 :: Integer
2
Prelude> :t 3
3 :: Num a => a
Prelude> :t 2
2 :: Num a => a
Prelude> :t True
True :: Bool
Prelude> :t 'A'
'A' :: Char
```

https://wiki.haskell.org/Learn_Haskell_in_10_minutes

References

- [1] <ftp://ftp.geoinfo.tuwien.ac.at/navratil/HaskellTutorial.pdf>
- [2] <https://www.umiacs.umd.edu/~hal/docs/daume02yaht.pdf>