

Un livre de Wikilivres.

À la découverte d'Unicode

Une version à jour et éditable de ce livre est disponible sur Wikilivres, une bibliothèque de livres pédagogiques, à l'URL :
http://fr.wikibooks.org/wiki/%C3%80_la_d%C3%A9couverte_d%27Unicode

Vous avez la permission de copier, distribuer et/ou modifier ce document selon les termes de la Licence de documentation libre GNU, version 1.2 ou plus récente publiée par la Free Software Foundation ; sans sections inaltérables, sans texte de première page de couverture et sans Texte de dernière page de couverture. Une copie de cette licence est incluse dans l'annexe nommée « Licence de documentation libre GNU ».

Avant Unicode : télégraphie, bidouilles et héritages

Unicode étant l'unification dans un seul et même ensemble de plusieurs jeux de caractères, il est difficile de parler d'Unicode sans parler des jeux de caractères qui lui ont précédé.

L'écriture

La **Préhistoire** est généralement définie comme la période comprise entre l'apparition de l'Humanité et l'apparition des premiers documents écrits, même si cette définition n'est pas sans poser des problèmes. La version anglophone de wikipedia donne l'histoire de l'écriture.

L'écriture matérielle est assez ancienne et forme en quelques sorte une préhistoire de l'écriture dématérialisée. Cette écriture dématérialisée est quelque peu plus moderne et s'est appuyée sur différentes technologies de communication avant l'arrivée du télégraphe électrique.

Télégraphie électrique et code Baudot

Le **code Baudot** est dans l'histoire un des premiers codes binaires utilisé grâce à une machine. Il est aussi appelé code télégraphique Alphabet International (AI) n°1 ou Alphabet International (AI) n°2 ou code CCITT n°2.

C'est un code binaire, c'est-à-dire que chaque caractère que l'on souhaite est codé par une combinaison de 0

et de 1. Le code ne prévoit que 5 bits pour coder chaque caractère, donc il n'existe que $2^5 = 32$ combinaisons. Or si on désire coder les lettres et les chiffres, il n'y a pas assez de combinaisons. C'est pourquoi le code Baudot contient deux jeux de caractères appelés *Lettres* et *Chiffres*. En fait, l'ensemble Chiffres contient aussi d'autres symboles (ponctuation, &, #...). Deux caractères Inversion Lettres (code 31) et Inversion Chiffres (code 27) permettent de commuter suivant que l'on en est mode lettre ou en mode chiffre.

Le premier code de ce type a été développé par Émile Baudot en 1874 : il s'agissait de l'Alphabet International n°1. Il n'est plus utilisé. Les caractères étaient composés à l'aide d'un clavier à cinq touches, où chaque touche correspondait à l'un des cinq bits de chaque caractère.

Cette technologie ancienne et limitée a dès le début posé des problèmes de standardisation.

Émile Baudot avait par exemple trouvé judicieux d'intégrer la lettre É (e accentué aigu) dans son jeu de caractère. La présence ou l'absence de l'accent changeant du tout au tout le sens d'un mot dans la langue française. Ce caractère a été supprimé des versions anglaises. Rendant déjà les deux versions incompatibles.

Multipléts élargis et caractères appropriés

Les différents développements économiques scientifiques et militaires ont conduit à un élargissement des multipléts, passant de cinq bits initialement à huit bits (dont l'octet du grec *ὀκτώ*, huit) durant la seconde moitié du vingtième siècle.

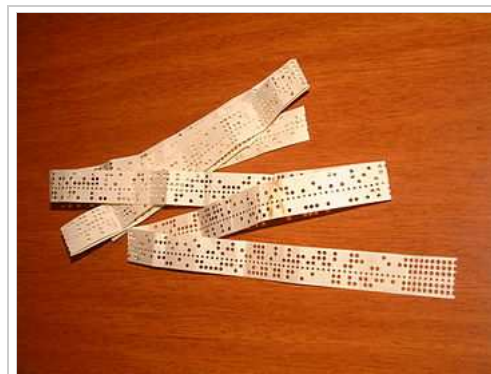
Les efforts de normalisations ont commencé tôt avec l'Wikipedia:ISO/CEI 646. Le nombre très limité des caractères disponibles dans un encodage sept bits a tout de suite posé problème, conduisant à l'invention de digraphes et de trigraphes dans le langage C pour représenter avec deux caractères les caractères indispensables au langage C lorsque ces caractères n'étaient pas disponibles.

Cet inconvénient fut pallié en adoptant à un niveau mondial la variante des États-Unis de la famille de jeux de caractères de l'ISO-646. Si ce sous-ensemble fut relativement standardisé dans son adoption, les différents acteurs ont redoublé d'inventivité et de créativité pour apporter aux utilisateurs de ces systèmes les caractères qu'ils souhaitaient utiliser.

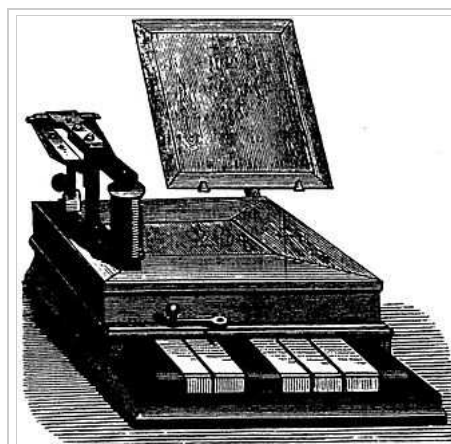
Ceci a conduit à l'utilisation d'une multitude de caractères représentant tout aussi bien les caractères traditionnels que l'on avait l'habitude de voir écrits et ou imprimés que des caractères nouvellement inventés ou recréés (comme les caractères € (euro), | ou @). Cela a également conduit à ce que ces caractères ne soient pas transmis par la même séquence ou convention numérique en fonction du standard utilisé.

Cela pouvait varier d'un pays à l'autre, mais aussi à l'intérieur d'un même pays. En France, par exemple, des codages différents sont utilisés pour le minitel, pour le DOS, pour Windows, pour Unix, et pour les SMS, sans que cela soit limitatif.

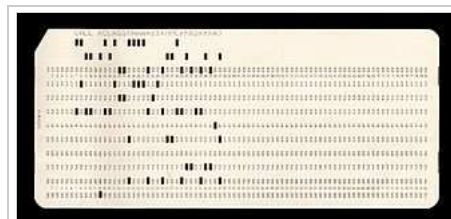
Le nombre de systèmes de codage de caractères ainsi inventés est de plusieurs centaines voir de plusieurs



bande de papier avec des trous représentant les "Code Baudot"



Le clavier et ses cinq touches.



Carte perforée à 80 colonnes, sur laquelle est codé le texte de programmation «CALL RCLASS (AAA,21,NNC,PX3,PX4)»

milliers, le tout s'étant effectué en raison de l'indépendance de chaque acteur dans la plus grande anarchie.

L'ASCII était alors un sous-ensemble commun à certains d'entre eux, offrant une interopérabilité limitée. A ce sujet, on peut lire le wikilivre *Les_ASCII_de_0_à_127*.

Gestion de la diversité et bidouilles

Pour gérer cette grande diversité, différentes techniques ont été inventées.

Parmi celles-ci on peut citer l'ISO-2022 et la notion de logiciel traitant les chaînes d'octets indépendamment des caractères représentés.

Dès 1987 deux idées (issues du Centre de recherche de Xerox à Palo Alto (PARC) et d'Apple) convergèrent:

- la conception d'un codage à largeur fixe mais suffisamment grand pour éviter les difficultés des traitements sur codages multioctets
- l'« unification han » pour représenter dans un seul jeu de caractères des textes chinois, japonais ou coréens (CJC). En 1987 et en 1988, Apple et Xerox s'impliquèrent sur ces sujets^[1].

Par la suite, le développement de l'industrie logicielle s'est orientée sur l'Unicode.

Références

1. http://www.cairn.info/article.php?ID_ARTICLE=DN_063_0329&DocId=43462&Index=%2Fcairn2Idx%2Fcairn&TypeID=226&BAL=anMlj%2FYMELumM&HitCount=4&hits=29ee+2276+a2c+76c+0&fileext=html

- Jacques André « Caractères, codage et normalization », Document numérique 3/2002 (Vol. 6), p. 13-49.
URL : www.cairn.info/revue-document-numerique-2002-3-page-13.htm.
DOI : 10.3166/dn.6.3-4.13-49.

Unicode : institutions et versions

Si l'on parle couramment d'unicode pour désigner le format UTF-8 ou le format UTF-16, Unicode est aussi tout à la fois un consortium, un répertoire de caractères et des considérations pour la mise en œuvre de ce répertoire. Ce répertoire est également semblable à l'ISO 10646.

Le site web Unicode présente tout cela de manière très technique et en langue anglaise. Ce livre se veut plus pratique dans la mesure où il s'adresse à un public francophone. Les anglophones pourront trouver des compléments d'information dans la spécification officielle disponible sur ce site.

Les institutions

Organisation internationale de normalisation (ISO)

L' *Organisation internationale de normalisation* (*International Organization for Standardization*), ou **ISO** est un organisme de normalisation international composé de représentants d'organisations nationales de normalisation de 164 pays^[1]. Cette organisation créée en 1947 a pour but de produire des normes

internationales dans les domaines industriels et commerciaux appelées normes ISO. Elles sont utiles aux organisations industrielles et économiques de tout type, aux gouvernements, aux instances de réglementation, aux dirigeants de l'économie, aux professionnels de l'évaluation de la conformité, aux fournisseurs et acheteurs de produits et de services, dans les secteurs tant public que privé et, en fin de compte, elles servent les intérêts du public en général lorsque celui-ci agit en qualité de consommateur et utilisateur.

Le secrétariat central de l'ISO est situé à Genève, en Suisse. Il assure aux membres de l'ISO le soutien administratif et technique, coordonne le programme décentralisé d'élaboration des normes et procède à leur publication.

L'ISO est décrite sur wikipedia [wikipedia:fr:Organisation internationale de normalisation](https://fr.wikipedia.org/wiki/Organisation_internationale_de_normalisation).

Elle est à l'initiative de la norme **ISO/CEI 10646**, intitulée **Technologies de l'information — Jeu universel de caractères codés (JUC)**,

Commission électrotechnique internationale (CEI)

La *Commission électrotechnique internationale (CEI)* ou '**International Electrotechnical Commission (IEC)**' en anglais, est l'organisation internationale de normalisation chargée des domaines de l'électricité, de l'électronique et des techniques connexes. Elle est complémentaire de l'Organisation internationale de normalisation (ISO), qui est chargée des autres domaines.

La CEI est composée de représentants de différents organismes de normalisation nationaux. La CEI a été créée en 1906 et compte actuellement 69 pays participants. Les normes CEI sont reconnues dans plus de 100 pays.

Originellement située à Londres, la Commission a rejoint ses quartiers généraux actuels de Genève en 1948. La CEI dispose de trois centres régionaux à Singapour, São Paulo et Worcester (Massachusetts) ^[2].

La CEI a été l'instrument du développement et de la distribution de normes d'unités de mesure, notamment le gauss, le hertz, et le weber. Elle contribua également à proposer un ensemble de références, le système Giorgi, qui finalement fut intégré au système international d'unités (SI) dont l'ISO est responsable.

En savoir plus sur wikipedia: [wikipedia:fr:Commission électrotechnique internationale](https://fr.wikipedia.org/wiki/Commission_électrotechnique_internationale)

Elle est à l'initiative de la norme **ISO/CEI 10646**, intitulée **Technologies de l'information — Jeu universel de caractères codés (JUC)**.

Le Consortium Unicode

Le *Consortium Unicode* est une organisation privée sans but lucratif qui coordonne le développement du standard Unicode. Elle a pour objectif ambitieux de succéder à terme aux codages de caractères pré-existants.

Elle travaille aujourd'hui en coordination, via son propre « comité technique Unicode » (UTC), aux travaux de normalisation de l'Organisation internationale de normalisation (ISO) dans ce domaine pour la norme ISO/CEI 10646 avec laquelle le standard Unicode a fusionné son répertoire universel de caractères codés (après avoir pendant un temps limité développé un standard Unicode 1.0 incompatible et désormais obsolète).

À côté de son objectif initial, l'organisation coordonne divers travaux de standardisation techniques, dont le développement de données de régionalisation de logiciels, comme le projet [Wikipedia:Common Locale Data Repository \(CLDR\)](https://fr.wikipedia.org/wiki/Common_Locale_Data_Repository), avec son « propre comité technique CLDR ».

Le consortium est aussi le bureau d'enregistrement officiel de quelques normes ISO relatives à ce domaine, telle que la norme ISO/CEI 15924.

Le consortium Unicode est décrit plus en détail sur son site web ainsi que sur le site Wikipedia.

- Le Consortium Unicode (<http://www.unicode.org>) [[archive](#)]

Norme et standards

Unicode est à la fois normalisé par les organismes publics de normalisation sous le nom d'ISO 10646, qui sont des normes payantes, et standardisé par le consortium éponyme qui diffuse un standard gratuit.

Norme ISO/CEI 10646

La norme **ISO/CEI 10646**, intitulée **Technologies de l'information — Jeu universel de caractères codés (JUC)**, tente de définir un système de codage universel pour tous les systèmes d'écriture. Cette norme est le fondement du standard Unicode.

La norme internationale **ISO/CEI 10646** définit le **jeu universel de caractères** (JUC), (en anglais **Universal Character Set** (UCS)) comme un jeu de caractères abstrait. Chaque caractère abstrait est identifié par un nom unique (un nom unique en anglais et un nom unique en français) et associé à un nombre entier naturel positif appelé son **point de code** (ou *position de code*).

Environ 110 000 caractères (symboles, lettres, nombres, idéogrammes, logogrammes) issus de langues, systèmes d'écriture, traditions du monde entier sont recensés dans le JUC. De nouveaux caractères provenant d'écritures plus rares ou plus anciennes, ou encore de systèmes nouveaux, sont fréquemment ajoutés ou mis à jour dans le JUC.

Depuis 1991, le Consortium Unicode collabore avec l'ISO pour développer le *Standard Unicode* (« Unicode ») et la norme **ISO/CEI 10646**. Les répertoires, noms de caractères, et points de code de la Version 2.0 d'Unicode correspondent exactement à ceux de la norme **ISO/CEI 10646-1:1993**^[3] avec ses sept premiers amendements publiés. Chaque publication d'une nouvelle version d'Unicode donne ensuite lieu à une mise à jour de la norme, c'est-à-dire l'adjonction de nouveaux caractères et la mise à jour de ceux déjà présents. Par exemple, la publication d'Unicode 3.0 en février 2000 correspond à la norme **ISO/CEI 10646-1:2000**. Voir la section Relation avec Unicode pour plus de détails.

Le JUC comprend plus d'1,1 million de points de code, mais seuls les 65 536 premiers (le **Plan Multilingue de Base**, ou PMB) ont été vulgarisés avant 2000. Cette situation commença à changer quand la Chine populaire (RPC) légiféra en 2000 que les systèmes informatiques vendus sur son territoire devaient supporter le GB 18030, ce qui nécessitait que les systèmes informatiques mis à la vente dans la RPC utilisent des caractères au-delà du PMB.

Le système laisse délibérément beaucoup de points de code non assignés à des caractères, même dans le PMB. Cela permet de ménager des extensions futures ou de minimiser les conflits avec d'autres codages.

Standard Unicode et versions

Le standard est édité en langue anglaise par le consortium Unicode. Chaque jeu d'évolutions donne lieu à une nouvelle version du standard. Ces standards sont librement et gratuitement consultables sur le site Unicode, à la page suivante: [1] (<http://www.unicode.org/standard/standard.html>).

Le travail sur Unicode est parallèle et synchronisé avec celui sur la norme ISO/CEI 10646 dont les buts sont les mêmes. L'ISO/CEI 10646, une norme internationale publiée en français et en anglais, ne précise

cependant ni les règles de composition de caractères, ni les propriétés sémantiques des caractères.

Unicode aborde cependant la problématique de la casse, du classement alphabétique, et de la combinaison d'accents et de caractères. Depuis la version 1.1 d'Unicode et dans toutes les versions suivantes, les caractères ont les mêmes identifiants que ceux de la norme ISO/CEI 10646 : les répertoires sont maintenus parallèlement, à l'identique lors de leur normalisation définitive, les deux normes étant mises à jour presque simultanément. Les deux normes Unicode (depuis la version 1.1) et ISO/CEI 10646 assurent une compatibilité ascendante totale : tout texte conforme à une version antérieure doit rester conforme dans les versions ultérieures.

Ainsi les caractères de la version 3.0 d'Unicode sont ceux de la norme ISO/CEI 10646:2000. La version 3.2 d'Unicode classait 95 221 caractères, symboles et directives.

La version 4.1 d'Unicode, mise à jour en novembre 2005, contient :

- 137 468 caractères à usage privé (assignés dans toutes les versions d'Unicode et suffisants pour tous les usages) ;
- plus de 97 755 lettres ou syllabes, chiffres ou nombres, symboles divers, signes diacritiques et signes de ponctuation, avec parmi eux :
 - plus de 70 207 caractères idéographiques, et
 - parmi eux, 11 172 syllabes hangûles précomposées ; ainsi que
- 8 258 positions de codes réservées de façon permanente, interdites pour le codage de texte (assignées dans toutes les versions d'Unicode) ; et
- plusieurs centaines de caractères de contrôle ou modificateurs spéciaux ;

soit un total de près de 245 000 positions de codes assignées dans un espace pouvant contenir 1 114 112 codes différents.

Quelques problèmes semblent cependant exister, pour le codage des caractères chinois, à cause de l'unification des jeux idéographiques utilisés dans différentes langues, avec une calligraphie légèrement différente et parfois signifiante, mais ils sont en cours de résolution par Unicode qui a défini des sélecteurs de variantes et ouvert un registre de séquences normalisées qui les utilise.

La version 5.0 a été publiée en juillet 2006, la version 5.2 en octobre 2009, la version 6.0 en février 2011 et la version 6.1 le 31 janvier 2012.

Les couches d'Unicode

Unicode est défini suivant un modèle en couches (Note technique Unicode #17^[4]). Les autres normes ne faisaient typiquement pas de distinction entre le jeu de caractères et la représentation physique. Les couches sont ici présentées en partant de la plus haute (la plus éloignée de la machine).

Références

1. Au 31 décembre 2006
2. *Centres régionaux de la CEI* (http://www.iec.ch/about/rc/rc_entry-f.htm) [[archive](#)], sur [iec.ch](http://www.iec.ch)
3. ISO/CEI 10646-1:1993: *Technologies de l'information — Jeu universel de caractères codés à plusieurs octets — Partie 1: Architecture et table multilingue*
4. Unicode Technical Report #17: Unicode Character Encoding Model (<http://www.unicode.org/reports/tr17/>) [[archive](#)]

La notion de caractère

Si de prime abord, la **notion de caractère** peut sembler évidente et triviale, vouloir la considérer dans son universalité l'est moins. On s'en rend compte en se questionnant sur l'écriture.

l'écriture

L'écriture existe depuis environ 5 000 ans. Elle est apparue pour la première fois en Égypte (environ en 3300 av. J.-C.) et en Mésopotamie (environ en 3500 av. J.-C.). Pour la première fois, en 2400 av. J.-C., les Égyptiens écrivirent sur des papyrus. Le premier véritable alphabet naquit en 1400 av. J.-C. Au III^e siècle av. J.-C., il y eut l'alphabet latin de 19 lettres, l'ancêtre de l'alphabet européen, qui fit son apparition.

La tradition veut que les caractères chinois aient été inventés par chinois * (c. 2750 av. JC). Ses compositions étaient fondées sur l'observation de la nature. Une autre tradition fait remonter l'invention des caractères à chinois *, le légendaire premier empereur^[1].

Les sinogrammes connurent plusieurs formes. Leur forme actuelle, Kǎishū, remonte à la dynastie des Han.

Un *texte* est une succession de caractères organisée selon une langue.

Le dessin du caractère

En typographie, le *caractère* est, la petite pièce fondue, généralement en plomb, dont l'empreinte forme la lettre ou le signe qui permet d'imprimer des textes sur du papier. Un caractère alphabétique peut s'appeler simplement « lettre^[2] ».

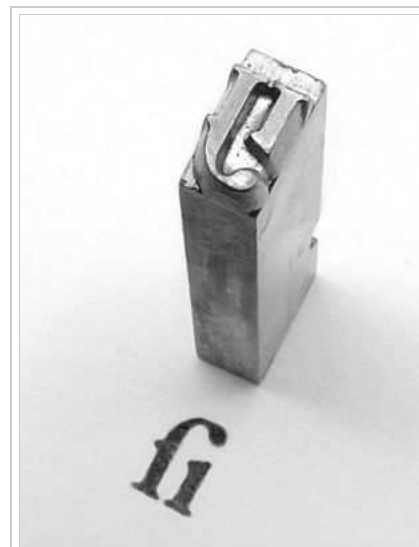
La typographie avec ses caractères mobiles et réutilisables débute avec l'invention de l'imprimerie au VII^e siècle et s'achève avec l'avènement de la photocomposition dans les années 1970^[3].

Un jeu de caractères typographiques cohérents porte aussi le nom de fonte de caractères ou police de caractères.

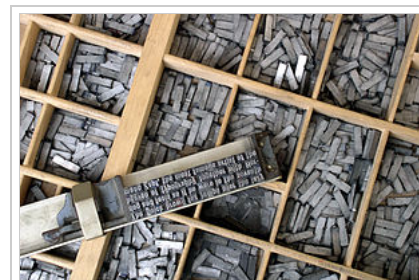
Ceci répond à la question du matérialisme du caractère, à son dessin, mais non à son abstraction.

Le caractère abstrait, informatique

En informatique, le *caractère* est à la fois un type de donnée et une notion abstraite. Comme en typographie, un caractère informatique peut représenter une lettre minuscule, une lettre majuscule, un chiffre ou un signe de ponctuation, mais aussi une espace typographique, une tabulation, un retour à la ligne et quelques autres opérations spéciales (sonnerie, effacement, etc.) (notion de Caractère de contrôle).



Caractère de la police Garamond en 12 points formant la ligature du s long et du i



Caractères en plomb dans une casse avec le composteur posé sur le dessus.



Caractère typographique Didot.

En informatique, la notion de caractère est une notion qui dans le principe associe à un graphème imprimable un numéro, de manière à dissocier la représentation physique du caractère de sa signification.

Questions de définition

où l'on parle de ligature, de diacritiques, de casse, et d'ordre alphabétique.

Si à la lecture de textes latins et anglais la notion de caractères peut sembler simple. Lorsque l'on pose les notions de ligatures et de diacritiques, la question se corse quelque peu.

Si les lettres e et o sont des caractères, la ligature œ formée par le positionnement de l'e dans l'o constitue-elle un caractère, ou bien deux ? De même la présence d'un accent sur la lettre e fait-elle de cette dernière un caractère différent, l'accent lui-même est-il un caractères ?

Toute lettre dispose-elle d'une graphie capitale et d'un graphie minuscule ? Si oui, cette graphie est-elle unique ?

La notion de caractère est-elle liée à la notion d'alphabet ? Si oui comment ? et qu'est ce que cela induit sur les algorithmes de tri informatique ?

Comment savoir si deux chaînes de caractères représentent le même texte ?

A toutes ces questions, Unicode apporte des réponses, tantôt issues d'un héritage, tantôt issues de compromis ou de situations de fait.

Cet ouvrage tente d'apporter les principales réponses à ces questions.

Unité de code et point de code

Parce qu'il est plus complet et plus complexe, l'approche d'Unicode nécessite un vocabulaire précisant certaines notions très techniques absentes des autres codages de caractères. Ces notions sont reprises dans le glossaire en fin d'ouvrage (Glossaire).

Toutefois elles sont tellement fondamentales dans la compréhension d'Unicode qu'il est indispensable de les comprendre dès le début.

Unicode définit notamment les notions suivantes :

■ Unité de code

L'unité de code est l'unité dans laquelle un système informatique stocke ou échange les données. Il s'agit généralement d'un multiplé de 8, 16 ou 32 bits, que l'on appelle suivant le cas, octet, byte, mot, entier.

■ Point de code, une entité abstraite

Les caractères sont groupés en blocs en fonction de leur usage et des écritures supportées, et reçoivent une identification numérique unique appelée *point de code*, identifiée généralement sous la forme U+xxxx (où xxxx est un nombre hexadécimal de 4 à 6 chiffres, entre U+0000 et U+10FFFF). La plage définie permet d'attribuer jusqu'à 1 114 112 points de code.

■ Caractères et glyphes

Un point de code unicode définit une entité abstraite comme la “lettre latine majuscule a” ou le “chiffre cinq bengali”. Sur l'écran ou le papier n'apparaissent que le glyphe, la représentation visuelle du caractère.

Le standard Unicode ne définit pas l'image des glyphes. Il définit seulement ce que les caractères signifient.

- Texte, éléments de texte

Le standard Unicode ne définit pas ce qu'est un texte ou un élément de texte. Il définit seulement les *encoded characters* aussi connu sous le nom de point de code. nombre allant de 0 à 10FFFF16 (hexadécimal)

Un élément textuel est représenté par une séquence de point de code.

Composition et équivalence de caractère

Unicode contient de nombreux caractères. Pour maintenir la compatibilité avec des standards existants, certains d'entre eux sont équivalents à d'autres caractères ou à des séquences de caractères. Unicode fournit deux notions d'équivalence : canonique et de compatibilité, la première étant un sous-ensemble de la deuxième. Par exemple, le caractère n suivi du diacritique tilde ñ est canoniquement équivalent et donc compatible au simple caractère Unicode ñ, tandis que la ligature typographique ff est seulement compatible avec la séquence de deux caractères f.

encoded characters Le lien (ou les liens) entre un caractère abstrait et les représentations encodées. Par exemple 0041 et 030A sont des formes encodées. A et À Å avant d'être affichés sont des abstractions. Elles peuvent être codées de différentes manières : 0041 et 030A, 00C5 ou 212B.

Notes de bas de page

1. . D'après Édouard Chavannes – dans son essai sur « La divination par l'écaille de tortue » – les caractères chinois seraient inspirés des dessins, que l'on trouve sur plusieurs milliers de fragments d'écaille de tortue et d'os, retrouvés près de la ville de Ngan Yang dans le nord du Hunan en 1899. D'après lui, ces fragments seraient antérieurs au premier millénaire av. JC. À l'origine, il semblerait que ces fragments servaient à la divination et aux augures. Les gouvernants s'en servaient pour régir leur destin et celui de leur peuple.
2. Comme dans « peintre en lettre »
3. Photocomposition (<http://www.worldlingo.com/ma/enwiki/fr/Phototypesetting>) [[archive](#)]

Présentation du répertoire de caractères Unicode

Unicode et la norme ISO/CEI 10646 vise à donner à tout caractère un identifiant numérique unique, et ce de manière unifiée, quelle que soit la plate-forme informatique ou le logiciel.

Unicode et la norme ISO/CEI 10646 attribuent à chaque caractère un nom officiel au sein d'un répertoire commun unifié entre toutes les langues et tous les usages. Dès que le répertoire commun est approuvé, les caractères sont groupés en blocs en fonction de leur usage et des écritures supportées, et reçoivent une identification numérique unique appelée *point de code*, identifiée généralement sous la forme U+xxxx (où xxxx est un nombre hexadécimal de 4 à 6 chiffres, entre U+0000 et U+10FFFF). La plage définie permet d'attribuer jusqu'à 1 114 112 points de code.

Plans et blocs

Unicode ayant été construit par blocs, ces blocs forment un partitionnement du jeu de caractères. En soi, la connaissance de ce partitionnement n'a pas une grande utilité, il est donné dans l'annexe Plans à titre

purement informatif, pour donner une idée de l'étendue de l'Unicode. En fait, le jeu de caractères (en fait le jeu de point de codes) Unicode est découpé en plans eux-même découpés en blocs. Les plans contiennent un multiple de 65536 points de codes et sont les suivants sont les suivants:

- 1 Plan multilingue de base (PMB, 0000 à FFFF)
- 2 Plan multilingue complémentaire (PMC, 10000 à 1FFFF)
- 3 Plan idéographique complémentaire (PIC, 20000 à 2FFFF)
- 4 Plans complémentaires réservés (30000 à DFFFF)
- 5 Plan complémentaire spécialisé (PCS, E0000 à EFFFF)
- 6 Plans complémentaires à usage privé (F0000 à 10FFFF)

Caractères non graphiques, codes réservés et non affectés

Le standard Unicode a hérité des caractères de commande utilisé par certains protocoles ainsi que par l'ISO-2022 et par l'ASCII. Ces caractères ne sont pas des caractères affichables. Le standard leur attribue une valeur similaire à ceux que faisaient les standard précédents. La signification de chacun de ces codes peut dépendre du terminal ou du protocole utilisé.

Les points de code non caractère spéciaux correspondent à des valeurs dont l'utilisation est interdite par le standard.

Le marqueur BOM est un marqueur qui peut se trouver en début d'un fichier ou d'un flux Unicode.

Les caractères de présentation ne sont pas affichable mais permettent de contrôler les fonction de joignage, le contrôle des textes bidirectionnels, et de formats alternatifs.

Le caractère de remplacement est un caractère qui indique un caractère inconnu d'Unicode.

Il existe également des caractères combinatoires, notamment pour les diacritiques.

Caractères combinatoires

Unicode reprend le concept usuel de combinaison: Par exemple le Å se produit en partant de la lettre A sur laquelle on superpose une diacritique. En langage Unicode, U+0041 et U+030A sont des formes encodées. A et Å Å avant d'être affichés sont des abstractions. Elles peuvent être codées grace à cette combinaison : U+0041 et U+030A, même si dans ce cas de figure elles peuvent aussi être obtenues par les codes U+00C5 ou U+212B. Toutefois, les significations sont les suivantes:

```
U+00C5 LATIN CAPITAL LETTER A WITH RING ABOVE
U+212B ANGSTROM SIGN
U+030A COMBINING RING ABOVE
U+0041 LATIN CAPITAL LETTER A
```

Propriétés de caractère

Codage

Si chaque caractère peut être représenté à travers les point des codes définis par le répertoire Unicode, l'informatique nécessite de leur attribuer en plus d'un numéro, une représentation binaire.

Si l'on pourrait attendre à ce qu'Unicode définisse un multipléte suffisamment large pour représenter chacun des points de code Unicode, pour es raisons historiques, la réalité est toute autre.

En pratique, il est souvent nécessaire de représenter ces points de codes dans des unités définies préalablement à une époque où l'on pensait que huit ou seize bits seraient largement suffisants.

Pour être traité informatiquement, les points de code (qui définissent des caractères) doivent être représentable par une séquence de bits et/ou d'octets. Ceci est rendu possible au travers des systèmes de codages de caractère attribuant à chaque point des codes une séquence d'unité de code:

En dehors de la Chine populaire, les codages de caractère les plus utilisés sont sans doute l'UTF-8 et l'UTF-16, dont nous donnons quelques notions ci-après.

Les autres codages existant sont les suivants: UTF-7; UTF-8; CESU-8; UCS-2; UTF-16; UTF-32 (UCS-4); UTF-EBCDIC; SCSU; Punycode; GB 18030;

UTF-8

Le répertoire Unicode peut contenir plus d'un million de caractères, ce qui est bien trop grand pour être codé par un seul octet (limité à des valeurs entre 0 et 255). Techniquement, il s'agit de coder les caractères Unicode sous forme de séquences de un à quatre codet de un octet chacun.

Par exemple le caractère "€" (euro) est le 8365^e caractère du répertoire Unicode, son index, ou point de code est donc 8364 (on commence à compter à partir de 0).

La principale caractéristique d'UTF-8 est qu'elle est rétro-compatible avec la norme ASCII, c'est-à-dire que tout caractère ASCII se code en UTF-8 sous forme d'un unique octet, identique au code ASCII. Par exemple "A" (A majuscule) a pour code ASCII 65 et se code en UTF-8 par l'octet 65. Chaque caractère dont le point de code est supérieur à 127 (caractère non ASCII) se code sur 2 à 4 octets. Le caractère "€" (euro) se code par exemple sur 3 octets : 226, 130, et 172.

Exemples

Exemples de codage UTF-8

Type	Caractère	Point de code (hexadécimal)	Valeur scalaire		Codage UTF-8	
			décimal	binaire	binaire	hexadécimal
Contrôles	[NUL]	U+0000	0	0000000	00000000	00
	[US]	U+001F	31	0011111	00011111	1F
Texte	[SP]	U+0020	32	0100000	00100000	20
	A	U+0041	65	1000001	01000001	41
	~	U+007E	126	1111110	01111110	7E
Contrôles	[DEL]	U+007F	127	1111111	01111111	7F
Texte	[NBSP]	U+00A0	160	00010 100000	11000010 10100000	C2 A0
	é	U+00E9	233	00011 101001	11000011 10101001	C3 A9
	☐	U+07FF	2047	11111 111111	11011111 10111111	DF BF
	☐	U+0800	2048	0000 100000 000000	11100000 10100000 10000000	E0 A0 80
	€	U+20AC	8 364	0010 000010 101100	11100010 10000010 10101100	E2 82 AC
	☐	U+D7FF	55 295	1101 011111 111111	11101101 10011111 10111111	ED 9F BF
<i>Demi-codets</i>		U+D800 U+DFFF	<i>(néant)</i>		<i>(codage interdit)</i>	
<i>Usage privé</i>	☐	U+E000	57 344	1110 000000 000000	11101110 10000000 10000000	EE 80 80
	☐	U+F8FF	63 743	1111 100011 111111	11101111 10100011 10111111	EF A3 BF
Texte	豈	U+F900	63 744	1111 100100 000000	11101111 10100100 10000000	EF A4 80
	☐	U+FDCE	64 975	1111 110111 001111	11101111 10110111 10001111	EF B7 8F
<i>Non-caractères</i>		U+FDD0	64 976	1111 110111 010000	11101111 10110111 10010000	EF B7 90
		U+FDEF	65 007	1111 110111 101111	11101111 10110111 10101111	EF B7 AF

UTF-16

UTF-16 est un codage Unicode où chaque caractère est codé sur une suite de un ou deux mots de 16 bits.

L'UTF-16 fait maintenant partie intégrante de la norme Unicode, qui dans son chapitre 3 *Conformance* la définit de façon très stricte.

L'UTF-16 n'est pas l'UCS-2 qui est le codage, plus simple, de chaque caractère sur deux octets. Ces deux normes sont pourtant appelées toutes les deux Unicode, car le codage est le même tant que l'on n'utilise pas les plages U+D800 à U+DFFF (en principe réservées) et les plages après U+FFFF (peu utilisées en occident).

L'UTF-16 est en particulier utilisé dans les environnements windows. Dans ce système, les API dites

unicode utilisent ce standard. Il en va de même du système NTFS.

UTF-16 est le standard de chaînes de caractères utilisé par l'UEFI^[1].

Description

Les points de code qui peuvent être représentés doivent être dans l'intervalle de validité U+0000 à U+10FFFF, et ne doivent pas être affectés à un non-caractère. Tous les caractères possibles dans Unicode possèdent de tels points de codes.

Tout point de code qui n'est pas un non-caractère, et dont la valeur peut être codée sur un seul codet de deux octets (16 bits), c'est-à-dire tout point de code U+0000 à U+D7FF et U+E000 à U+FFFD, est stocké sur un seul mot de 16 bits (la plage de non-caractères U+D800 à U+DFFF est donc exclue, c'est-à-dire les points de code dont les 5 bits de poids fort sont 11011).

Dans les autres cas, le caractère est un point de code d'un plan supplémentaire (donc entre U+10000 et U+10FFFF et dont les 16 bits de poids faible ne doivent pas égaier 0xFFFE ou 0xFFFF) ; il est alors stocké sur 2 mots (codets) successifs de 16 bits chacun, dont les valeurs correspondent aux points de codes réservés dans les *demi-zones d'indirection* allouées dans le plan multilingue de base des normes Unicode et ISO/CEI 10646 :

- le premier mot aura les 6 bits de poids fort égaux à 110110 et sera donc compris dans l'intervalle [0xD800 .. 0xDBFF] (ici en numération hexadécimale) ; ce mot contiendra dans ses 10 bits de poids faible les 10 bits de poids fort de la différence (représentée sur 20 bits) entre le point de code à stocker et le premier point de code supplémentaire U+10000 ;
- le second mot aura les 6 bits de poids fort égaux à 110111 et sera donc compris dans l'intervalle [0xDC00 .. 0xDFFF] (ici en numération hexadécimale) ; ce mot contiendra dans ses 10 bits de poids faible les 10 bits de poids faible du point de code à stocker.

Puis suivant le format de stockage des mots de 16 bits dans un flux ordonné d'octets, deux systèmes sont possibles pour le codage final :

Principe du codage UTF-16 en big endian (*on représente ici les bits*)

Numéro du caractère	00000000 000uuuuu xxxxxxYY YYYYYYYY			
Codage UTF-16BE (sur 2 octets)			xxxxxxxY	YYYYYYY
	(seulement si uuuuu = 00000)			
Codage UTF-16BE (sur 4 octets)	110110ww	wwxxxxxx	110111YY	YYYYYYY
	avec www = uuuuu - 1 (si uuuuu > 00000)			

Principe du codage UTF-16 en little endian (*on représente ici les bits*)

Numéro du caractère	00000000 000uuuuu xxxxxxYY YYYYYYYY			
Codage UTF-16LE (sur 2 octets)			YYYYYYY	xxxxxxxY
	(seulement si uuuuu = 00000)			
Codage UTF-16LE (sur 4 octets)	wwxxxxxx	110110ww	YYYYYYY	110111YY
	avec www = uuuuu - 1 (si uuuuu > 00000)			

L'indication du type de codage utilisé (ordre des octets) peut être implicite pour le protocole utilisé, ou précisé explicitement par ce protocole (en indiquant par exemple les noms réservés "UTF-16BE" ou "UTF-16LE" dans un entête de *charset* MIME). Si le protocole ne permet pas de spécifier l'ordre des octets, et s'il permet l'une ou l'autre des alternatives, on pourra utiliser le codage UTF-16 du point de code valide U+FEFF comme indicateur en tête du flux de données (car un changement d'ordre de ses octets à la lecture du flux conduira à un point de code U+FFFE, valide dans Unicode mais affecté à un non-caractère et donc interdit dans ce cas dans tout flux UTF-16. Ce point de code ainsi représenté (appelé marque d'ordonnement des octets, *byte order mark* en anglais, abrégé *BOM*) ne sera codé qu'au début du flux de données, et permet de savoir comment a été codé le flux :

Modèle:1er octet	Modèle:2e octet	Codage effectif
0xFE	0xFF	big endian
0xFF	0xFE	little endian

Si l'une des deux séquences de deux octets chacune est présente en tête de flux, le type de codage en est déduit et la séquence est retirée du flux : elle ne représente aucun caractère du texte stocké dans ce flux de données. Si aucune des deux séquences ne figure en tête du flux de données, la norme Unicode spécifie que le flux doit être décodé en big endian (UTF-16BE).

Références

1. <http://x86asm.net/articles/uefi-programming-first-steps/>

Applications

Unicode est en progression constante, surtout sur Internet. Aujourd'hui, on peut considérer qu'il s'agit du jeu de caractères standard à utiliser partout où du texte doit être utilisé, sauf spécifications contraire, notamment lorsque la compatibilité avec des systèmes dont la conception est vieille de plusieurs dizaines d'année est recherchée.

Toutefois, il faut garder à l'esprit que si une application utilise es caractères Unicode, elle n'est pas pour autant toujours conformes à tous les critères de conformance définis par le standard.

L'utilisation d'Unicode comme jeu de caractère par une application se bases souvent sur l'un de standards UTF-8 u UTF-16.

De par sa nature, UTF-8 est d'un usage de plus en plus courant sur internet, et dans les systèmes devant échanger de l'information. L'UTF-16 est en particulier utilisé dans les environnements Windows.

Environnements

éco-systèmes Gnu, Linux et compatibles

UTF-8 Il s'agit également du standard Unicode le plus utilisé dans les éco-systèmes Gnu, Linux et compatibles.

Windows

L'UTF-16 est en particulier utilisé dans les environnements Windows. Dans ce système, les API dites Unicode utilisent ce standard.

Il en va de même des système de fichier NTFS, Virtual FAT, Joliet (cédéromes) et ReFS qui utilisent une jeu de caractères UTF-16 pour les noms de fichiers.

Environnements réseaux

dans les systèmes modernes, le partage de fichier sur réseau est conçu pour échanger des noms de fichier Unicode.

- C'est le cas de Samba (connu dans Windows sous le nom de "voisinage réseau") à partir des versions Windows NT, 200x, XP
- C'est aussi le cas d'Active directory et de certains protocoles FTP
- RFC 5198: Unicode Format for Network Interchange

Limites du DOS

Le système DOS et les logiciels associés n'ont pas été adapté aux évolution de l'Unicode. L'incapacité de ces logiciels à s'adapter au monde moderne à conduit à la désuétude de leur usage.

Matériel électronique

Unicode (en l'occurrence UCS-2) est également considéré dans les plus basses couches du logiciel, aux prise directes avec le matériel;

C'est le cas de l' *UEFI Shell Specification* (May 22, 2012 Revision 2.0 Errata “A”).

UTF-16 ou bien l'UCS-2 est le standard de chaînes de caractères utilisé par l'UEFI^[1].

Unicode est également utilisé par le logiciel multiboot Grub 2.0^[2].

Logiciels et formats de fichiers

Bureautique

A priori, Unicode est le standard sous-jacent utilisé par des logiciels devant représenter du texte comme Microsoft Office, OpenOffice.

Dans OpenOffice 3.2.1, des caractères Unicode sont proposés dans le menu «Insertion → caractères spéciaux».

Navigation Internet

A priori, Unicode est le standard sous-jacent utilisé par des logiciels devant représenter du texte comme Firefox, Chrome.

Développement logiciel

A priori, Unicode est le standard sous-jacent utilisé par des machine virtuelles comme l'environnement Java

ou l'environnement dot Net.

Des langages récents comme Perl et Python offrent également une assez bonne approche d'Unicode.

Des langages hérités comme les scripts shells Unix ne prennent pas en compte spécifiquement l'Unicode et peuvent présenter des aspects particuliers.

Ce point est traité plus en profondeur dans le chapitre Programmation.

Internet et télécommunications

SMS

Sur téléphones mobiles et dans les SMS Unicode n'est pas toujours disponible.

Sites internet

Unicode est le codage de base de nombreux sites internet parmi lesquels on trouve pour ne donner que deux exemples et non des moindres, le site de Wikipédia et le site du parlement européen. Concrètement, le codage utilisé est UTF-8.

Limites du courriel

Dans ses origines nord-américaine, le courriel est une chose qui contient du texte ASCII. Les caractères qui peuvent être utilisés étaient d'abord ASCII, puis des encodages régionaux. Aujourd'hui, certains logiciels supportent également l'UTF-8, ce qui permet d'augmenter le nombre de caractères différents que l'on peut utiliser dans un même courriel.

Avec la technologie MIME (Multipurpose Internet Mail Extensions), différents fichiers informatiques peuvent être joints au courriel.

L'utilisation du format HTML pour la structuration ou la mise en forme des courriels est possible, mais souffre d'un manque important d'interopérabilité,^[3] Il en est de même du recours aux feuilles de style en cascade (CSS) pour leur présentation^[4].

UTF-8 et les caractères régionaux ne sont pas toujours interopérables, en fonction du logiciel de messagerie utilisé par le destinataire et de sa localisation géographique.

Références

1. <http://x86asm.net/articles/uefi-programming-first-steps/>
2. <http://www.gnu.org/software/grub/manual/grub.html>
3. souligné en 2007 par le séminaire *Mail HTML* du W3C anglais W3C HTML Mail Workshop, 24 May 2007, Paris, France (<http://www.w3.org/2007/05/html-mail/>) ^[archive]
4. anglais David Greiner, A Guide to CSS Support in Email: 2007 Edition (http://www.campaignmonitor.com/blog/archives/2007/04/a_guide_to_css_support_in_email_2.html) ^[archive], Campaign Monitor

Fonctionnalités usuelles et algorithmes

Le standard Unicode et les différents documents y afférant traitent des domaines suivants pour lesquels ils peuvent servir de guide:

- algorithmes pour gérer les changements de sens d'écriture de gauche à droite et de droite à gauche (comme en arabe ou en hébreux)
- ordre alphabétiques et collation dans différentes langue
- conversion de casse
- équivalences de texte
- division des mots et coupure de ligne
- régionalisation des nombres et autres considérations
- spécificités des combinaisons et ré-ordonnancement des langues sudasiatiques
- problématique de sécurité liés à la ressemblance de caractères différents à travers le monde.

sens d'écriture de gauche à droite et de droite à gauche (comme en arabe ou en hébreux)

Certains systèmes d'écritures, tels que l'alphabet arabe et hébreu, s'écrivent de droite à gauche (*Right-To-Left*, RTL, en anglais). Dans ce cas, le texte commence du côté droit de la page et se termine du côté gauche, au contraire du sens d'écriture conventionnel de gauche à droite (*Left-To-Right*, LTR) des langues utilisant l'alphabet latin (telles que le français). Lorsqu'un texte LTR est mélangé avec un texte RTL dans le même paragraphe, chaque type de texte doit être écrit dans son propre sens, phénomène connu sous le nom de **texte bidirectionnel**.

En savoir plus sur wikipedia: [wikipedia:Texte bidirectionnel](#)

ordre alphabétiques et collation dans différentes langue

Le **classement alphabétique** ne doit pas être confondu avec le simple *ordre alphabétique*, qui ordonne un alphabet et ne concerne donc que des signes, tandis que le *classement* alphabétique porte sur des mots de longueurs diverses et peut comprendre plusieurs milliers d'éléments. Bien qu'on en ait vu quelques tentatives sous diverses latitudes et à diverses époques^[1], il apparut véritablement avec la Souda, une encyclopédie byzantine du X^e siècle contenant 30 000 entrées^[2].

L'ordre alphabétique resta longtemps une spécialité purement européenne : jusqu'au milieu du XX^e siècle, les dictionnaires non-occidentaux demeurèrent classés par thèmes ou par racines grammaticales.

Le classement alphabétique révolutionna l'activité commerciale en Occident. Il devint possible avec lui de travailler avec des listes de plusieurs centaines – voire de milliers – de noms de clients, articles, fournisseurs, villes, sous-traitants, créanciers, débiteurs et correspondants divers sans plus de difficulté de recherche que sur une liste désordonnée d'une vingtaine de noms. Il révolutionna aussi l'activité savante et la structure des ouvrages grâce à la constitution d'index et à la mise en place de dictionnaires et d'encyclopédies faciles à consulter.

Pour des raisons d'habitudes, d'ancienneté du principe, ou de facilité de mise en œuvre, de nombreux développeurs de logiciels utilisent ou ont utilisé le classement selon l'ordre des codes dans le codage de caractères utilisé (par exemple ASCII ou UTF-8), nommé ordre lexicographique. Ce classement coïncide avec le classement alphabétique pour les mots contenant uniquement des lettres sans diacritique et toutes en majuscule (ou en minuscules), mais donne un résultat généralement incorrect dès qu'il y a des diacritiques, des espaces, des signes de ponctuations ou un mélange de lettres capitales et minuscules (ce dernier point est toutefois facilement résolu en convertissant tout en capitale).

La notion de "LOCALE" dans les systèmes d'exploitation permet aux fonctions de comparaison de mots d'effectuer les bonnes équivalences dans la langue considérés. En français, (A,À,Â,Ä,ä,â,à), etc. - alors que les codes de ces lettres sont loin d'être voisins. Ainsi, il n'y a pas à recompiler une même application pour chaque langue existante.

Ce sujet est discuté, en langue anglaise par les règles technique Unicode n°10 (Unicode Technical Standard «UTS 10 Unicode Collation Algorithm»).

conversion de casse



Cette section est vide, pas assez détaillée ou incomplète.

équivalences de texte

Unicode contient de nombreux caractères. Pour maintenir la compatibilité avec des standards existants, certains d'entre eux sont équivalents à d'autres caractères ou à des séquences de caractères. Unicode fournit deux notions d'équivalence : canonique et de compatibilité, la première étant un sous-ensemble de la deuxième. Par exemple, le caractère n suivi du diacritique tilde ñ est canoniquement équivalent et donc compatible au simple caractère Unicode ñ, tandis que la ligature typographique ff est seulement compatible avec la séquence de deux caractères f.

La normalisation Unicode est une normalisation de texte qui transforme des caractères ou séquences de caractères en une même représentation équivalente, appelée « forme normale » dans cet article. Cette transformation est importante, car elle permet de faire des comparaisons, recherches et tris de séquences Unicode. Pour chacune des deux notions d'équivalence, Unicode définit deux formes, l'une composée, et l'autre décomposée, conduisant à quatre formes normales, abrégées NFC, NFD, NFKC et NFKD, qui seront détaillées ci-dessous et qui sont aussi décrites dans Normalisation Unicode.

Les deux notions d'équivalence

L'équivalence canonique est une forme d'équivalence qui préserve visuellement et fonctionnellement les caractères équivalents. Ils ont un codage binaire différents mais représentent un texte identique.

L'équivalence de compatibilité correspond plutôt à une équivalence de texte ordinaire, et peut réunir ensemble des formes distinctes sémantiquement.

Pour aller plus loin

- [Wikipedia:Équivalences unicode](#)
- [UAX 15 Unicode Normalization Forms](#)

division des mots et coupure de ligne



Cette section est vide, pas assez détaillée ou incomplète.

régionalisation des nombres et autres considérations



Cette section est vide, pas assez détaillée ou incomplète.

spécificités des combinaisons et ré-ordonnement des langues sudasiatiques



Cette section est vide, pas assez détaillée ou incomplète.

problématique de sécurité liés à la ressemblance de caractères



Cette section est vide, pas assez détaillée ou incomplète.

Références

- <http://www.unicode.org/reports/>

1. Voir Knuth, *Sorting and Searching*, Addison-Wesley, 1973.
2. Peter Burke, 2000, p. 184.

Programmation

Si ce livre veut s'adresser à un lectorat néophyte, cette section s'adresse plus particulièrement à toutes les personnes qui travaillent dans les métiers liés au développement logiciel.



Cette section est vide, pas assez détaillée ou incomplète.

Bibliothèques logicielles

ICU

La bibliothèque logicielle multi plate-forme ICU permet de manipuler des données unicodées.

Elle est en particulier utilisée à partir de LibreOffice 4.0^[1].

Le langage

Un support d'Unicode spécifique à certaines plates-formes (non compatible quant au code-source) est également fourni par les systèmes modernes (Java, MFC, GNU/Linux).

Les types à utiliser pour stocker des variables Unicode, sont les suivants :

Types compatibles avec Unicode dans les langages de programmation

Langage de programmation	Type pour le codage d'un seul caractère ou d'une partie de caractère	Type pour le codage d'un texte quelconque
C	<code>char[4]</code> ^[alpha 1] ou <code>wchar_t[2]</code> ^[alpha 2]	<code>char[]</code> ou <code>wchar_t[]</code>
C++	<code>char[4]</code> ^[alpha 1] ou <code>wchar_t[2]</code> ^[alpha 1]	<code>char[]</code> ou <code>wchar_t[]</code> ou <code>std::string</code> ou <code>std::wstring</code>
Java	<code>char[2]</code> ou <code>int</code> ^[alpha 3]	<code>char[]</code> ou <code>String</code>
Bibliothèque ICU (pour C/C++ ou Java)	<code>UChar</code>	<code>UChar[]</code> ou <code>String</code> , <code>UnicodeString</code>
JavaScript ou ECMAScript	<code>char</code> ^[alpha 4]	<code>string</code>
C# ou J#	<code>char</code>	<code>string</code>
Delphi	<code>char[4]</code> ^[alpha 1] ou <code>widechar[2]</code>	<code>string</code> ^[alpha 1] ou <code>widestring</code>
Python		<code>unicode</code>
Vala	<code>uint8</code> ^[alpha 5] ou <code>char</code> ^[alpha 6] ou <code>unichar</code> ^[alpha 7]	<code>string</code> ^[2]

Notes du tableau alpha

1. En UTF-8
2. On notera toutefois que le type `wchar_t` du langage C ne permet pas toujours de coder tous les caractères Unicode, car la norme de ce langage ne prévoit pas de nombre minimum suffisante pour ce type standard. Cependant de nombreux compilateurs du langage définissent `wchar_t` sur 32 bits (voire 64 bits sur les environnements manipulant les entiers standards sur 64 bits), ce qui suffit pour stocker n'importe quel point de code Unicode normalisé. Mais d'autres compilateurs représentent `wchar_t` sur 16 bits (notamment sous Windows en environnement 16 ou 32 bits), voire sur 8 bits seulement (notamment dans les environnements embarqués ne disposant pas d'un système d'exploitation d'usage général) car `wchar_t` peut utiliser la même représentation que le type `char` qui compte un minimum de 8 bits.
3. De manière similaire au C et au C++, le langage Java dispose de type unitaire permettant de coder 16 bits, mais ne permettant pas de coder un seul point de code d'une valeur quelconque (le type natif `char` est un entier positif sur 16 bits seulement). Pour manipuler les caractères normalisés hors du premier plan, il faut utiliser une paire de codets, chacun contenant une valeur égale aux deux codets définis par la forme UTF-16. Aussi les types d'objets `String` ou `char[2]` sont les plus appropriés pour représenter un caractère Unicode. Depuis Java 1.4.1, la bibliothèque standard fournit un support

complet d'Unicode grâce au type natif `int` (qui est un entier défini sur 32 bits) et aux méthodes statiques de la classe standard `Character` (cependant un objet instancié de ce type `Character` ne permet pas, tout comme le type natif `char`, de stocker n'importe quel point de code).

4. JavaScript comporte diverses implémentations non normalisées dont certaines plus anciennes ne supportent pas plus de 16 bits par caractère, et parfois seulement 8 bits. Toutefois la norme ECMAScript de ce langage définit une classe utilitaire `Character` sur 32 bits (en fait basée sur la classe `Number`) devant supporter tous les points de code des 17 plans normalisés, tandis que les chaînes de caractères utilise des caractères codés obligatoirement sur 16 bits (mais sans restriction renforçant l'appariement des unités de code UTF-16, les chaînes ECMAScript de type `String` n'étant pas restreintes au seul codage UTF-16 mais étant des vecteurs de constantes entières codées sur 16 bits sans restriction, afin d'assurer l'interopérabilité avec Java et d'autres langages qui eux non plus ne renforcent pas les restrictions de conformité UTF-16 dans leurs types natifs de données). Ces deux langages ne supportent pas de typage explicite des variables, le type étant défini dynamiquement par les valeurs qu'on leur assigne (aussi, plusieurs représentations internes sont possibles, leurs différences étant normalement transparentes pour le programmeur).
5. un octet

6. non spécifié

7. un caractère complet

Autres bibliothèques

La bibliothèque Qt permet aussi de gérer Unicode. <http://doc.qt.digia.com/qt/unicode.html>

Le cas de Linux

Dans le cas de Linux, le développement en langage C peut se faire avec de interfaces spécifiques, en 2001 [3].

Notes

1. <https://fr.libreoffice.org/telecharger/nouveautes-et-correctifs-de-la-version-4-0/>
2. valadoc.org/#!/api=glib-2.0/string
3. <http://www.ibm.com/developerworks/library/l-linuni/index.html> Linux Unicode programming How to incorporate and utilize Unicode for foreign language support

Unicode souffre toutefois encore d'un faible support des expressions rationnelles par certains logiciels, même si des bibliothèques comme ICU et Java peuvent les supporter. Un tel support n'a pas encore été standardisé pour ECMAScript et n'est fourni qu'avec l'aide de bibliothèques créées avec le langage ou des interfaces d'interopérabilité avec d'autres systèmes (notamment avec CORBA, COM) ou langages (notamment C++ et Java).

Voir aussi

- Coder avec Unicode

Programmation/International Components for Unicode

International Components for Unicode (ICU) est un projet open source qui fournit des bibliothèques de traitement utilisables dans les langages informatiques C/C++ et Java, afin de prendre en charge les textes utilisant le répertoire universel de caractères codés (UCS, normalisé dans la norme ISO/CEI 10646 et le standard informatique Unicode), l'internationalisation et la localisation des logiciels. ICU est largement portable vers de nombreux systèmes d'exploitation et environnements. Il donne aux applications les mêmes comportements et résultats sur toutes les plateformes et entre les langages de programmation fournissant une interface avec les langages C, C++ ou Java.

Le projet ICU et licence de distribution et d'utilisation

Le projet ICU est un projet libre, collaboratif et indépendant des organisations commerciales ; il est destiné à

écrire, faire évoluer et distribuer ces bibliothèques dont les codes sources sont disponibles en même temps que des versions précompilées directement utilisables dans d'autres logiciels ; il est activement supporté et utilisé par IBM qui en est l'actuel principal promoteur, et par d'autres entreprises, organisations et individus créant du logiciel.

Ces bibliothèques sont fournies avec une licence ouverte et libre (dérivée de la licence X) et compatible avec les licences libres (de type GPL selon les critères du copyleft de la Free Software Foundation) et les licences ouvertes (selon les critères de l'Open Source Initiative), permettant la réutilisation, la modification, et la redistribution ; cette licence est gratuite mais sans garantie offerte, à la seule condition de fournir une copie de cette licence et de mentionner l'origine (copyright) du logiciel original (dont *IBM* et les participants au projet ICU sont détenteurs des droits d'auteur sur l'œuvre collective).

Services fournis

Les principaux services fournis par les bibliothèques ICU sont les suivants :

- Texte : gestion de texte Unicode, propriétés normatives ou informatives des caractères codés (et de certaines combinaisons de caractères codés), le plus souvent indépendantes de la langue utilisée.
- Transcodages : conversion du texte entre de nombreux jeux de caractères codés sur un octet ou plusieurs octets d'une part (issus de normes nationales ou internationales ou de standards de l'industrie informatique), et d'autre part le répertoire universel de caractères codés (UCS) défini dans la norme internationale ISO/CEI 10646 et le standard informatique Unicode.
- Transformations de base normalisées : délimitation des séquences inséparables de caractères codés et normalisations Unicode (NFC, NFD, NFKC, NFKD) ; algorithme *NamePrep* (pour le support de l'architecture des noms de domaines Internet internationalisés).
- Assistance pour l'affichage et la saisie du texte : support de la disposition du texte selon l'ordre visuel des graphèmes, et analyse des variantes graphiques contextuelles requises dans les écritures complexes (arabe, hébreu, devanagari, thaï, etc.).
- Comparaison : algorithmes rapides de tri alphabétique multi-niveau pour le tri et la recherche de texte, selon la catégorie des caractères, l'écriture ou la casse, et paramétrables selon diverses règles linguistiques ou conventions régionales.
- Analyse lexicale : délimitation des graphèmes, mots, phrases et lignes ; expressions rationnelles utilisant le jeu Unicode complet et ses propriétés.
- Transformations lexicales simples : forçage ou normalisation de la casse (capitales, majuscules et minuscules) ou des variantes de présentation de caractères.
- Transformations lexicales complexes : transcriptions et translittérations selon les langues et conventions nationales ou internationales.
- Formatage et analyse syntaxique du texte : dates, heures, nombres, monnaies et messages, conformes aux normes internationales ou basés sur des règles de localisation (y compris grammaticales).
- Temps : calculs et transformations de données temporelles selon de nombreux calendriers et fuseaux horaires.
- Régionalisation : architecture complète pour les données de localisation et paquets de ressources, support du format d'échange LDML et intégration des données communes de localisation issues du projet CLDR dans ce format.

Origine et développement

Originellement, ICU a été intégralement écrit en Java. Mais certains travaux initiaux d'ICU viennent du *framework* pour C++ écrit par l'entreprise Taligent, qui fut rachetée par IBM.

Certaines des fonctionnalités liées à la gestion de texte, au formatage des dates, etc. ont été réécrites en Java

pour devenir les *API d'internationalisation pour JDK 1.1*, qui ont été proposées à Sun Microsystems par l'équipe ICU pour l'intégration à la plateforme de base Java. Une grande portion du code initial en Java existe toujours dans les paquets `Modèle:Javadoc:SE` et `Modèle:Javadoc:SE` qui intègrent une version limitée de l'actuelle bibliothèque ICU pour Java.

Ces fonctionnalités furent ensuite portées et étendues en C et en C++ pour surmonter les défauts d'internationalisation de ces langages et de nombreux autres disposant de bibliothèques très incomplètes pour le traitement de l'internationalisation des logiciels et le support correct et complet des algorithmes de traitement du texte basés sur les spécifications du standard Unicode et ses annexes, ainsi que sur les travaux issus d'autres normes internationales dans ces domaines.

D'ordinaire un système d'exploitation fournit ces fonctionnalités, mais le support d'une telle API d'internationalisation n'est pas assuré de façon homogène par tous les systèmes d'exploitation, ni d'une façon suffisante pour prendre en charge un nombre aussi important d'écritures, langues et conventions régionales.

ICU a été livré en 1999 en tant que projet de développement en source ouvert, sous le nom *IBM Classes for Unicode*. Suite au transfert des droits de propriété intellectuelle d'IBM à une organisation indépendante et à but non lucratif (où d'autres acteurs peuvent participer et prendre part au processus de décision ou d'évaluation des nouvelles fonctionnalités), il fut finalement renommé *International Components For Unicode* (ICU), et sa licence d'utilisation et de distribution a été libéralisée.

La version Java existe aujourd'hui sous le nom ICU4J, et la version C/C++ existe aujourd'hui sous le nom ICU4C. Les deux bibliothèques disposent de fonctionnalités pratiquement identiques et évoluent selon la même architecture générale en fonction des besoins propres à chaque plateforme (comme la plateforme de base normalisée pour les langages C et C++ ne dispose pas de fonctionnalités suffisantes et est la plus hétérogène, ICU4C les complète pour les amener au même niveau que celles disponibles en Java, qui en a déjà intégré une partie significative et que la version ICU4J n'a pas besoin de remplacer, et les autres différences mineures sont généralement gommées par la mise à niveau de l'un ou l'autre projet lorsque leur développement spécifique est nécessaire).

Le projet ICU et ses deux sous-projets continuent à être développés en parallèle pour le support plus avancé d'Unicode, et de façon plus générale, celui de l'internationalisation (i18n) des logiciels selon l'état de l'art en la matière, et selon l'évolution des normes dont le projet ICU est même devenu un modèle de référence presque incontournable (utilisé aussi dans le développement et le test des évolutions des diverses normes et travaux collaboratifs qu'ICU supporte).

Exemple

Exemple de code C++ utilisant la bibliothèque ICU, pour le formatage de nombres décimaux sous forme ici de noms correspondant à une suite d'intervalles de valeurs.

```
#include <unicode/unistr.h>
#include <unicode/ustream.h>
#include <unicode/choicfmt.h>

int main(int argc, char *argv[]) {
    // Bornes inférieures des intervalles de valeurs.
    double limits[] = {1, 2, 3, 4, 5, 6, 7};
    // Noms donnés à chaque intervalle.
    UnicodeString weekdayNames[] = {
        "lundi", "mardi", "mercredi", "jeudi", "vendredi", "samedi", "dimanche"};
    // Crée un format correspondant à la liste de choix bornée par les limites.
    ChoiceFormat fmt(limits, weekdayNames, 7);
    // Déclare une classe destinée à stocker des chaînes de caractères Unicode.
```

```

UnicodeString str;
for (double x = 1.0; x <= 8.0; x += 1.0) {
    // Formate selon la liste de choix le nombre x dans la chaîne str.
    fmt.format(x, str);
    // Affiche le nombre ainsi que la chaîne formatée.
    cout << x << " -> " << str << endl;
}
cout << endl;
return 0;
}

```

Liens externes

- (anglais) ICU website (<http://www.icu-project.org/>) [archive]

Conformance

La conformance est un sujet traité dans le chapitre 3 de la spécification Unicode.

<http://www.unicode.org/versions/Unicode6.2.0/ch03.pdf>

Liste de symboles utiles

Ce chapitre donne une liste de caractères utiles classés en fonction des différents domaines permettant des les retrouver en fonction de leur utilité.

Le classement du standard (Unicode reference tables) étant moins facile à utiliser).

Typographie

§ 00A7 section

¶ 00B6 paragraph

· 00B7 interpunct, middle dot

¼ 00BC one-quarter

½ 00BD one-half

¾ 00BE three-quarters

– 2012 figure dash

– 2013 en dash

— 2014 em dash

— 2015 quotation dash

‘ 2018 left single quote

’ 2019 right single quote

“ 201C left double quote

” 201D right double quote

† 2020 dagger

‡ 2021 double dagger

• 2022 bullet

? 203D interrobang

** 2042 asterism

𐞀 2150 one-seventh

𐞁 2151 one-ninth

𐞂 2152 one-tenth

⅓ 2153 one-third

⅕ 2155 one-fifth

⅙ 2159 one-sixth

⅛ 215B one-eighth

← 2190 left arrow

↑ 2191 up arrow

→ 2192 right arrow

↓ 2193 down arrow

↵ 21A9 carriage return symbol

␣ 2423 space symbol

◻ 25A1 square

★ 2605 starf, bigstar

♣ 269C fleur-de-lys

✓ 2713 check mark

✕ 2717 cross mark

𐞀 2E3A two-em dash

𐞁 2E3B three-em dash

“ 3003 ditto

Symboles typographiques spécifiques

00A0 non-breaking space - invisible forced space.

00AD soft hyphen - is shown where it is inserted in a word if the word is too long for the line and breaks just after the soft hyphen. Otherwise the hyphen is invisible. In several wordprocessor soft hyphen can be entered by holding the control key and at the same time pressing the hyphen key, and is shown as a hyphen with gray background.

Mathématiques

Opérateurs unaires

$\sqrt{\quad}$ U+221A radic, Sqrt, radical, square root
 $\sqrt[3]{\quad}$ U+221B cube root
 $\sqrt[4]{\quad}$ U+221C fourth root
 \pm U+00B1 plus or minus
 \mp U+2213 minus or plus
 $'$ U+2032 prime, derivative
 $"$ U+2033 Prime, double prime
 $'''$ U+2034 tprime, triple prime
 \prod U+220F N-ary product
 \sum U+2211 N-ary summation

Opérateurs binaires

$-$ U+2212 minus
 \times U+00D7 multiplication sign
 \cdot U+22C5 sdot
 $/$ U+2215 division slash
 \div U+00F7 division sign

Relations

\propto U+221D prop, propto, Proportional, vprop, varpropto, proportional to
 \ll U+226A much less than
 \gg U+226B much greater than
 \triangleq U+225C trie, triangleq, delta equal to
 \triangleq U+225D equal to by definition

$\stackrel{?}{=}$ U+225F equest, questioned equal to
 \neq 2260 ne, NotEqual, not equal to
 \equiv 2261 equiv, Congruent, equivalent to
 \approx 2243 asymptotically equal to
 $\not\approx$ 2244 not asymptotically equal to
 \cong 2245 approximately equal to
 \approx 2246 approximately but not actually equal to
 $\not\approx$ 2247 neither approximately nor actually equal to
 \simeq 2248 almost equal to
 $\not\approx$ 2249 not almost equal to
 $|$ 2223 divides
 \nmid 2224 does not divide

Calculs, calcul vectoriel, équations différentielles

∞ 221E infinity
 ∂ 2202 partial differential
 \int 222B integral
 \iint 222C double integral
 \iiint 222D triple integral
 \oint 222E contour integral
 \oiint 222F surface integral
 \iiint 2230 volume integral
 \int_{\curvearrowright} 2231 clockwise integral
 \oint 2232 clockwise contour integral
 \oint 2233 anticlockwise contour integral

∇ 2207 nabla, Del, backward difference, gradient

Ensembles

\subset 2282 subset, included in, proper subset
 \supset 2283 superset, includes, proper superset
 \cap 2229 intersection
 \cup 222A union
 \in 2208 isin, isinv, Element, in, element of (large symbol)
 ϵ 220A element of (small symbol)
 \notin 2209 notin, NotElement, notinva, not element of
 \ni 220B niv, ReverseElement, ni, SuchThat, contains as member (large symbol)
 \ni 220D contains as member (small symbol)
 $\not\ni$ 220C notni, notniva, NotReverseElement, does not contain as member
 \emptyset 2205 empty, emptyset, emptyv, varnothing, empty set

Typographie mathématique

\vdots 22EE vellip, vertical ellipsis
 \cdots 22EF hellip, horizontal ellipsis
 $\cdot\cdot$ 22F0 utdot, rising dots,

up right diagonal ellipsis
 ∙ ∙ 22F1 dtdot, falling dots,
 down right diagonal ellipsis
 ‹ 27E8 lang , left angle
 bracket
 › 27E9 rang, right angle
 bracket

Logique mathématique

∧ 2227 et Logical AND
 ∨ 2228 ou Logical OR

Science

° 00B0 degree
 ′ 2032 prime
 ″ 2033 double prime
 μ 00B5 micro, mu

Ingénierie

⌀ 232D cylindricity (drafting)
 Ω 2126 omega, ohm
 ⌘ 2104 Centerline symbol
 †† FE4A overline dashes
 ‡‡ FE4E underline dashes

Finances

Monnaies

€ 20AC Euro
 £ 00A3 Sterling

■ 220E, cqfd END OF PROOF
 ∴ 2234, THEREFORE ,
 there4, therefore, Therefore
 ∵ 2235, BECAUSE ,
 becaus, because, Because
 ∀ 2200, pour tous FOR ALL
 , forall, ForAll
 ∃ 2203, il existe THERE
 EXISTS , exist, Exists
 ∄ 2204, il n'existe pas
 THERE DOES NOT EXIST ,
 nexist, NotExists, nexists

Geometrie

∠ 2220 ang, angle
 ∠ 2221 angmsd, measured
 angle
 ∠ 2222 spherical angle

Autre

ℓ 2113 Small script letter L,
 litre.

₹ 20B9 Indian Rupee
 ¥ 00A5 Yen
 ¢ 00A2 Cent

‰ 2030 Per mille
 ‰ 2031 Basis point, per
 ten thousand, permyriad

Autre



Autres symboles

Musique



























♪ 2669 quarter note
 ♪ 266A eighth note
 ♪ 266B beamed eighth
 notes
 ♪ 266C beamed sixteenth













notes
 ♭ 266D music flat sign
 ♮ 266E music natural sign
 ♯ 266F music sharp sign
 ◡ 1D110 fermata
 ◡ 1D11C six-string
 fretboard (tablature)
 ◡ 1D11E G clef

◡ 1D12A double sharp
 ◡ 1D134 common time
 ◡ 1D13B whole rest
 ◡ 1D13D quarter rest
 ◡ 1D15E half note
 ◡ 1D161 sixteenth note
 ◡ 1D192 crescendo
 ♪ 1F3B5 musical note















-  1F3B8 guitar
-  1F3BB violin

Météorologie et astronomie


-  2601 cloud
-  2607 lightning
-  2602 umbrella
-  2614 umbrella with rain drops
-  2608 thunderstorm
-  2609 sun
-  263C white sun with rays
-  2600 black sun with rays
-  263D first quarter moon
-  263E last quarter moon
-  263F Mercury
-  2640 female sign, Venus
-  2641 Earth
-  2642 male sign, Mars
-  2643 Jupiter
-  2644 Saturn
-  26E2 Uranus
-  2646 Neptune
-  2647 Pluto
-  26B3 Ceres
-  26B7 Chiron
-  2648 constellation Aries
-  2653 constellation Pisces
-  26C4 snowman without snow (light snow)
-  26C5 sun behind cloud (partly cloudy)
-  26C6 rain













-  26C7 black snowman (heavy snow)
-  26C8 thunder cloud and rain
-  1F300 cyclone, typhoon
-  1F301 foggy
-  1F302 closed umbrella
-  1F30C Milky Way
-  1F311 new moon
-  1F313 first quarter moon
-  1F315 full moon
-  1F317 last quarter moon
-  1F319 crescent moon
-  1F320 shooting star

Industrie alimentaire





-  2615 hot beverage
-  26FE restaurant
-  1F345 tomato
-  1F34A tangerine
-  1F34F green apple
-  1F354 hamburger
-  1F359 rice ball
-  1F35E bread
-  1F363 sushi
-  1F368 ice cream
-  1F36D lollipop
-  1F372 pot of food
-  1F377 wine glass
-  1F37C baby bottle

Santé et sureté











-  2620 skull and crossbones

-  2621 caution sign (curves ahead on road)
-  2622 radioactive sign (trefoil)
-  2623 biohazard sign
-  2624 caduceus
-  262E peace symbol
-  262F yin yang
-  267F wheelchair symbol
-  2695 staff of Asclepius
-  26A0 warning sign
-  26A1 high voltage sign
-  26D0 sliding car
-  26E8 hospital







Échec

-  2654 white chess king
-  2655 white chess queen
-  265D black chess bishop
-  265F black chess pawn

Sports, jeux

-  26BD soccer
-  26BE baseball
-  26F7 skier
-  1F3AE video game
-  1F3B3 bowling
-  1F3BD running shirt and sash
-  1F3C0 basketball
-  1F3C3 runner
-  1F3C6 trophy
-  1F3C9 rugby

Block des caractères d'illustration des caractères de contrôle

-  2400 symbol for null
-  2406 symbol for acknowledge
-  2408 symbol for backspace
-  2409 symbol for horizontal tabulation
-  240A symbol for line feed
-  240D symbol for carriage return

Box Drawing

Block Elements

Figures géométriques

△ 25B3 equilateral triangle

⚡ 22BF right triangle

◻ 2B1C square

◊ 2662 rhombus, diamond

▭ 25AD rectangle

⬠ 2B20 pentagon

⬡ 2B21 heagon

○ 25CB circle

◌ 2B2D ellipse

Dingbats

▶ 25BA Black Right-Pointing Pointer

Liens externes

- Unicode lookup (<http://www.fileformat.info/info/unicode/char/search.htm>) [[archive](#)]
- Unicode blocks (<http://www.fileformat.info/info/unicode/block/index.htm>) [[archive](#)]
- Input of all unicode 6.1 symbols (<http://farlingo.narod.ru/unimapboard/unimapboard.html>) [[archive](#)]

Quelques alphabets

La première fonction d'Unicode étant de permettre les textes de différentes langues, il permet naturellement de mettre en œuvre de nombreux alphabets.

S'il serait naïf de croire que chaque langue s'écrit avec un alphabet, les alphabets les plus connus sont les suivants:

Alphabets des langues latines

Il s'agit de l'alphabet utilisé par quasiment toutes les langues d'origine européenne. Il compte vingt-six lettres environ, bien que chaque langue peut disposer de ses propres variantes.

Majuscules

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Minuscules

a	b	c	d	e	f	g	h	i	j	k	l	m
n	o	p	q	r	s	t	u	v	w	x	y	z

Alphabet ga

Il s'agit d'un alphabet de vingt-six lettres utilisé dans la région Ghana Togo Nigéria.

Aa Bb Dd Ee Εε Ff Gg Hh Ii Jj Kk Ll Mm Nn Ìη Oo Ɔɔ Pp Rr Ss Tt Uu Vv Ww Yy Zz

Alphabet grec

Il s'agit d'un alphabet de vingt-quatre lettres, dont certaines présentent es variantes de graphies en fonction de leur position.

Lettre capitale Α Β Γ Δ Ε Ζ Η Θ Ι Κ Λ Μ Ν Ξ Ο Π Ρ Σ Τ Υ Φ Χ Ψ Ω

Lettre minuscule α β (var. β) γ δ ε ζ η θ ι κ λ μ ν ξ ο π ρ σ (var. ς) τ υ φ/ϕ χ ψ ω

Voir aussi le livre wikibooks: Grec_ancien/Alphabet

Alphabet russe

L'alphabet de la langue russe se compose normalement de trente-trois lettres, mais des variantes existent pour d'autres langues.

Capitale А Б В Г Д Е Ё Ж З И Й І К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ъ Ы Ь Ъ Э Ю Я Ѡ ѡ

Minuscule а б в г д е ё ж з и й і к л м н о п р с т у ф х ц ч ш щ ъ ы ь ѡ э ю я Ѡ ѡ

Alphabet phénicien

Lettre phénicienne (ʾāleph) (bēth) (gīmel) (dāleth) (hē) (zayin) (ḥēth) (ṭēth) (yōdh) (kaph) (lāmedh) (mēm) (nun) (sāmekh) (ʾayin) (pē) (rēš) (šin) (tāw) (wāw) origine discutée (ʾayin)

Alphabet japonais

ん、ン /n/	わ ワ <i>wa</i> /ɰa/	ら ラ <i>ra</i> /ɾa/	や ヤ <i>ya</i> /ja/	ま マ <i>ma</i> /ma/	は ハ <i>ha</i> /ha/	な ナ <i>na</i> /na/	た タ <i>ta</i> /ta/	さ サ <i>sa</i> /sa/	か カ <i>ka</i> /ka/	あ ア <i>a</i> /a/	voyelle /a/
	ゐ ¹ ヰ <i>wi</i> /ɰi/	り リ <i>ri</i> /ɾi/		み ミ <i>mi</i> /mi/	ひ ヒ <i>hi</i> /hi/	に ニ <i>ni</i> /ni/	ち チ <i>chi</i> /tɕi/	し シ <i>shi</i> /ɕi/	き キ <i>ki</i> /ki/	い イ <i>i</i> /i/	voyelle /i/
		る ル <i>ru</i> /ɾɯ/	ゆ ユ <i>yu</i> /jɯ/	む ム <i>mu</i> /mɯ/	ふ フ <i>fu</i> /ɸɯ/	ぬ ヌ <i>nu</i> /nɯ/	つ ツ <i>tsu</i> /tsw/	す ス <i>su</i> /sw/	く ク <i>ku</i> /kɯ/	う ウ <i>u</i> /ɯ/	voyelle /u/
	ゑ ¹ ヱ <i>we</i> /ɰe/	れ レ <i>re</i> /ɾe/		め メ <i>me</i> /me/	へ ヘ <i>he</i> /he/	ね ネ <i>ne</i> /ne/	て テ <i>te</i> /te/	せ セ <i>se</i> /se/	け ケ <i>ke</i> /ke/	え エ <i>e</i> /e/	voyelle /e/
	を ヲ <i>wo</i> /ɰo/	ろ ロ <i>ro</i> /ɾo/	よ ヨ <i>yo</i> /jo/	も モ <i>mo</i> /mo/	ほ ホ <i>ho</i> /ho/	の ノ <i>no</i> /no/	と ト <i>to</i> /to/	そ ソ <i>so</i> /so/	こ コ <i>ko</i> /ko/	お オ <i>o</i> /o/	voyelle /o/

Alphabet N'ko

Voyelles

o	ô	ou	è	i	é	a
/ɔ/	/o/	/u/	/ɛ/	/i/	/e/	/a/
𞤎	𞤏	𞤐	𞤑	𞤒	𞤓	𞤔
𞤕	𞤖	𞤗	𞤘	𞤙	𞤚	𞤛

Consonnes

ra	da	tcha	dja	ta	pa	ba
/ra/	/da/	/tʃa/	/dʒa/	/ta/	/pa/	/ba/
𞤜	𞤝	𞤞	𞤟	𞤠	𞤡	𞤢
𞤣	𞤤	𞤥	𞤦	𞤧	𞤨	𞤩
ma	la	ka	fa	gba	sa	rra
/ma/	/la/	/ka/	/fa/	/g̃ba/	/sa/	/ra/
𞤪	𞤫	𞤬	𞤭	𞤮	𞤯	𞤰
𞤱	𞤲	𞤳	𞤴	𞤵	𞤶	𞤷

n'		ya	wa	ha	na	nya
/ŋ/		/ja/	/wa/	/ha/	/na/	/ɲa/
ᵑ		ɟ	ɓ	ɥ	ŋ	ɟ͡ɲ
ᵑ		ɟ	ɓ	ɥ	ŋ	ɟ͡ɲ

Alphabet tchèque

L'alphabet tchèque comprend les lettres suivantes: a á b c č d d' e é ě f g h ch i í j k l m n ň o ó p q r ř s š t ť u ú ů v w x y ý z ž ^[1].

Alphabet thaï

Consonnes: Lettre ก ข ฃ ค ฅ ฌ ง จ ฉ ช ซ ฌ ญ ฎ ฏ ฐ ฑ ฒ ด ต ถ ท ธ น บ ป ผ ฝ พ ฟ ภ ม ย ร ล ว ศ ษ ส ห ฬ อ ฮ

Voyelles กะ กั กา กิ กี้ ก็ กือ กุ กู เกะ เก แกะ แก เกอะ เกอ โกะ โก เกาะ กอ* เกียะ เกีย เกือะ เกือ กัวะ กัว กวก* ก่า ไก ไก เกา ฤ ฤา ฤ ฤา กร กรร* กรรร*

Alphabet braille

⠁ ⠃ ⠉ ⠇ ⠋ ⠊ ⠌ ⠍ ⠎ ⠏ ⠑ ⠒ ⠓ ⠔ ⠕ ⠖ ⠗ ⠘ ⠙ ⠚

Références et notes

1. Lire aussi le wikilivre Tchèque

Exemples de textes

L'écriture est un système de représentation graphique d'une langue, au moyen de signes inscrits ou dessinés sur un support, et qui permet l'échange d'informations sans le support de la voix.

漢 汉 ABC देवनागरी אֱלֶפְבֵּית ひカ
 字 字 αλφάβητο Кириллица أبجدية らタ
 ๐๑๒๓๔๕๖๗๘๙ ไทย ၀၁၂၃၄၅၆၇၈၉ 한글 なナ

Cette définition est bien théorique. Rien de tel pour l'illustrer que de montrer comment on pourrait donner une description de l'écriture, en l'écrivant dans une langue tierce:

- papel, pedra, madeira, arxila ou calquera outro material.
- 書寫係搵滴子符號、標記去記錄一隻語言嗰方式。
- Kirotaminõ om keelelidse sõnõmi edesiandminõ teksti kujol, midä saa nätäq vai kumpiq.
- 쓰기는 특정한 구문을 이용하여 의도적으로 무언가를 적는 것을 말한다.
- भारत में लेखन ३३०० ईपू का है। सबसे पहले की लिपि सरस्वती लिपि थी, उसके पश्चात ब्राह्मी आई।
- Pisanje je ljudska aktivnost kojom se na nečemu ostavlja trag u vidu simbola (slova) iz skupa unaprijed utvrđenih znakova (pisma).
- כתיבה היא הפעולה של רישום סימני גראפיים בעלי משמעות כגון אותיות, מספרים, מילים ומשפטים על מצע כלשהו: דף נייר, לוח חרס, בד וכדומה. הכתיבה נעשית על ידי שימוש בכלי כתיבה ובעידן הטכנולוגי היא יכולה להיעשות גם באמצעות מחשב או מכונת כתיבה, על ידי הקלדה על גבי מקלדת ומקרנת על גבי מסך.
- Ekri, Mete plizyè lèt ansanm pou fòmè mo. Mete plizyè mo ansanm pou fòmè fraz.
- Írásnak nevezzük azt az egyezményes karakterekkel írt jelsort, amelynek rögzített értelmezése, szabályos és egyértelmű kiolvasása létezik.
- Menulis adalah suatu kegiatan untuk menciptakan suatu catatan atau informasi pada suatu media dengan menggunakan aksara.
- Skriburo esas la vidanta ekspreso di pensajo e parolajo.
- Skrift er sú athöfn að skrá niður bókstafi í þeim tilgangi að mynda orð og setningar, eða skrá niður upplýsingar á annan hátt.
- La scrittura è la rappresentazione grafica della lingua per mezzo di un insieme di segni detti grafemi che compongono un sistema di scrittura.
- 筆記(ひっき、英語: writing)とは、言葉を意図的に選択し、特定の構文を用いて、何かを書き記すことである。それを記録するための道具や手段の選択肢は無限であるといってもよい。
- დამწერლობა — გრაფიკულ ნიშანთა, ან გამობატულებათა მეშვეობით სამეტყველო ინფორმაციის ფიქსაციის საშუალება.
- ഭാഷയെ ഒരു കൂട്ടം ചിഹ്നങ്ങളോ (പ്രതീകങ്ങളോ) (ഇതിനെ ആലേഖനവ്യവസ്ഥ എന്നുവിളിക്കുന്നു) ഉപയോഗിച്ച് രേഖപ്പെടുത്തുന്നതാണ് എഴുത്ത്.
- Tira yussa'd deg we myag aru, tella ugar n 5300 iseggasen aya. tira d'a nagraw n we beggen n tutlayt.
- Жазу — адамның ой-пікірін, басқа адамға хабарлап айтқысы келген сөзін, мағлұматын таңбалар арқылы жеткізуді қамтамасыз ететін белгілер жүйесі.
- Scriptura est ars scribendi, actus commemorandi nuntios per modum oculorum, qui ab aliis legi potest.
- Rašymas - kalbos perteikimas teksto pavidalu, naudojant ženklų arba simbolių rinkinius, vadinamus rašto sistemomis.
- Ny soratra dia fomba fametrahana an-tsary ny volan'ny fiteny iray, amin'ny alalan'ny endri-tsoratra soratana na natao sary (eo amina taratasy, ohatra), hahafahana mifanakalo fampahalalana haingana kokoa ary tsy mila feo.
- Пишување е човечка активност со која на некој медиум се остава траг во вид на симболи (букви) од низа строго утврдени симболи (азбука).
- Penulisan boleh merujuk kepada dua kegiatan: menulis aksara pada perantara dengan tujuan untuk membentuk perkataan dan gagasan yang lain yang mewakili bahasa atau maklumat tercatat; dan

penciptaan bahan-bahan yang disampaikan melalui bahasa tulisan.

- La scritta ye ua technologie, criada i zambolbida storicamente nas sociedades houmanas, podendo ser globalmente caratelizada como la ocorrência de marcas nun suporte.
- Āmatlalcāyōtl ītōcā in āmatl cualli tlahcuilōliztli tlamatiliztli.
- Schrijven is het maken van originele tekst of de taalweergave via een medium door tekens of symbolen van een schrift te gebruiken.
- Skrift er eit samleomgrep brukt om teikn laga på ei flate som er meinte til å formidla språk.
- Skrivning er kunsten å formidle en tekst gjennom tegn og symboler.
- L'escritura es un sistema de representacion grafica d'una lenga per mejan de signes traçats sus un supòrt.

- Pismo – system umownych znaków, za pomocą których można utrwalić język mówiony.
- لکھائی کسے بولی نوں اکھریاں دی مورت ج لکھ کے دسن دا ناں اے۔
- Escrita ou grafia consiste na utilização de sinais (símbolos) para exprimir as ideias humanas.
- Qillqa nisqaqa ima qillqasqapas, qillqana p'anqapi (liwrupi, willay p'anqapi, chaski qillqapi icha huk willa ñiqipi, antañiqipi, siq'isqa rikch'akuna icha sanampakuna, willakunata, yachayta waqaychanapaq.
- Un sistem de scriere este un sistem de simboluri și reguli de combinare a acestora, folosit pentru a înregistra (scrie) pe un suport fizic enunțuri formulate într-o anumită limbă (numită astfel limbă scrisă), în așa fel încît cel care vede simbolurile scrise să le poată descifra (citi) fără ajutorul celui care le-a scris.
- Письменность — знаковая система, предназначенная для формализации, фиксации и передачи тех или иных данных (речевой информации и др.
- लिपिः वर्णानां लेखनं लिपिः भवति । नाम अक्षराणि लेखितुं यः विन्यासः उपयुज्यते सा लिपिः । अस्याः लिपी इत्यपि कथयन्ति ।
- Pismo je način grafičkog predstavljanja odnosno zapisivanja jezika. Nauka koja se bavi izučavanjem njegovog nastanka, vrsta i razvoja naziva se gramatologija.
- Písmo má v jazykovede viacero definícií, konkrétne ich možno zhrnúť takto
- Shkrimi është paraqitja pamore e sendeve dhe mendimeve të shprehura në një gjuhë, duke përdorur shkronjat apo shenja të tjera.
- Saad bee al'ijhni éi saad bik'é'alchi'go naaltsoos bikáa'gi saad shijaago oolyé'. Akwii shijh al'aah át'éego saad bee al'ijhni' hóló.
- Eune manniéthe d'êcrithe (audgo manniéthe d'êcrithe) est un mouoyen écrit d'èrprésenter eune langue.
- Писмо је начин графичког представљања односно записивања језика.
- En skrift eller ett skriftsystem är en form av symbolsystem för att representera språkliga uttryck.
- Язу — теге яки бу мэгълүматларны формалаштыру, үзләштерү һәм житкерү өчен кулланылган знаклар системасы. Язу — тел яшәеше формаларының берсе.
- ระบบการเขียน (อังกฤษ: writing system) คือลักษณะสัญลักษณ์หรือตัวอักษรที่ใช้ในการแสดงความหมายที่ใช้ในภาษาต่างๆ ระบบการเขียนแตกต่างจากระบบสัญลักษณ์ทั่วไปคือ บุคคลที่ใช้ระบบเดียวกันสามารถอ่านและเข้าใจภาษานั้นได้ตรงกันโดยไม่จำเป็นต้องมีความรู้เฉพาะทางสำหรับตีความหมายของสัญลักษณ์นั้น ต่างกับสัญลักษณ์ใน ภาพวาด แผนที่ ป้าย คณิตศาสตร์และสัญลักษณ์ต่างๆ
- Bir yazı sistemi dildeki unsurları ve tarif edilebilir durumları temsil etmek için kullanılan bir çeşit semboller sistemidir.
- Писémність — засіб передачі людської мови за допомогою знаків (див. алфавіт, абетка, ієрогліфіка, клинопис); також література, сукупність писемних пам'яток певного часу, певного народу; письменство.
- زبان کو تحریر میں لانے کے لئے استعمال ہونے والی ترکیب کو رسم خط کہتے ہیں۔ اُردو کے رسم خط کا نام نستعلیق ہے۔
- Iton sistema hin panurat in uska sistema hin mga tigaman biswal nga nahipapatik ha papel o ha iba nga medyum, nga gingagamit hit pagrepresentar hin mga elemento nga nahipapagawas ha linggwahe.
- 原始文字是人类用来纪录特定事物、简化图像而成的书写符号。
- 文字，以紋誌事也。蓋文明肇興，社稷初立，人事日多，遂思誌事之法。或以圖畫，或以結繩。終有賢士，定紋飾以繫各事，乃有文字。
- Bùn-jī hē-thóng sī iōng tô·-hêng lâi kì-lok gí-giân ê hē-tóng.
- 文字就係符号嘅一種，用嚟表示概念、記錄口語、事、物，用作人同人溝通嘅一種媒介。

Saisie des caractères

L'objet de ce livre n'est pas d'indiquer comment saisir des caractères sur tel ou tel système, mais bien de présenter Unicode.

Toutefois, un livre intitulé Unicode en pratique serait sans doute incomplet s'il n'offrait quelques éléments d'information sur la saisie de caractères Unicode.

Avant toute chose, il est utile de préciser que tout utilisateur d'ordinateur saisit déjà des caractères Unicode chaque fois qu'il tape sur son clavier : par exemple l'utilisateur francophone saisit des caractères Unicode latin tandis que l'utilisateur grec saisit des caractères grecs.

Les questions qui se posent sont donc d'une part de savoir comment configurer son clavier pour des langues étrangères, et d'autres part comment saisir les caractères biens pratiques qui ne sont pas disposés sur le clavier. Par exemple, il n'est pas toujours pratique de saisir un accent grave sur un A majuscule.

Configuration du clavier

Pour utiliser les caractères des différents alphabets, une configuration du clavier peut être suffisante. La configuration du clavier est propre à chaque système, notamment Windows, ou X-Window sous Unix et MacOSX.

Méthode d'entrée

Une **méthode de saisie** (en anglais, *input method* en général ou *input method editor* (IME) chez Microsoft) est un programme ou un composant d'un système d'exploitation qui permet aux utilisateurs d'un ordinateur de saisir des caractères complexes et des symboles (tels que les caractères chinois, coréen, japonais ou d'origine Indiennes (Sanskrit, Tamoul, Tibétain...), à l'aide d'un clavier occidental classique. Le terme *input method environment* est également employé en anglais.

Le terme *Input Method* est généralement utilisé (Mac OS, BeOS, X Window System, terminal texte Unix...).

Microsoft utilise d'autres noms : Le terme **IME** est plutôt employé dans le contexte de Microsoft Windows. et *FEP* pour MS-DOS.

XIM est une infrastructure pour les méthodes d'entrée sous X Window System.

Utilisation sous Linux et Unix

- Sous Mac OS X et dans Mac OS 8.5 et suivants : il faut choisir la méthode de saisie *Unicode Hex Input*. La combinaison se fait en pressant la touche Option et en saisissant les 4 chiffres hexadécimaux du code point Unicode^[1].
- Sous l'environnement logiciel GNOME, maintenir la touche U tout en tapant le nombre Unicode. Les anciennes versions nécessitent de maintenir *Ctrl* et *Maj* en plus de la touche U.
- Accessoirement, et pour un public averti, dans l'éditeur de texte Vim, la combinaison `Ctrl-V u,` puis nombre hexadécimal, permet de saisir un caractère Unicode.

Logiciel de consultation

Certains logiciels sont dédiés à la consultation des caractères Unicode. Ils ne permettent pas de saisir un flot de texte comme cela se fait avec un clavier où l'on peut saisir plusieurs caractères par seconde, mais ils

offrent d'autres possibilités en ne se limitant pas à un sous-ensemble restreint des caractères d'Unicode.

C'est notamment le cas du logiciel graphique `gucharmap` et du logiciel en ligne de commande `unicode`.

gucharmap

`gucharmap` est un logiciel qui permet d'afficher les tables de caractères Unicode, de les rechercher, de les voir avec différents niveaux de zoom, et de les copier.

Un onglet dédié permet également de détailler le nom du point de code, son codage, le caractère, la catégorie et le sens d'écriture.

unicode

`Unicode` est un logiciel qui permet d'obtenir à partir de son numéro ou d'une chaîne de caractère elle-même, toutes les principales informations relatives à ce caractère, comme illustré dans l'exemple ci-après :

Invocation du logiciel pour le point de code U+1234:

```
unicode U+1234
```

Réponse du logiciel détaillant le nom du point de code, son codage, le caractère, la catégorie et le sens d'écriture:

```
U+1234 ETHIOPIC SYLLABLE SEE
UTF-8: e1 88 b4  UTF-16BE: 1234  Decimal: &#4660;
ሴ
Category: Lo (Letter, Other)
Bidi: L (Left-to-Right)
```

Autre exemple:

```
unicode ≠
U+2260 NOT EQUAL TO
UTF-8: e2 89 a0  UTF-16BE: 2260  Decimal: &#8800;
≠
Category: Sm (Symbol, Math)
Bidi: ON (Other Neutrals)
Character is mirrored
Decomposition: 003D 0338
```

Saisie de caractères par les développeurs

Les informaticiens disposent d'un autre moyen pour désigner des caractères lorsqu'ils écrivent un logiciel.

Ce moyen est dépendant du langage informatique utilisé. Mais le langage HTML comme la plupart des langages de programmation moderne permettent de saisir des caractères en les désignant par leur numéro



Quelques-unes des méthodes de saisie gérées par le logiciel libre SCIM utilisé dans l'environnement X Window

décimal et/ou hexadécimal.

Exemples:



Cette section est vide, pas assez détaillée ou incomplète.

Notes de bas de page

- anglais Taper des caractères spéciaux et accentués (<http://mac.sillydog.org/archives/001703.php>) [[archive](#)]

Glossaire

De par sa nature technique, cet ouvrage utilise un vocabulaire très spécifique. Le glossaire vise à rendre ce vocabulaire compréhensible du néophyte.

binaire Le système binaire est un système de numération utilisant la base 2. On nomme couramment bit (de l'anglais *binary digit*, soit « chiffre binaire ») les chiffres de la numération binaire positionnelle. Ceux-ci ne peuvent prendre que deux valeurs, notées par convention 0 et 1.

décimal Le système décimal est un système de numération utilisant la base dix. Dans ce système, les puissances de dix et leurs multiples bénéficient d'une représentation privilégiée. Il utilise les dix chiffres 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

hexadécimal Le système hexadécimal est un système de numération positionnel en base 16. Il utilise ainsi 16 symboles, en général les chiffres arabes pour les dix premiers chiffres et les lettres A à F pour les six suivants.

Unité de code L'unité de code est l'unité dans laquelle un système informatique stocke ou échange les données. Il s'agit généralement d'un multiplet de 8, 16 ou 32 bits, que l'on appelle suivant le cas, octet, byte, mot, entier.

Point de code Les caractères sont groupés en blocs en fonction de leur usage et des écritures supportées, et reçoivent une identification numérique unique appelée *point de code*, identifiée généralement sous la forme U+xxxx (où xxxx est un nombre hexadécimal de 4 à 6 chiffres, entre U+0000 et U+10FFFF). La plage définie permet d'attribuer jusqu'à 1 114 112 points de code.

Caractères et glyphes Un point de code unicode définit une entité abstraite comme la “lettre latine majuscule a” ou le “chiffre cinq bengali”. Sur l'écran ou le papier n'apparaissent que le glyphe, la représentation visuelle du caractère.

Le standard Unicode ne définit pas l'image des glyphes. Il définit seulement ce que les caractères signifient.

Texte, éléments de texte Le standard Unicode ne définit pas ce qu'est un texte ou un élément de texte. Il définit seulement les *encoded characters* aussi connu sous le nom de point de code. nombre allant de 0 à 10FFFF16 (hexadécimal)

Un élément textuel est représenté par une séquence de point de code.

Plain Text Le *Plain text* est une séquence de codes de caractères pure. Le *plain Unicode-encoded text* est donc une séquence de codes de caractères Unicode. Par opposition le texte stylé aussi connu sous le nom de texte riche est enrichi d'information relative à l'identification de la langue, à la taille de la police de caractères à la couleur aux liens hypertextes.

encoded characters Le lien (ou les liens) entre un caractère abstrait et les représentations encodées. Par exemple 0041 et 030A sont des formes encodées. A et À Å avant d'être affichés sont des abstractions. Elles peuvent être codées de différentes manières : 0041 et 030A, 00C5 ou 212B.

Autres lectures sur Unicode

Si cet ouvrage tente de rendre Unicode compréhensible de tous, vous y aurez peut-être trouvé des défauts ou des manques.

Vous aurez peut-être alors à vous tourner vers les lectures suivantes qui vous permettront peut-être de corriger le présent ouvrage.

Lectures papier

- Yves Desrichard. Bibliothèque et écritures, d'ASCII à Unicode. Editions du Cercle de la Librairie, 2009. ISBN 978-2-7654-0974-8
- Andries, Patrick. Unicode 5.0 en Pratique. Dunod, 2008. ISBN 978-2100511402
- Desgraupes, Bernard. Passeport pour Unicode. Paris: Vuibert informatique, 2005. ISBN 2-7117-4827-8
- André, Jacques. Unicode, écriture du monde? / Jacques André, Henri Hudrisier. Paris: Lavoisier, 2002. (Document numérique, v. 6, no. 3–4.) ISBN 2-7426-0594-7

«Application web»

- <http://unicode-table.com>

Lectures numériques

- Coder avec Unicode sur wikibooks
- Unicode et ISO 10646 en français (<http://hapax.qc.ca/>) [[archive](#)]
- Patrick Andries « Introduction à Unicode et à l'ISO 10646 », Document numérique 3/2002 (Vol. 6), p. 51-88.
URL : www.cairn.info/revue-document-numerique-2002-3-page-51.htm.
DOI : 10.3166/dn.6.3-4.51-88.
- Patrick Andries « Entretien avec Ken Whistler, directeur technique du consortium Unicode », Document numérique 3/2002 (Vol. 6), p. 329-351.
URL : www.cairn.info/revue-document-numerique-2002-3-page-329.htm.
DOI : 10.3166/dn.6.3-4.329-351.

- Document numérique 2002/3-4 (Vol. 6). 240 pages.
ISSN : 1279-5127
ISSN en ligne : 1963-1014.
Lien : <<http://www.cairn.info/revue-document-numerique-2002-3.htm>>.

Unicode en bibliothèque

- Unicode en bibliothèque (<http://www.mindmeister.com/fr/52872193/unicode-en-biblioth-que>) [[archive](#)]



Vous avez la permission de copier, distribuer et/ou modifier ce document selon les termes de la **licence de documentation libre GNU**, version 1.2 ou plus récente publiée par la Free Software Foundation ; sans sections inaltérables, sans texte de première page de couverture et sans texte de dernière page de couverture.

Récupérée de « https://fr.wikibooks.org/w/index.php?title=À_la_découverte_d%27Unicode/Version_imprimable&oldid=483909 »

Dernière modification de cette page le 14 juillet 2015 à 15:09.

Les textes sont disponibles sous licence Creative Commons attribution partage à l’identique ; d’autres termes peuvent s’appliquer.

Voyez les termes d’utilisation pour plus de détails.

Développeurs