sisco

Enfin, pour finir, il y a juste un point qui n'est pas clair dans ma tête et qui, sauf erreur de ma part, n'est pas expliqué (ou pas explicitement) dans le tuto (si c'est le cas en fin de compte, vraiment toutes mes excuses). Je le signale ici, peut-être que ça pourrait figurer dans le tuto. Avec l'encodage UTF-8, les caractères sont « numérisés » sur un nombre variable d'octets (ça peut être 1 octet pour « A » par exemple, mais ça peut être 2 octets pour « é » par exemple). Je n'ai pas réussi à comprendre comment un éditeur de texte qui utilise l'UTF-8 pour lire un fichier faisait pour savoir si 2 octets consécutifs correspondent à un seul caractère ou bien s'ils correspondent à 2 caractères (1 par octet) ? Étant donné que le nombre d'octet(s) pour un caractère donné est variable, je ne vois pas comment l'éditeur peut faire pour deviner comment faire le découpage de la séquence d'octets par caractère ? Si l'éditeur lit un octet il peut s'agir de l'encodage d'un seul caractère mais il peut aussi s'agir du début de l'encodage d'un caractère qui est encodé sur 2 octets (j'espère que je suis clair) ? C'est une question que je me suis posée en lisant ce tuto.

Maëlan

En fait, l'UTF-8 a été conçu pour que ça ne soit pas un problème. Son format exact (que je n'ai pas détaillé pour ne pas allonger trop ce cours) permet aux programmes qui lisent de l'UTF-8 de savoir le nombre d'octets que comporte un caractère ; cette indication est donnée par la valeur du premier octet :
− si elle est inférieure à 128, un seul octet (caractère ASCII) ;
− sinon, plusieurs octets (dont le nombre augmente avec la valeur de ce 1er octet.
UTF-8 permet même de savoir, en prenant un octet au hasard dans un texte, de savoir si c'est le premier octet d'un caractère ou un octet suivant, ce qui permet de se replacer au début du caractère (on parle d'« auto-synchronisation »). C'est ce que signifiait la « résistance aux erreurs » dans la liste de ses avantages.
Toutes ces propriétés se retrouvent d'ailleurs aussi dans l'UTF-16.

Pour plus de détails, je t'invite à consulter la page Wikipédia. :)

SpaceFox

C'est vrai que c'est pas expliqué, mais les octets de l'UTF-8 ont un système de préfixe, qui permet de dire, quel que soit l'octet lu, s'il a un sens seul ou si c'est le début d'une série de n octets.
Typiquement les octets qui codent un caractère seuls sont de la forme 0xxxxxxx, ceux qui sont au milieu ou à la fin d'une série de plusieurs sont de la forme 10xxxxxx, le premier a un préfixe différent selon le nombre d'octets supplémentaires qu'il faut lire après cet octet pour avoir le point de code complet.
Tous les détails ici : https://fr.wikipedia.org/wiki/UTF-8

sisco

Merci bien Maëlan et SpaceFox pour vos réponses. C'est très clair.

Perso, je trouve que le coup du « 0xxxxxxx (ie un code ASCII si j'ose dire) -> un seul octet

et 0xxxxxxx -> 2, 3 ou 4 octets » mériterait d'avoir une petite place dans le tuto. Même si ça reste une explication incomplète, ça permet d'avoir une idée du genre de tambouille qui permet gérer la taille variable des caractères à mon humble avis. Mais bon, c'est un détail.

Merci encore à vous deux pour le travail.
À+

Maëlan

Citation : sisco

> Perso, je trouve que le coup du « 0xxxxxxx (ie un code ASCII si j'ose dire) -> un seul octet et 0xxxxxxx -> 2, 3 ou 4 octets » mériterait d'avoir une petite place dans le tuto. Même si ça reste une explication incomplète, ça permet d'avoir une idée du genre de tambouille qui permet gérer la taille variable des caractères à mon humble avis. Mais bon, c'est un détail.

On ne peut pas non plus tout dire sur le sujet. Il y a un moment où il faut savoir s’arrêter. Le but est de sensibiliser le bas peuple et de lui offrir une vue synthétique, pas de faire une compilation de tout ce qu’il y a à savoir sur le thème (auquel cas le cours serait beaucoup, beaucoup plus long, et sans intérêt puisque l’on trouve déjà toutes les informations sur Internet). Le lecteur intéressé peut compléter en cherchant par lui-même, et je l’y encourage avec de nombreux liens (en insistant particulièrement sur Wikipédia qui contient tous les renseignements utiles).
En particulier, il y aurait énormément de choses à dire sur l’Unicode. La connaissance des spécifications exactes de l’UTF-8 (comme de l’UTF-16) n’est pas nécessaire pour comprendre ses caractéristiques principales, avantages et inconvénients, c’est pourquoi j’ai choisi de l’éluder.

D’ailleurs, comme je l’avais déclaré sur le sujet de la bêta-zone, j’ai dans l’idée de rédiger un cours spécifique axé sur l’Unicode. Ce n’est pas encore certain (il me faut encore définir précisément ce qu’il y aurait dedans), mais pour l’instant des passages que j’avais rédigé pour ce tuto-ci et finalement enlevés (dont les explications sur l’UTF-8) y attendent.

sisco

Pas de souci Maëlan, ton point de vue est parfaitement cohérent. Rien à dire.
Bon courage dans la rédaction de ton cours sur l'Unicode alors… :)
À+

Pic-Sou

En ce qui concerne l'UTF-8, je ne comprends pas non plus pourquoi les caractères non-latins mais qui tiennent sur un octet en ISO-8859-1 en occupent deux en UTF-8 : pourquoi par exemple l'espace insécable devient-elle C2-AB au lieu de AB (ou 00-AB en UTF-16) ?

tuomasi

bonjour,

tout d'abord je vais bien entendu te feliciter pour l'ecriture de ce tutoriel qui est long mais interressant et fluide alors merci.

Je ne voudrais pas abuser, mais j'aurais une question.

concretement comment fait-on pour parler avec son correspondant japonais, son amie chinoise ou encore des coreens par le biais de messenger ou tout autre chat?

comment fait-on pour passer du pinyin ou du romaji automatiquement vers les ideogrammes chinois ou les syllabaires japonais ou encore l'aphabet coreen?

merci d'avance.

Maëlan

@ tuomasi :
À vrai dire, je n’en ai aucune idée. Je ne pratique aucune langue asiatique. D’ailleurs, tu remarqueras que je parle moins des encodages asiatiques.*
En utilisant un encodage Unicode, tu pourras sans problème discuter en chinois ou en japonais, voire les deux dans la même discussion.

Je pense que l’encodage dépend du protocole de communication utilisé (MSN, IRC, etc.) et de ton client (le logiciel utilisé : MSN Messenger, Pidgin, etc.). Certains protocoles peuvent intégrer dans leurs spécifications l’encodage à utiliser, d’autres non et demander une information sur l’encodage dans les données, que ton client se charge d’écrire (auquel cas tu peux certainement configurer ton client pour choisir l’encodage)…
Par exemple, d’après cette page, le protocole MSN inclut un champ similaire à l’en-tête HTTP des pages web et des courriels, donc on devrait en théorie pouvoir encoder ses messages comme on le souhaite. Mais (d’après ce qui suit et cette page) seuls latin-1 et UTF-8 sont acceptés (ça pourrait être pire ^^) ; du moins, les serveurs officiels et le client officiel (MSN Messenger) ne supportent que ces deux-là.
Autre exemple : le protocole IRC ne spécifie pas d’encodage particulier (chacun encode donc comme il veut), et ne prévoit aucune information pour dire aux autres quel encodage on utilise. Ce qui fait un assez beau bazar dans une discussion quand deux personnes n’utilisent pas le même encodage. On se met donc d’accord sur un encodage (l’UTF-8 est un bon choix).

Je t’invite donc à rechercher par toi-même. Renseigne-toi sur le protocole, sur ton client… Cependant, si tu trouves la réponse à ta question, un retour d’expérience sera le bienvenu ici. :) Peut-être que je rédigerai une partie sur le sujet.

* Dans les encodages asiatiques importants, et dont je ne parle pas dans ce cours, on peut citer le GB 18030, dont le support est obligatoire en Chine. À la base, c’était un encodage pour les idéogrammes chinois, mais il a été étendu pour supporter tout caractère Unicode (mais c’est compliqué à gérer parce qu’il n’y a pas de règle pour encoder un point de code

Unicode en GB 18030, il faut faire au cas par cas).

Pour ta question concernant le romaji et le pinyin, je ne sais pas non plus, ça doit se jouer au niveau de ton logiciel.
Édit : Tu peux trouver des outils en ligne qui font le remplacement. Par exemple, pour l'arabe, j'utilisais ça (enfin, une version du script bidouillée par mes soins ^^), le même site propose d'autres langues dont le japonais (avec support du romaji pour conversions avec hiragana et katakana) et le chinois (dont le pinyin). Si ça correspond à que tu souhaites.

tuomasi

Merci pour ta reponse maellan.
en fait il y a deux problemes:
-le fait que l'ecriture asiatique en general compte beaucoup de caracteres donc logiquement il ne peuvent etre stockés dans UTF-8 car il ne supporte que 2^8=256 possibilités, il faut donc partir sur (au moins) UTF-16 2^16=65536 car il existe environ 15000 caractere chinois plus les deux syllabaires japonais et les autres alphabet cyrillique arabe coréens etc.

-la deuxieme chose a savoir et que en chinois (comme dans la plupart des langues autre que celle latine)
il existe un syteme de transcription basee sur la phonetique qui est pratiquée depuis 1958.
chaque caractere correspond a une seule syllabe en pinyin.
donc en fait tu ecrit du pinyin sur ton clavier et comme par magie les caracteres apparaissent a la place par exmple dans MSN(j'ai deja vu des chinois le faire sur leur pc)

mais en tant qu'occidental je ne vois pas comment on peut faire,
alors il paraît qu'il faut installer des polices de caracteres mingliu pour le chinois etc, mais je n'ai jamais reussi a ce que cela fonctionne.

voila je vous tiens au courant si j'ai du neuf ;)

Maëlan

Citation : tuomasi

> en fait il y a deux problemes:
> -le fait que l'ecriture asiatique en general compte beaucoup de caracteres donc logiquement il ne peuvent etre stockés dans UTF-8 car il ne supporte que 2^8=256 possibilités, il faut donc partir sur (au moins) UTF-16 2^16=65536 car il existe environ 15000 caractere chinois plus les deux syllabaires japonais et les autres alphabet cyrillique arabe coréens etc.

Non non, relis ce cours, UTF-8 permet bien de représenter n'importe quel caractère Unicode, il a été conçu pour ça. Le « 8 » dans son nom indique la taille **minimale** (8 bits = 1 octet) d'un caractère dans cet encodage, mais les caractères peuvent être encodés avec plus d'octets.

Citation : tuomasi

> -la deuxieme chose a savoir et que en chinois (comme dans la plupart des langues autre que celle latine) il existe un syteme de transcription basee sur la phonetique qui est pratiquée depuis 1958.
> chaque caractere correspond a une seule syllabe en pinyin.
> donc en fait tu ecrit du pinyin sur ton clavier et comme par magie les caracteres apparaissent a la place par exmple dans MSN(j'ai deja vu des chinois le faire sur leur pc) mais en tant qu'occidental je ne vois pas comment on peut faire,
> alors il paraît qu'il faut installer des polices de caracteres mingliu pour le chinois etc, mais je n'ai jamais reussi a ce que cela fonctionne.

Il s'agit peut-être d'une méthode de saisie (ou d'un pilote), donc due au système d'exploitation (à moins que ce remplacement soit spécifique au logiciel ?).
Par exemple, chez moi sous **Windows** (XP) :
> Panneau de Configuration
> « Options régionales et linguistiques »
> onglet « Langues »
> bouton Détails… dans la section « Services de texte et langues d'entrée » (regarde aussi la section « Prise en charge de langues supplémentaires » pour t'assurer que les langues asiatiques sont installées).
> Ensuite, cherche s'il y a des choses intéressantes, essaie (je ne peux pas t'aider, étant donné que je ne l'ai pas fait moi-même)… Par exemple, dans l'onglet « Avancé », décoche la case « Arrêter les services de texte avancés » (ça semble être en rapport, et il y a une mention à l'Asie de l'Est dans la description). dans l'onglet « Paramètres », j'imagine aussi qu'il faut que tu rajoutes la langue que tu veux (japonais, chinois, etc.) dans les langues d'entrée.

Au pire, demande à des Chinois (si tu en as sous la main) comment ils font.

Adilh

@tuomasi

Pour le chinois, il y a deux façon (Wubi), compliqué à expliquer et à utiliser pour un non-chinois. et sinon tu peux écrire en Pinyin avec un clavier normal, il suffit d'installer le clavier Chinois sur ton ordinateur, quand tu tapes les mots en pinyin une fenêtre s'affiche avec des propositions de caractère.
(les clavier chinois sont des QWERTY, alors je sais pas si ça va bien marcher sur un clavier français)
- http://www.google.com/intl/zh-CN/ime/pinyin/

Pour le Coréen, le coréen a un alphabet, la méthode est donc différente, chaque touche est une lettre. cependant l'agencement des lettres est différentes donc Pour chaque combinaison de lettre, la méthode de saisie donne le caractère avec le bon agencement.
ㅋ+ ㅜ = 쿠 (ㅋ au dessus de ㅜ)
http://kpopchan.com/kr/src/131228926627.png

je ne connais pas vraiment le japonnais mais je crois que cela ce rapproche de la manière chinoise.

Fabrice g

bonjour!!! voilà je suis nouveau et je m'intéresse depuis peu à l'informatique. j'ai vraiment appris beaucoup avec les cours du site du zero. Mais j'ai une question qui reste toujours sans réponse a propos du code Ascii: Comment à partir de ce code l'ordi arrive à afficher le caractère à l'écran? de quel façon est stocké un A par exemple numeriquement? en espérant que vous comprendrez ma question. si quelqu'un pouvait m'aidé ce serai super sympas!!!! merci d'avance!

Maëlan

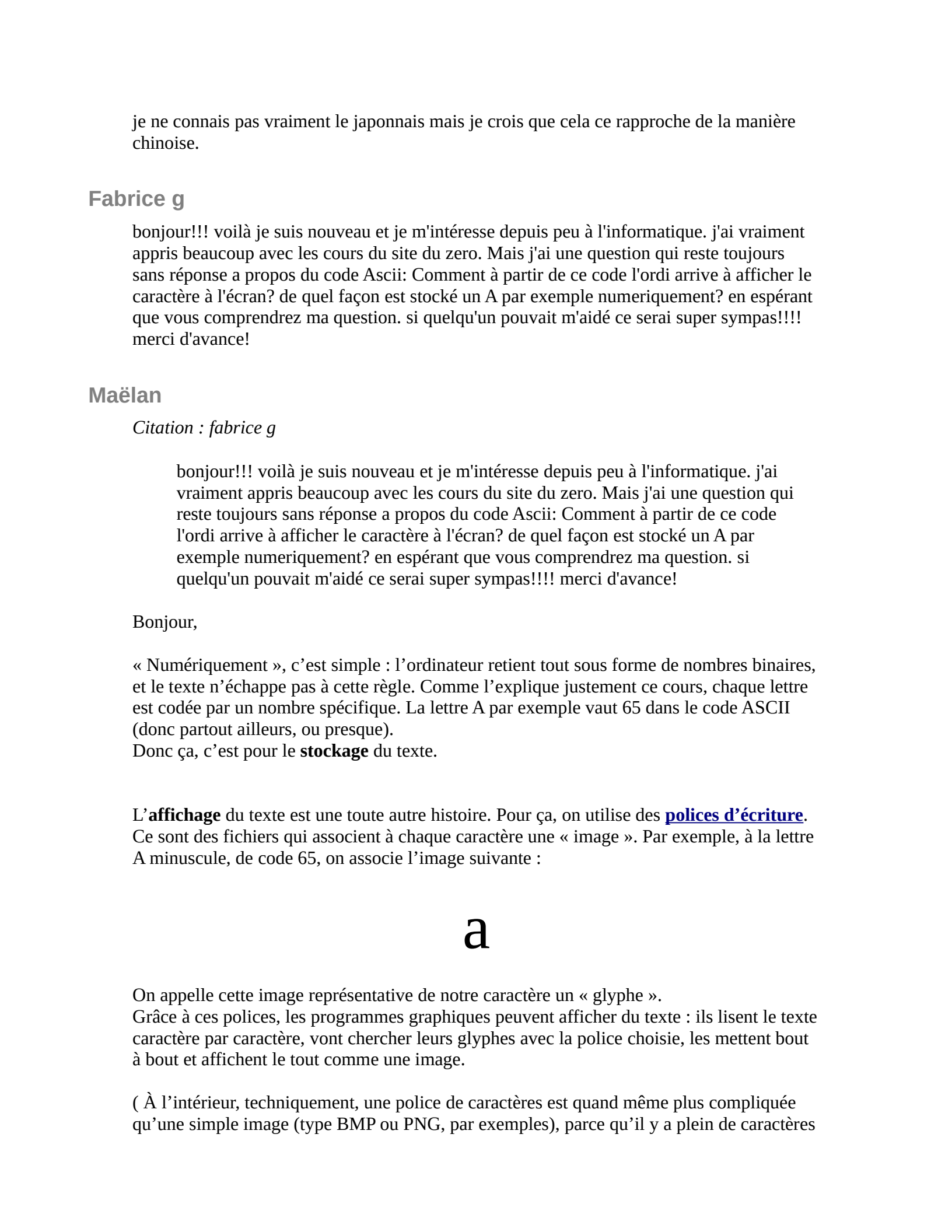
Citation : fabrice g

> bonjour!!! voilà je suis nouveau et je m'intéresse depuis peu à l'informatique. j'ai vraiment appris beaucoup avec les cours du site du zero. Mais j'ai une question qui reste toujours sans réponse a propos du code Ascii: Comment à partir de ce code l'ordi arrive à afficher le caractère à l'écran? de quel façon est stocké un A par exemple numeriquement? en espérant que vous comprendrez ma question. si quelqu'un pouvait m'aidé ce serai super sympas!!!! merci d'avance!

Bonjour,

« Numériquement », c’est simple : l’ordinateur retient tout sous forme de nombres binaires, et le texte n’échappe pas à cette règle. Comme l’explique justement ce cours, chaque lettre est codée par un nombre spécifique. La lettre A par exemple vaut 65 dans le code ASCII (donc partout ailleurs, ou presque).
Donc ça, c’est pour le **stockage** du texte.

L’**affichage** du texte est une toute autre histoire. Pour ça, on utilise des **polices d’écriture**. Ce sont des fichiers qui associent à chaque caractère une « image ». Par exemple, à la lettre A minuscule, de code 65, on associe l’image suivante :



On appelle cette image représentative de notre caractère un « glyphe ».
Grâce à ces polices, les programmes graphiques peuvent afficher du texte : ils lisent le texte caractère par caractère, vont chercher leurs glyphes avec la police choisie, les mettent bout à bout et affichent le tout comme une image.

(À l’intérieur, techniquement, une police de caractères est quand même plus compliquée qu’une simple image (type BMP ou PNG, par exemples), parce qu’il y a plein de caractères

différents à gérer, qu’il faut dire à quoi ils ressemblent pour différentes tailles et dif**férents** styles, qu’on doit pouvoir leur donner la couleur qu’on veut, sans parler d’autre détails comme la façon dont ils s’agencent les uns par rapport aux autres (par exemple, le p minuscule commence plus bas que le P majuscule.
En pratique, on utilise donc des polices de type « vectoriel », c’est-à-dire que l’image est décrite par ses formes et proportions, mais n’a pas une taille fixe (comme ça, on peut l’afficher avec la taille qu’on veut). De plus, on fournit des informations sur chaque glyphe : ses « métriques », c’est-à-dire des nombres pour décrire sa longueur, sa hauteur, sa position par rapport à la ligne d’écriture et à la lettre précédente, etc.)