C'est quoi l'ASCII, l'UNICODE, l'UTF-8 ?

L'ASCII

L'ordinateur est une grosse machine à calculer : tout ce qu'il sait faire, c'est effectuer des calculs sur des **nombres**. Il est incapable de comprendre le texte.

Il faut donc faire un choix : par quel nombre on prend pour représenter la lettre 'A' ? Et pour les signes de ponctuation, quels nombres utiliser ?

Il existe différentes conventions (ou codes). L'un des plus connus est le code **ASCII** (*American Standard Code for Information Interchange*). C'est un standard américain, mais c'est l'un des plus utilisés, en particulier sur la plupart des ordinateurs.

Le code ASCII définit précisément la correspondance entre symboles et nombres jusqu'au nombre 127:

0 NUL	32 espace	64 @	96 `
1 SOH	33 !	65 A	97 a
2 STX	34 "	66 B	98 b
3 ETX	35 #	67 C	99 c
4 EOT	36 $	68 D	100 d
5 ENQ	37 %	69 E	101 e
6 ACK	38 &	70 F	102 f
7 BEL	39 '	71 G	103 g
8 BS	40 (	72 H	104 h
9 HT	41)	73 I	105 i
10 LF	42 *	74 J	106 j
11 VT	43 +	75 K	107 k
12 FF	44 ,	76 L	108 l
13 CR	45 -	77 M	109 m
14 SO	46 .	78 N	110 n
15 SI	47 /	79 O	111 o
16 SLE	48 0	80 P	112 p
17 CS1	49 1	81 Q	113 q
18 DC2	50 2	82 R	114 r
19 DC3	51 3	83 S	115 s
20 DC4	52 4	84 T	116 t
21 NAK	53 5	85 U	117 u
22 SYN	54 6	86 V	118 v
23 ETB	55 7	87 W	119 w
24 CAN	56 8	88 X	120 x
25 EM	57 9	89 Y	121 y
26 SIB	58 :	90 Z	122 z
27 ESC	59 ;	91 [	123 {
28 FS	60 <	92 \	124 |
29 GS	61 =	93]	125 }
30 RS	62 >	94 ^	126 ~
31 US	63 ?	95 _	127 ■

Il faut donc utiliser le nombre 97 pour représenter un '**a**' minuscule. Pour représenter un '**?**', il faut utiliser le code 63.
Certains codes (ceux inférieurs à 32) sont des codes de contrôle (il ne sont pas faits pour être affichés). Par exemple le code 10 permet d'aller à la ligne, le code 7 fait biper l'ordinateur, etc.

Mais vous avez remarqué ? Il n'y a aucun caractère accentué ! Les américains nous ont oublié. Nous et d'autres pays : l'Espagne (avec le point d'interrogation retourné par exemple), l'Allemagne, etc.

Sans parler des pays comme la chine ou le japon avec leurs différents alphabet...

Il nous arrive souvent d'utiliser les codes de 128 à 255 pour les accents, mais ces codes sont différents d'un pays à l'autre ! Pas pratique pour échanger des documents.

Il faut donc trouver un code plus pratique. Il existe: c'est l'**UNICODE**.

L'Unicode

Au lieu d'utiliser les codes 0 à 127, il utilise les codes 0 à 65535 (en base 16 : de 0000 à FFFF).

Le code UNICODE permet de représenter tous les caractères spéficiques aux différentes langues. De nouveaux codes sont régulièrement attribués pour de nouveaux caractères: caractères latins (accentués ou non), grecs, cyrillics, arméniens, hébreux, thaï, hiragana, katakana...

L'Unicode définie donc un correspondance entre symboles et nombres.
(Le symbole "Ő" sera représent par le nombre 213).

Voici une toute petite partie des tables UNICODE (les nombres sont présentés en notation hexadécimal):

Caractères Unicode 0000 à 007F (0 à 127) (caractères latins)

Caractères Unicode 0080 à 00FF (128 à 255) (caractères latins, dont accentués)

Caractères Unicode 0900 à 097F (2304 à 2431) (caractères devanagari)

Caractères Unicode 1100 à 117F (4352 à 4479) (caractères hangul jamo)

Vous pourrez trouver plus d'informations sur l'UNICODE sur http://www.unicode.org.

Même si l'UNICODE est bien conçu, il reste assez peu utilisé par rapport à l'ASCII. (Ne vous amusez pas à envoyer un message en UNICODE à quelqu'un : il ne saurait probablement pas comment le lire !). Pour les programmeurs, ça n'est pas toujours très facile à manipuler non plus.

Ce standard se développe de plus en plus. Les langages **Java**, **.Net** (C#) et **Python** supportent déjà nativement l'UNICODE. La plupart des systèmes d'exploitation (Windows, Linux, MacOS X...) supportent déjà l'Unicode.

Unicode dans la pratique: UTF-8

Bon. Unicode, dans la théorie, c'est très bien.

Mais dans la pratique, c'est une autre paire de manches:
Généralement en Unicode, un caractères prend **2 octets**. Autrement dit, le moindre texte prend **deux fois plus de place qu'en ASCII**. C'est du gaspillage.

De plus, si on prend un texte en français, la grande majorité des caractères utilisent seulement le code ASCII. Seuls quelques rares caractères nécessitent l'Unicode.

On a donc trouvé une astuce: l'**UTF-8**.

Un texte en UTF-8 est simple: il est partout en ASCII, et dès qu'on a besoin d'un caractère appartenant à l'Unicode, on utilise un caractère spécial signalant "*attention, le caractère suivant est en Unicode*".

Par exemple, pour le texte "`Bienvenue chez Sébastien !`", seul le "**é**" ne fait pas partie du code ASCII. On écrit donc en UTF-8:

```
Bienvenue chez SÃ©bastien !
```

Pour être rigoureux, on indique quand même au début du fichier que c'est un fichier en UTF-8 à l'aide de caractères spéciaux:

```
i»¿Bienvenue chez SÃ©bastien !
```

Et voilà !

L'UTF-8 rassemble le meilleur de deux mondes: l'efficacité de l'ASCII et l'étendue de l'Unicode. D'ailleurs l'UTF-8 a été adopté comme norme pour l'encodage des fichiers XML. La plupart des navigateurs récents supportent également l'UTF-8 et le détectent automatiquement dans les pages HTML.

Alors dans les pages web, comment on fait ?

Si vous mettez directement le caractère "é" dans une page web, **ce n'est pas bien.** Il faut obligatoirement choisir une des 3 solutions suivantes:

- soit utiliser les entités HTML, et donc mettre **`&eacute;`** à la place de "**é**".
- soit laisser le "**é**" tel quel et préciser le charset que vous utilisez au début du fichier HTML (dans la balise <head>):
  `<meta http-equiv="Content-type" content="text/html; charset=ISO-8859-1">`
  (ISO-8859-1 est le jeu de caractère latin courant sous Windows.)
- soit travailler directement en UTF-8 dans votre éditeur HTML (s'il le permet). Ajoutez alors:
  `<meta http-equiv="Content-type" content="text/html; charset=UTF-8">`

L'ISO-8859-1 convient pour la plupart des langues latines ou occidentales (anglais, français, allemand, espagnol...), et l'UTF-8 vous sera indispensable pour les autres langues (japonais, hébreu, etc.).