

## Le codage numérique du texte: Da ASCII code ! (activité sans ordinateur).

Dominique Larrieu, Pr. de Mathématiques, Lycée Régional de Valbonne, exercices pour classes de 2<sup>nd</sup>.

Le code ASCII est l'un des plus anciens codes utilisés pour représenter du texte en informatique.

ASCII signifie (American Standard Code for Information Interchange ).

Il se base sur un tableau contenant les caractères les plus utilisés **en langue anglaise** : les lettres de l'alphabet en majuscule (de A à Z) et en minuscule (de a à z), les dix chiffres arabes (de 0 à 9), les signes de ponctuation (point, virgule, point-virgule, guillemet, parenthèses, etc.), quelques symboles et certains caractères spéciaux invisibles (espace, retour-chariot, tabulation, retour-arrière, etc.).

Les créateurs de ce code ont limité le nombre de ses caractères à 128, c'est-à-dire 2<sup>7</sup>, pour qu'il puissent être codés avec seulement 7 bits : les ordinateurs utilisaient des cases mémoire de un octet, mais ils réservaient toujours le 8<sup>e</sup> bit pour le contrôle de parité (c'est une sécurité pour éviter les erreurs, qui étaient très fréquentes dans les premières mémoires électroniques).

*Exemple* : Le caractère A est codé en ASCII par le nombre 65 (dans notre système décimal habituel), qui correspond en binaire au nombre 1000001.

Chaque caractère d'un texte codé en ASCII occupe alors un octet.

Un texte de 5000 caractères occupe donc 5 ko.

### Code ASCII étendu

Longtemps après, les mémoire devenant plus fiables et de nouvelles méthodes plus sûres que le contrôle de parité ayant été inventées, le code ASCII ne suffisait plus. En effet, il avait l'inconvénient très gênant de ne contenir que les lettres non accentuées, ce qui pouvait suffire en anglais, mais pas dans les autres langues (comme le français et l'espagnol par exemple). Il manquait aussi des caractères qui pouvaient être utiles comme ceux servant à fabriquer des bordures de tableaux (plus jolies que celles formées de points d'exclamation et de traits-d'union) etc.

C'est pourquoi le code ASCII fut étendu sur 8 bits, ce qui permit d'obtenir un tableau de 256 caractères (2<sup>8</sup>) au lieu de 128. Il apparut alors possible d'écrire des lettres comme é, è, ç, à, ù, ô, æ, œ, ñ, etc... Microsoft a adopté le code ANSI.

### Autres jeux de caractères

Le code *Unicode* est un système de codage des caractères sur 16 bits mis au point en 1991. Le système Unicode permet de représenter n'importe quel caractère par un code sur 16 bits, indépendamment de tout système d'exploitation ou langage de programmation. Il regroupe ainsi la quasi-totalité des alphabets existants (arabe, arménien, cyrillique, grec, hébreu, latin, ...) et est compatible avec le code ASCII.

---

### Activité :

La phrase : « Marie mange un abricot sec. » a une taille de 27 octets (il faut compter les espaces, l'apostrophe et le point final bien sûr).

L'expérience peut être faite en utilisant un éditeur de texte simple comme le bloc-notes de Windows. Il suffit d'écrire le texte, puis de l'enregistrer sous (observer les différents formats)et ensuite de vérifier la taille en octets du fichier obtenu (ce qui peut se faire en cliquant d'abord avec le bouton droit sur l'icône du fichier puis sur "Propriétés").

Maintenant tape cette phrase en utilisant la touche : Alt + (code du caractère en décimal) en utilisant la table des caractères ASCII de l'apage suivante.

## Table des caractères ASCII

caractère	code ASCII	code hexadécimal
NUL ( <i>Null</i> )	0	00
SOH ( <i>Start of heading</i> )	1	01
STX ( <i>Start of text</i> )	2	02
ETX ( <i>End of text</i> )	3	03
EOT ( <i>End of transmission</i> )	4	04
ENQ ( <i>Enquiry</i> )	5	05
ACK ( <i>Acknowledge</i> )	6	06
BEL ( <i>Bell</i> )	7	07
BS ( <i>Backspace</i> )	8	08
TAB ( <i>Tabulation horizontale</i> )	9	09
LF ( <i>Line Feed</i> , saut de ligne)	10	0A
VT ( <i>Vertical tabulation</i> , tabulation verticale)	11	0B
FF ( <i>Form feed</i> )	12	0C
CR ( <i>Carriage return</i> , retour à la ligne)	13	0D
SO ( <i>Shift out</i> )	14	0E
SI ( <i>Shift in</i> )	15	0F
DLE ( <i>Data link escape</i> )	16	10
DC1 ( <i>Device control 1</i> )	17	11
DC2 ( <i>Device control 2</i> )	18	12
DC3 ( <i>Device control 3</i> )	19	13
DC4 ( <i>Device control 4</i> )	20	14
NAK ( <i>Negative acknowledgement</i> )	21	15
SYN ( <i>Synchronous idle</i> )	22	16
ETB ( <i>End of transmission block</i> , fin de bloc de transmission)	23	17
CAN ( <i>Cancel</i> , annulation)	24	18
EM ( <i>End of medium</i> , fin du médium)	25	19
SUB ( <i>Substitute</i> , substitut)	26	1A
ESC ( <i>Escape</i> , caractère d'échappement)	27	1B
FS ( <i>File separator</i> , séparateur de fichier)	28	1C
GS ( <i>Group separator</i> , séparateur de groupe)	29	1D
RS ( <i>Record separator</i> , séparateur d'enregistrement)	30	1E
US ( <i>Unit separator</i> , séparateur d'enregistrement)	31	1F
SP ( <i>Space</i> , espace)	32	20
!	33	21
"	34	22
#	35	23
\$	36	24
%	37	25
&	38	26
'	39	27
(	40	28
)	41	29
*	42	2A
+	43	2B
,	44	2C
-	45	2D
.	46	2E
/	47	2F
0	48	30
1	49	31
2	50	32
3	51	33
4	52	34
5	53	35
6	54	36
7	55	37
8	56	38
9	57	39
:	58	3A
;	59	3B
<	60	3C
=	61	3D
>	62	3E
?	63	3F
@	64	40
A	65	41
B	66	42
C	67	43
D	68	44
E	69	45
F	70	46
G	71	47

H  
 I  
 J  
 K  
 L  
 M  
 N  
 O  
 P  
 Q  
 R  
 S  
 T  
 U  
 V  
 W  
 X  
 Y  
 Z  
 [  
 \  
 ]  
 ^  
 \_  
 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  
 {  
 |  
 }  
 ~  
 Touche de suppression

72	48
73	49
74	4A
75	4B
76	4C
77	4D
78	4E
79	4F
80	50
81	51
82	52
83	53
84	54
85	55
86	56
87	57
88	58
89	59
90	5A
91	5B
92	5C
93	5D
94	5E
95	5F
96	60
97	61
98	62
99	63
100	64
101	65
102	66
103	67
104	68
105	69
106	6A
107	6B
108	6C
109	6D
110	6E
111	6F
112	70
113	71
114	72
115	73
116	74
117	75
118	76
119	77
120	78
121	79
122	7A
123	7B
124	7C
125	7D
126	7E
127	7F