Unicode et UTF-8

L’Unicode

Sommaire

Unicode

- Normes et standards
- Avant l'Unicode
- Qu'est ce que l'Unicode
- Comment ca marche ?

UTF-8

- Qu'est ce que UTF
- UTF-32/16/8
- Passer de l'Unicode à l'UTF-8
- Un exemple de codage

Normes et standards

Mais qu'est ce que c'est ?

Une norme est un ensemble de règles de conformité ou de fonctionnement légiféré par un organisme mandaté pour (Comme l'ISO avec la norme ISO/CEI 10646).

Un standard est un ensemble de recommandations et de préférences données par certains groupes, qui ne sont pas issue d'un organisme mandaté pour et qui ne seront pas forcément respecter.

Avant l'Unicode

Avant l'Unicode, les tables de caractères (ou pages de codes) étaient définis à un niveau national avec comme base commune la norme ASCII. Sur les machines 8 bits de l'époque, un caractère est codé sur un octets, soit 256 possibilités, l'ASCII occupe les caractères de 0 a 127 et nous avions des significations différents pour les caractères 128 à 255, car aucune norme n'existait, pouvant entrainer des conflits de caractères en fonction du pays.

Des conflits avant l'Unicode

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1x	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2x	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

Page de code 850 (DOS Latin 1) — mode standard[1]

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8x	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
9x	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	ø	£	Ø	×	ƒ
Ax	á	í	ó	ú	ñ	Ñ	ª	º	¿	®	¬	½	¼	¡	«	»
Bx	░	▒	▓	│	┤	Á	Â	À	©	╣	║	╗	╝	¢	¥	┐
Cx	└	┴	┬	├	─	┼	ã	Ã	╚	╔	╩	╦	╠	═	╬	¤
Dx	ð	Ð	Ê	Ë	È	ı	Í	Î	Ï	┘	┌	█	▄	¦	Ì	▀
Ex	Ó	ß	Ô	Ò	õ	Õ	µ	þ	Þ	Ú	Û	Ù	ý	Ý	¯	´
Fx	SHY	±	‗	¾	¶	§	÷	¸	°	¨	·	¹	³	²	■	NBSP

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1x	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2x	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

Page de code 852 (DOS Latin 2) — mode standard[1]

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8x	Ç	ü	é	â	ä	ů	ć	ç	ł	ë	Ő	ő	î	Ź	Ä	Ć
9x	É	Ĺ	Í	ô	ö	Ľ	ľ	Ś	ś	Ö	Ü	Ť	ť	Ł	×	č
Ax	á	í	ó	ú	Ą	ą	Ž	ž	Ę	ę	¬	ź	Č	ş	«	»
Bx	░	▒	▓	│	┤	Á	Â	Ě	Ş	╣	║	╗	╝	Ż	ż	┐
Cx	└	┴	┬	├	─	┼	Ă	ă	╚	╔	╩	╦	╠	═	╬	¤
Dx	đ	Đ	Ď	Ë	ď	Ň	Í	Î	ě	┘	┌	█	▄	Ţ	Ů	▀
Ex	Ó	ß	Ô	Ń	ń	ň	Š	š	Ŕ	Ú	ŕ	Ű	ý	Ý	ţ	´
Fx	SHY	˝	˛	ˇ	˘	§	÷	¸	°	¨	˙	ű	Ř	ř	■	NBSP

Qu’est ce que l’Unicode ?

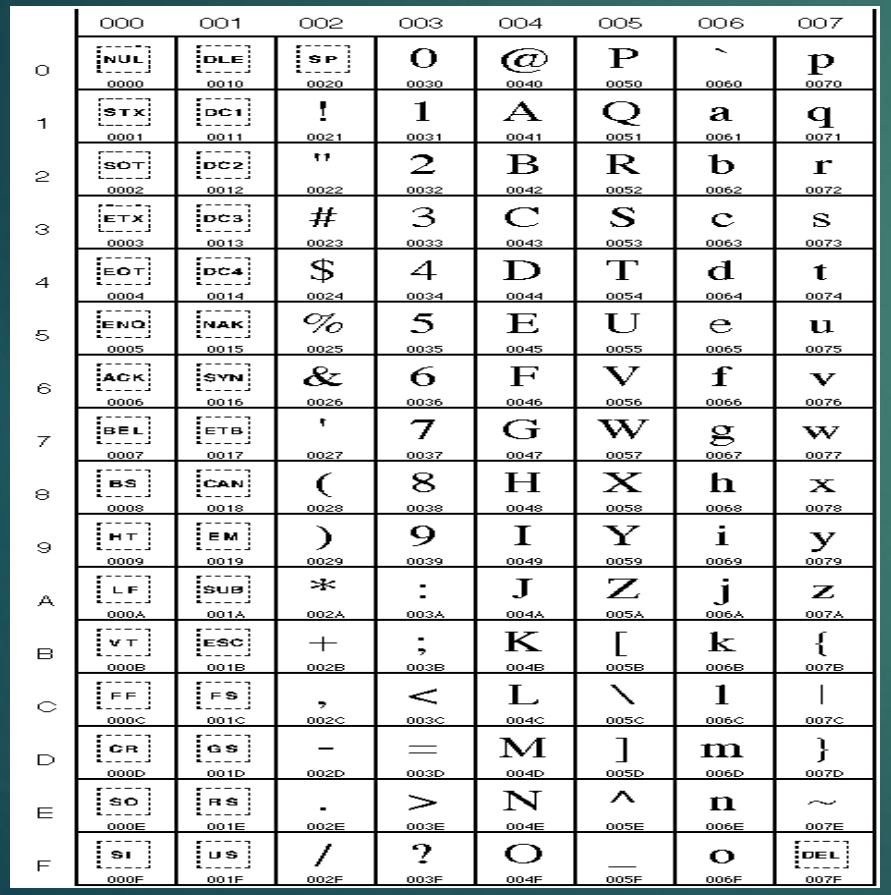
Une énorme base de données de caractères universelle, où chaque caractères possédait un seul et unique encodage (code en hexa type 06A5), allant de 0000 à FFFF. Un codage possible dans de nombreuses langues : caractères latins (accentués ou non), grecs, cyrilliques, arméniens, hébreux, thaï, hiragana, katakana…

Etant codé en hexadécimales, l’Unicode utilise donc 2 octets pour 1 caractères, contrairement à l’ASCII qui n’utilisait que 1 octet mais permet de coder ainsi jusqu’à 65 536 caractères contrairement a 256 auparavant rien qu’avec la première couche de l’Unicode.

Comment ca marche ?

Le codage d'un caractère sous Unicode reste relativement simple, il suffit d'utiliser une sorte de grille de code (page de codes) afin d'obtenir le caractère souhaité et son codage.

Un codage sous Unicode



	000	001	002	003	004	005	006	007
0	NUL 0000	DLE 0010	SP 0020	0 0030	@ 0040	P 0050	` 0060	p 0070
1	STX 0001	DC1 0011	! 0021	1 0031	A 0041	Q 0051	a 0061	q 0071
2	SOT 0002	DC2 0012	" 0022	2 0032	B 0042	R 0052	b 0062	r 0072
3	ETX 0003	DC3 0013	# 0023	3 0033	C 0043	S 0053	c 0063	s 0073
4	EOT 0004	DC4 0014	$ 0024	4 0034	D 0044	T 0054	d 0064	t 0074
5	ENQ 0005	NAK 0015	% 0025	5 0035	E 0045	U 0055	e 0065	u 0075
6	ACK 0006	SYN 0016	& 0026	6 0036	F 0046	V 0056	f 0066	v 0076
7	BEL 0007	ETB 0017	' 0027	7 0037	G 0047	W 0057	g 0067	w 0077
8	BS 0008	CAN 0018	(0028	8 0038	H 0048	X 0058	h 0068	x 0078
9	HT 0009	EM 0019	) 0029	9 0039	I 0049	Y 0059	i 0069	y 0079
A	LF 000A	SUB 001A	* 002A	: 003A	J 004A	Z 005A	j 006A	z 007A
B	VT 000B	ESC 001B	+ 002B	; 003B	K 004B	[005B	k 006B	{ 007B
C	FF 000C	FS 001C	, 002C	< 003C	L 004C	\ 005C	l 006C	| 007C
D	CR 000D	GS 001D	- 002D	= 003D	M 004D	] 005D	m 006D	} 007D
E	SO 000E	RS 001E	. 002E	> 003E	N 004E	^ 005E	n 006E	~ 007E
F	SI 000F	US 001F	/ 002F	? 003F	O 004F	_ 005F	o 006F	DEL 007F

	304	305	306	307	308	309
0		ぐ 3050	だ 3060	ば 3070	む 3080	ゐ 3090
1	ぁ 3041	け 3051	ち 3061	ぱ 3071	め 3081	ゑ 3091
2	あ 3042	げ 3052	ぢ 3062	ひ 3072	も 3082	を 3092
3	ぃ 3043	こ 3053	っ 3063	び 3073	や 3083	ん 3093
4	い 3044	ご 3054	つ 3064	ぴ 3074	や 3084	ゔ 3094
5	ぅ 3045	さ 3055	づ 3065	ふ 3075	ゆ 3085	か 3095
6	う 3046	ざ 3056	て 3066	ぶ 3076	ゆ 3086	け 3096
7	ぇ 3047	し 3057	で 3067	ぷ 3077	ょ 3087	

	304	305	306	307	308	309
8	え 3048	じ 3058	と 3068	へ 3078	よ 3088	
9	お 3049	す 3059	ど 3069	べ 3079	ら 3089	゙ 3099
A	お 304A	ず 305A	な 306A	ぺ 307A	り 308A	゚ 309A
B	か 304B	せ 305B	に 306B	ほ 307B	る 308B	゛ 309B
C	が 304C	ぜ 305C	ぬ 306C	ぼ 307C	れ 308C	° 309C
D	き 304D	そ 305D	ね 306D	ぽ 307D	ろ 308D	ゝ 309D
E	ぎ 304E	ぞ 305E	の 306E	ま 307E	ゎ 308E	ゞ 309E
F	く 304F	た 305F	は 306F	み 307F	わ 308F	ゟ 309F

L’UTF-8

Qu’est ce que UTF ?

Avec l’immensité , du répertoire de caractère qu’est Unicode , l’idée de pouvoir coder tout les caractère par un seul octet est abandonné, il est crée une norme , l’UTF : Unicode Transfer Format .

Il existe sous plusieurs tailles : UTF-32 , UTF-16 et UTF-8.

UTF-32

Il permet de représenter vraiment tout les caractère possible sur Unicode (1 114 112)

Il consiste simplement a coder sur 32 bits /4 octets, le numéro du caractère. Si sous d'autre format on peut aussi obtenir un caractère sur 32 bit , ici il est fixe . Si il est simple à comprendre , c'est aussi celui qui prend le plus de place , car certain caractères sont codés sur moins d'octets.

UTF-16

On choisit un module de deux octets, soit 16 bits : on utilise un seul module pour tous les caractères du BMP (les caractères « multilingue ») et deux modules (quatre octets) pour ceux des autres plans. Plus compliqué, mais plus économique en place, surtout si on sort peu du BMP.

UTF-8

On choisit un module d'un octet, soit 8 bits : on utilise un seul module pour les caractères ASCII (7 bits), et deux, trois ou quatre modules (octets) pour les autres. Nettement plus compliqué, mais encore plus économique en place, et surtout compatible avec l'ASCII.

De l'Unicode a l'UTF-8

- Le numéro de chaque caractère est donné par le standard Unicode.
- Les caractères de numéro 0 à 127 sont codés sur un octet dont le bit de poids fort est toujours nul.
- Les caractères de numéro supérieur à 127 sont codés sur plusieurs octets. Dans ce cas, les bits de poids fort du premier octet forment une suite de 1 de longueur égale au nombre d'octets utilisés pour coder le caractère, les octets suivants ayant 10 comme bits de poids fort.

De l'Unicode à l'UTF-8

Représentation binaire UTF-8	Signification
0xxxxxxx	1 octet codant 1 à 7 bits utiles
110xxxxx **10**xxxxxx	2 octets codant 8 à 11 bits utiles
1110xxxx **10**xxxxxx **10**xxxxxx	3 octets codant 12 à 16 bits utiles
11110xxx **10**xxxxxx **10**xxxxxx **10**xxxxxx	4 octets codant 17 à 21 bits utiles

Exemple de codage en UTF-8

Caractère	Numéro du caractère	Codage binaire UTF-8
A	65	**0**1000001
é	233	**110**00011 **10**101001
€	8364	**1110**0010 **10**000010 **10**101100
𒍅	119070	**11110**000 **10**011101 **10**000100 **10**011110