12-03-2015
Dernière modification : 03-06-2018

Les pages de codes (ASCII, Unicode, UTF-8...)

Une page de code est un standard informatique qui vise à donner un numéro (= point de code) à chaque caractère (= glyphe) d'une langue.

ASCII (1963)

Le code ASCII [*aski:*] (American Standard Code for Information Interchange) est la page de codes standard américaine utilisée aux débuts de l'informatique. Le code ASCII donne un numéro à chaque caractère jusqu'au nombre 127, cad que le codage se fait sur moins d' 1 octet, 7 bits précisément, d'où 2^7 = 128. Mais cette page de codes est limitée à la langue anglaise et ne présente aucun caractère accentué.

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 ■

ASCII

ISO-8859-1 (1986), ANSI, MacRoman etc

Pour arriver à 1 octet il reste 1 bit de disponible, donc au total 2^8=256 possibilités. Les codes 128 à 255 peuvent donc être utilisés pour les caractères accentués, c'est ce que propose l'ISO-8859-1.

ISO-8859-1																
	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
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
8x	PAD	HOP	BPH	NBH	IND	NEL	SSA	ESA	HTS	HTJ	VTS	PLD	PLU	RI	SS2	SS3
9x	DCS	PU1	PU2	STS	CCH	MW	SPA	EPA	SOS	SGCI	SCI	CSI	ST	OSC	PM	APC
Ax	NBSP	¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	-	®	¯
Bx	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
Cx	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
Dx	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
Ex	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
Fx	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

reprise ASCII

ISO 8859-1

Mais ANSI (American National Standards Institute) et MacRoman (Apple Macintosh) proposent d'autres associations pour les codes 128 à 255, donc ces pages de codes ne sont pas standard et posent des problèmes d'interopérabilité. Un texte encodé en ISO-8859-1 apparaîtra avec des caractères incompréhensibles sur un ordinateur utilisant ANSI ou MacRoman.

Unicode (1991)

Unicode a pour vocation d'uniformiser le codage en donnant à tout caractère de n'importe quelle langue un identifiant unique au niveau mondial.

Unicode utilise des points de code jusqu'à 4 octets soit plus de $2^{4x8} = 2^{32}$ = 4,3 milliard de possibilités ! En 2015 Unicode définit plus de cent mille caractères. Chaque caractère abstrait est identifié par un nom unique (un en anglais et un en français) et associé à un point de code (= position de code).
Remarque : l'alphabet Chinois Kanji comporte à lui seul 6 879 caractères.

Gros avantage de l'unicode : tous les caractères sont codés de façon unique et universelle.
Inconvénient : un caractère prend jusqu'à 4 octets soit 4 fois plus de place qu'en ASCII.

UTF-8 (1996)

UTF-8 (Unicode Transformation Format 8 bits) est dérivé de l'Unicode, et rétro-compatible avec la norme ASCII (codes de 0 à 127). Tout caractère ASCII est codé avec un seul octet, identique au code ASCII, alors que les autres caractères sont codés selon Unicode, sur 2 à 4 octets.

UFT-8 donne le moyen de différencier les codes ASCII des codes Unicode.
Pour ASCII, le bit de poids fort est nul et les 7 autres bits (surlignés en vert ci-dessous) donnent le code ASCII.
Pour Unicode, le bit de poids fort est non nul et suit la règle : 110, 1110, 11110... le reste (surligné en vert ci-dessous) étant le code Unicode.

Représentation binaire UTF-8	Signification
0xxxxxxx	1 octet codant 1 à 7 bits
110xxxxx 10xxxxxx	2 octets codant 8 à 11 bits
1110*0000* 10*1*xxxxx 10xxxxxx	3 octets codant 12 à 16 bits
1110*0001* 10xxxxxx 10xxxxxx	
1110*001x* 10xxxxxx 10xxxxxx	
1110*01xx* 10xxxxxx 10xxxxxx	
1110*10xx* 10xxxxxx 10xxxxxx	
1110*1100* 10xxxxxx 10xxxxxx	
1110*1101* 10*0*xxxxx 10xxxxxx	
1110*111x* 10xxxxxx 10xxxxxx	
11110*000* 10*01*xxxx 10xxxxxx 10xxxxxx	4 octets codant 17 à 21 bits
11110*000* 10*1*xxxxx 10xxxxxx 10xxxxxx	
11110*001* 10xxxxxx 10xxxxxx 10xxxxxx	
11110*01x* 10xxxxxx 10xxxxxx 10xxxxxx	
11110*100* 10*00*xxxx 10xxxxxx 10xxxxxx	

reprise ASCII

reprise Unicode

UTF-8

Ainsi, l'UTF-8 cumule à la fois les avantages de l'ASCII (taille des fichiers faible) et de l'Unicode (universalité des caractères).

UTF-16 et UTF-32

UTF-16 et **UTF-32** suivent le même principe que UTF-8 mais le codage se fait sur 16 bits (UTF-16) ou 32 bits (UTF-32) au minimum, au lieu de 8 bits.

Du coup, pour la plupart des langues latines qui utilisent abondamment les caractères ASCII, UTF-8 nécessite moins d'octets que UTF-16 ou UTF-32.
Par contre pour les langues utilisant beaucoup de caractères extérieurs à ASCII, UTF-8 occupe sensiblement plus d'espace. En effet les idéogrammes employés dans les textes de langues asiatiques comme le chinois, le coréen ou le japonais utilisent 3 octets en UTF-8, contre 2 octets en UTF-16.

UTF-32 sera plus efficace uniquement pour les textes utilisant majoritairement des écritures anciennes ou rares codées hors du plan multilingue de base, c'est-à-dire à partir de U+10000, mais il peut aussi s'avérer utile localement dans certains traitements pour simplifier les algorithmes, car les points de code y ont toujours une taille fixe, la conversion des données d'entrée ou de sortie depuis ou vers UTF-8 ou UTF-16 étant triviale.

Utilisation actuelle de UTF-8 et UTF-16

Windows. **UTF-16 est le standard dans Windows** (et donc NTFS) depuis Windows NT. Mais UTF-8 est totalement pris en charge depuis Windows 2000 et Windows XP.

UEFI et GPT. Comme Windows, c'est **UTF-16** qui a été choisi comme standard pour UEFI et GPT.

Mac. Avec l'avènement Mac OS X, le codage MacRoman a été remplacé par **UTF-8 en tant que codage par défaut sur les systèmes d'exploitation Macintosh**.
Remarque : le codage MacRoman inclut un glyphe correspondant à la pomme du logo Apple (point de code F0). Ce caractère n'existe pas en Unicode et doit donc posséder une correspondance dans la Zone d'Usage Privée. Apple utilise le point de code U+F8FF à cet effet.

Internet. Depuis 2003, c'est **l'UTF-8 qui est un des standards de l'Internet**. Il est utilisé par 82,2 % des sites web en décembre 2014.

Messagerie mail. A l'heure actuelle, **tous les logiciels de messagerie (mail) supportent UTF-8**.

Taille d'un fichier texte contenant 5 000 caractères, et sauvegardé dans différents formats :

	Caractères latins	Kanjis japonais
ANSI	5 ko	*non supporté*
Unicode	10 ko	10 ko
UTF-8	5 ko	15 ko

Cas de la console DOS (cmd.exe) et des fichiers Batch (.bat)

La console de commandes DOS ne gère pas Unicode, elle gère seulement les pages de codes locales ANSI, il faut donc enregistrer les fichiers batch en mode ANSI (un fichier batch enregistré en Unicode se ferme automatiquement dès son exécution). Par défaut, les commandes de la console utilisent la table **CP-1252** (Windows Latin 1 = « ANSI Windows par défaut »).

Mais sur un Windows version française, les programmes ne gérant pas Unicode sont réglés pour utiliser par défaut la table **CP-850** (MS-DOS Latin 1) :
→ Panneau de config > Région et langues > Administration > Langue pour les programmes non Unicode : Français (France)
→ C:\> chcp → Page de codes active : 850

Quand on rédige un fichier texte sous Notepad (mode ANSI) la table d'affichage utilise donc CP-850, mais quand la commande est passée à CMD.EXE, cette dernière l'affiche et l'exécute selon CP-1252. Par exemple, le caractère 'é' (130 en décimal dans la table CP-850) est affichée et exécutée en tant que caractère ',' (130 dans la table CP-1252) qui est nullement la virgule classique (44 en décimal) mais une autre sorte de virgule. Cette virgule n'étant pas comprise par l'interpréteur de commandes, elle échoue.

850 MS-DOS Latin 1

	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
0		0	@	P	`	p	Ç	É	á	▒	└	ð	Ó	
1	!	1	A	Q	a	q	ü	æ	í	▓	┴	Ð	ß	±
2	"	2	B	R	b	r	é	Æ	ó	■	┬	Ê	Ô	‗
3	#	3	C	S	c	s	â	ô	ú	│	├	Ë	Ò	¾
4	$	4	D	T	d	t	ä	ö	ñ	┤	─	È	õ	¶
5	%	5	E	U	e	u	à	ò	Ñ	Á	┼	ı	Õ	§
6	&	6	F	V	f	v	å	û	ª	Â	ã	Í	µ	÷
7	'	7	G	W	g	w	ç	ù	º	À	Ã	Î	þ	¸
8	(	8	H	X	h	x	ê	ÿ	¿	©	╚	Ï	Þ	°
9	)	9	I	Y	i	y	ë	Ö	®	╣	╔	┘	Ú	¨
A	*	:	J	Z	¡	z	è	Ü	¬	║	╩	┌	Û	·
B	+	;	K	[	k	{	ï	ø	½	╗	╦	■	Ù	¹
C	,	<	L	\	l	|	î	£	¼	╝	╠	▄	ý	³
D	-	=	M	]	m	}	ì	Ø	¡	¢	═	¦	Ý	²
E	.	>	N	^	n	~	Ä	×	«	¥	╬	Ì	¯	■
F	/	?	O	_	o	⌂	Å	ƒ	»	┐	¤	▀	´	NBSP

1252 Windows Latin 1 (ANSI)

	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
0		0	@	P	`	p	NOTUSED	NOTUSED	NBSP	°	À	Đ	à	ð
1	!	1	A	Q	a	q	NOTUSED	‘	¡	±	Á	Ñ	á	ñ
2	"	2	B	R	b	r	,	’	¢	²	Â	Ò	â	ò
3	#	3	C	S	c	s	ƒ	“	£	³	Ã	Ó	ã	ó
4	$	4	D	T	d	t	„	”	¤	´	Ä	Ô	ä	ô
5	%	5	E	U	e	u	…	•	¥	µ	Å	Õ	å	õ
6	&	6	F	V	f	v	†	-	¦	¶	Æ	Ö	æ	ö
7	'	7	G	W	g	w	‡	–	§	·	Ç	×	ç	÷
8	(	8	H	X	h	x	ˆ	~	¨	¸	È	Ø	è	ø
9	)	9	I	Y	i	y	‰	™	©	¹	É	Ù	é	ù
A	*	:	J	Z	j	z	Š	š	ª	º	Ê	Ú	ê	ú
B	+	;	K	[	k	{	‹	›	«	»	Ë	Û	ë	û
C	,	<	L	\	l	|	Œ	œ	¬	¼	Ì	Ü	ì	ü
D	-	=	M	]	m	}	NOTUSED	NOTUSED		½	Í	Ý	í	ý
E	.	>	N	^	n	~	NOTUSED	NOTUSED	®	¾	Î	Þ	î	þ
F	/	?	O	_	o		NOTUSED	Ÿ	¯	¿	Ï	ß	ï	ÿ

Pour avoir une correspondance directe entre l'affichage Notepad et les commandes passées sous CMD, il faudrait que la « Langue pour les programmes non Unicode » soit réglée sur CP-1252, or ce n'est pas le cas par défaut sous Win7. De plus, la liste des tables par défaut que l'on peut choisir sous Win7 se limite aux tables OEM de Windows (Anglais Etats-Unis = CP-437, Français Canadien = CP-863 etc…) donc la table CP-1252 est absente de la liste.

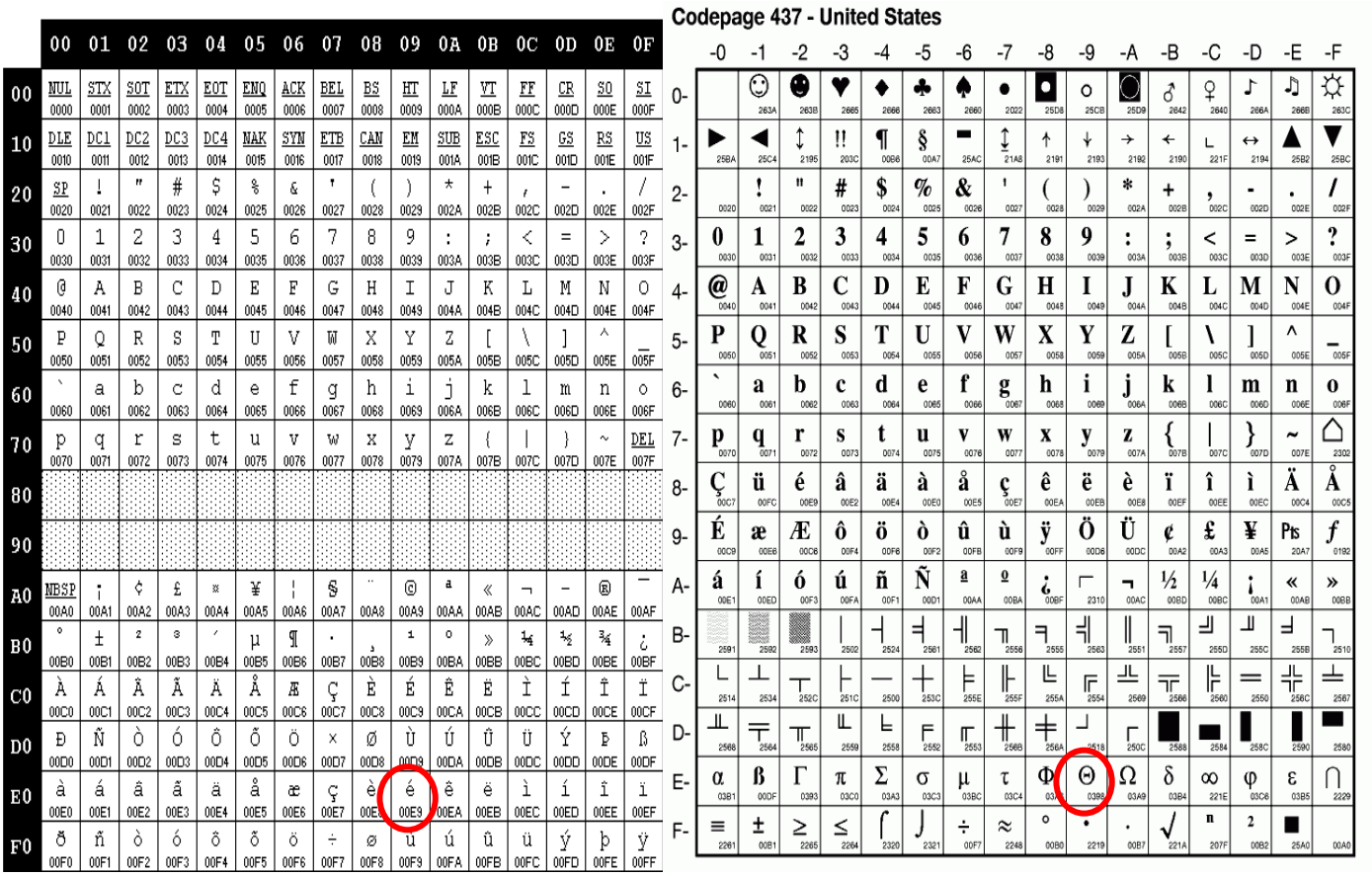
Solution : changer la page de codes en passant une commande CHCP (Change Code Page) au début du fichier batch. Par exemple :

chcp 28591

CP-28591 est la table « Latin 1; Western European (ISO) » définie par l'ISO 8859-1. Elle reprend la table CP-1252 et les premiers codes Unicodes. C'est donc une page créée pour permettre une transition vers l'Unicode. Tous les caractères français sont gérés, sauf Œ, œ, et le très rare Ÿ.

Après avoir passé la commande « chcp 28591 » dans le fichier batch, toutes les commandes suivantes sont exécutées et affichées selon CP-28591 donc les accents sont pris en charges et les commandes exécutées sans problème. Mais l'affichage dans la console peut poser problème selon la police utilisée.

La police d'affichage par défaut de la console est Raster = police ancienne Bitmap (= par points = non vectorielle) basée sur la page de codes CP-437 (Etats-Unis). Avec cette police, le caractère 'é' (E9 en hexadécimal dans la table CP-28591) est affiché comme 'Θ' (E9 en hexadécimal dans la table CP-437).



Solution : changer la police d'affichage de la console et choisir une police TrueType (= vectorielle) qui a une correspondance correcte entre l'affichage et la table CP-28591. Pour cela : Clic-droit sur la console > Propriétés > Police > puis choisir Lucida console par ex (icône TT = True Type).

Remarque :
La table CP-65001 est la table Unicode (UTF-8). La commande **chcp 65001** permet donc de passer des commandes en Unicode, mais l'affichage des caractères pose problème, y compris avec les polices True Type disponibles à ce jour pour la console.

Bilan
- enregistrer les fichiers texte Batch en **mode ANSI** (dans Notepad),
- écrire la commande « **chcp 28591** » au début du batch
- régler la **police d'affichage de la console sur Lucida console**, ou autre police True Type
- si certains caractères ne passent pas lors de l'exécution d'une commande, utiliser **ponctuellement « chcp 65001 »** pour faire fonctionner la commande, mais son affichage dans la console sera mauvais.

IDN, attaques homographiques et Punycode

Unicode est créé en 1991 mais il faut attendre **2001** pour qu'il soit utilisé dans les adresses web (URLs). Avant 2001, les adresses web ne pouvaient contenir que des caractères ASCII. La mise en place de l'IDNA (Internationalizing Domain Names in Applications) créé les **adresses IDN (Internationalized Domain Names)** = **noms de domaines internationalisés**. En France, les noms de domaines IDN en « .fr » ont été ouverts officiellement en **2012**.

Cette ouverture ouvre la porte aux **attaques homographiques**. C'est une attaque basée sur l'affichage par le navigateur d'un nom de domaine qui n'appartient pas à son propriétaire supposé. En effet, certains caractères ASCII sont difficiles à discerner d'autres caractères Unicode. Par exemple pour le nom de domaine « apple.com » . Ça ne se voit pas au premier coup d'oeil mais le « a » d'Apple (code U+0061) est en fait le caractère cyrillique « а » (code U+0430).

Pour se prémunir de genre d'attaque, le mieux est de forcer l'affichage des adresses web en ASCII. C'est là qu'intervient le **Punycode**, littéralement « code chétif ».

Punycode transforme une chaîne Unicode en une chaîne ASCII de manière unique et réversible. Par exemple, l'adresse précédente « apple.com » devient « xn--pple-43d.com » en punycode. Les 4 caractères du préfixe « xn-- » désignent une adresse punycode. Le choix de ce préfixe XN a été déterminé en 2003 parmi une liste de candidats (AA, QM à QZ, XA à XZ, et ZZ) grâce à un algorithme public et reproductible décrit ici :
http://www.ietf.org/mail-archive/web-old/ietf-announce-old/current/msg22565.html

Grâce au punycode, des noms de domaine contenant des emoji est possible. Par exemple l'adresse ♨.com est traduite en punycode par xn--j6h.com

Bilan : désactiver l'affichage des noms de domaine internationalisés pour forcer l'affichage en punycode :

- **Firefox** : aller sur la page « about:config », chercher la variable « network.IDN_show_punycode », changer « false » en « true ».
- **Chrome** : pas de possibilité de forcer l'affichage en punycode, et l'accès à la config de Chrome n'est plus possible depuis la version 57.

wikipedia
https://msdn.microsoft.com/en-us/library/windows/desktop/dd317756%28v=vs.85%29.aspx
https://msdn.microsoft.com/en-us/goglobal/cc305167.aspx
https://korben.info/se-proteger-contre-attaques-de-phishing-utilisent-noms-de-domaine-dautres-alphabets.html