

Codage des fichiers texte



1 Rappel sur la numération :

1.1 Sur combien de bits est codé un octet ?


--

1.2 Compléter le tableau en réalisant les conversions dans les différentes bases.

Valeur décimale	Valeur binaire	Valeur hexadécimale
33		
	01101110	
	01000001	

2 Comment coder un texte avec des '0' et des '1' ?

2.1 Les informations (texte, images, sons,...) sont désormais fréquemment numérisées. A cet effet, différents codes existent pour leur représentation.

- **Lancez** le Bloc-notes ( « Démarrer » /Tous les programmes/Accessoires).
- **Ecrivez** sur une seule ligne (sans retour à la ligne « Enter ») vos **nom** et **prénom** tout en minuscules et sans accent.
- **Enregistrez** votre fichier texte.
- **Lancez** un éditeur hexadécimal (exemple : <https://hexed.it/>), puis ouvrez à partir de celui votre fichier texte créé précédemment.
- **Affichez** son contenu au format Hexadécimal.

Comme vous le constatez le fichier ne contient que des valeurs numériques comprises entre 0 et 7F en hexadécimal. **Une table d'équivalence permet d'associer chaque lettre de l'alphabet à un code (ou valeur) numérique.**

Notez ci-dessous les premières lettres et codes associés.

[illegible][illegible]

Le jeu de caractères codes **ASCII** (American Standard Code for Information Interchange, « Code américain normalisé pour l'échange d'information ») est la norme de codage de caractères en informatique la plus connue, apparue dans les **années 1960**. [ASCII](#)
En ASCII, chaque symbole est codé sur 1 seul octet.

La table de correspondance ASCII décimal et hexadécimale est en annexe 1.

Ex : le caractère « A » correspond à la valeur ASCII 0x41 en hexadécimale.

- **Vérifiez** rapidement que la table d'équivalence ci-dessus permet de décoder votre fichier texte contenant votre nom et prénom.
- **Créez** un nouveau fichier texte en inscrivant votre nom et prénom **sur deux lignes**. (ajout d'un retour à la ligne)
- **Editez** ce fichier avec <https://hexed.it/>
- Quelles valeurs particulières sont présentes **entre** votre nom et prénom ? A quoi correspondent-elles selon la table ASCII ?

Valeurs particulières :

Signification :

Note : Les systèmes d'exploitation (Linux, Android ou Mac OS), se distinguent en utilisant qu'un seul de ces caractères pour signifier le saut de ligne et retour en début de ligne.

La table ASCII permet-elle de coder les caractères accentués ?

- Avec le Bloc-notes de Windows, nous avons créé un fichier texte avec des caractères accentués dont voici une visualisation sous <https://hexed.it/> (fichier sauvegardé au format par défaut appelé ANSI)

		EncodageAnsi.txt ✕	
00000000	53 70 E9 63 69 61 6C 69	74 E9 20 4E 53 49 20 0D	Specialite.NSI..
00000016	0A 41 6E 6E E9 65 20 32	30 31 39 2D 32 30 32 30	.Annee.2019-2020
00000032	0D 0A +		..

- A partir de la capture [Hexed](#) ci-dessus, **indiquez** par quelle valeur est codé le caractère accentué 'é' ? :

Rappelons que chaque symbole ASCII est codé sur un **seul octet**, permettant **256 valeurs** et que la table ASCII n'utilise que les **128 premières valeurs** !

Ainsi une multitude de tables dérivées de l'ASCII exploitent les 128 valeurs restantes pour coder les caractères régionaux ou accentués. (Toujours **en restant sur un seul octet**)

2.2 Pourquoi plusieurs tables de codage ?

Car en utilisant un codage sur un seul octet, il n'est pas possible de coder l'ensemble des caractères existants au niveau mondial.

Ces tables varient selon les **régions** et les **systèmes d'exploitations**.

Codes sur 1 octet (8 bits)	ISO/CEI 8859	-1 (Latin-1)-2 (Latin-2)-3 (Latin-3)-4 (Latin-4)-5 (Cyrillique)-6 (Arabe)-7 (Grec)-8 (Hébreu)-9 (Latin-5)-10 (Latin-6)-11 (Thaï)-12 (Devanagari)-13 (Latin-7)-14 (Latin-8)- 15 (Latin-9) -16 (Latin-10)
	Pages de code Windows	874 / TIS-620932 / Shift JIS936 / GBK949 / EUC-KR950 / Big5-1250-1251 1252 – 1253 – 1254 – 1255 – 1256 – 1257 – 1258-54936 / GB18030
	Pages de code Mac OS	MacRoman
	Pages de code DOS	437 - 720 - 737 - 775 - 850 - 852 - 855 - 857 - 858 - 860 - 861 - 862 863 - 864 - 865 - 866 - 869 - Kamenický - Mazovia - MIK - Iran System
	Pages de code diverses	ANSEL - DEC-MCS - KOI8-R (russe) - KOI8-U (ukrainien) - StandardEncoding (PostScript) - Page de code vidéotex - Page de code ZX Spectrum - Page de code Amstrad CPC - Page de code Atari ST - Page de code Amiga
	Non basés sur ISO 646	VISCII – EBCDIC - EBCDIC 297 - EBCDIC 8859

https://fr.wikipedia.org/wiki/ISO/CEI_8859-1

Astuce : pallier une touche de clavier défectueuse, utiliser des caractères spéciaux !

Il est possible sous Windows de saisir directement un caractère par le biais de son code. Pour cela il suffit de maintenir la touche ALT appuyée et de saisir la valeur du code correspondante au caractère souhaité. Le caractère apparaît lorsque vous relâchez la touche ALT.

(sous linux maintenir ctrl et shift appuyés puis taper la touche 'u' + le code puis relâcher ctrl et shift)

- **Vérifiez** dans un traitement de texte pour quelques caractères basiques. (caractères A, Z...).
- Puis **complétez** le tableau ci-dessous pour les valeurs proposées.

Valeur numérique	Table 850 (table régionale) (alt + num)	Table Windows 1252 (alt 0+num)
128		
163		
168		
191		
250		

Ces différentes tables de codage qui varient suivant les régions et les systèmes d'exploitations nuisent à la compatibilité des données.

2.3 Interopérabilité

Les plates-formes Windows utilisent la table Windows-1252 qui n'est pas reconnue dans d'autres systèmes d'exploitation : DOS n'utilise que les pages de codes d'IBM, et Linux/Unix utilisent nativement les pages de codes ISO (exemple ISO-8859-15) ou UTF-8 (Unicode).

Pour ces raisons notamment, Unicode (et plus généralement ISO/CEI 10646) tend à lui être préféré.

3 Un codage unique... L'UNICODE ?

3.1 Description

Le standard **Unicode**, dont la première publication remonte à 1991, a été développé dans le but de remplacer l'ensemble des codages existants (ASCII, CP1252, ISO/CEI 8859 etc...). Il est constitué d'un répertoire de 137 929 caractères, couvrant une centaine d'écritures (version Unicode 12.0 de mars 2019).

<https://unicode.org/consortium/consort.html>

Chaque caractère (ou symbole) est identifié par un code (valeur numérique) unique appelé **Point de Code**.

Extraits de la table Unicode présentée en valeurs décimales et hexadécimales

Contrôles C0 et latin de base

HEX		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	DEC	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
10	16	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
20	32	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
30	48	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	64	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	80	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	96	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	112	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

Exemple, caractère 'K'

En décimal : $64 + 11 = (75)_{10}$

En hexadécimal : $40 + B = (4B)_{16}$

notation unicode U+004B

Contrôles C1 et supplément latin-1

HEX		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	DEC	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
80	128	PAD	HOP	BPH	NBH	IND	NEL	SSA	ESA	HTS	HTJ	VTS	PLD	PLU	RI	SS2	SS3
90	144	DCS	PU1	PU2	STS	CCH	MW	SPA	EPA	SOS	SGCI	SCI	CSI	ST	OSC	PM	APC
A0	160	NBSP	¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	®	¯	
B0	176	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C0	192	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	208	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E0	224	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	240	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Voir Annexe 2, les tables de caractères Unicode en Latin étendu A, B et Grec

Pour les caractères usuels, les Points de Code de l'Unicode conservent une certaine similitude avec les Codes ASCII.

Afin de couvrir toutes les combinaisons de l'Unicode, un codage **sur strictement 1 seul octet n'est plus possible !**

Sous le traitement de texte Word, la séquence **alt + c** permet de basculer entre l'affichage du **point de code Unicode** (en notation hexadécimale) et l'affichage du caractère. Exemple taper E9, puis alt + c.
(sous libre office, la séquence alt+x permet le même résultat)

Valeur numérique du Point de Code en hexadécimal	Symbole correspondant (point de code + alt + c)
25F1	
3D0	

3.2 L'Unicode dans la pratique

Rappels :

Pour les caractères usuels, les Points de Code de l'Unicode conservent une certaine similitude avec les Codes ASCII.

En ASCII, chaque symbole est codé sur 1 seul octet.

Afin de couvrir toutes les combinaisons de l'Unicode, un codage **sur strictement 1 seul octet n'est plus possible** et peut nécessiter **jusqu'à 4 octets**.

Que sont UTF-8, UTF-16 ?

"UTF" est formé des initiales de "Unicode Transformation Format".

Si on veut stocker sur un support informatique un texte constitué de caractères Unicode, il faut choisir un procédé transformant une suite de caractères Unicode en une suite d'octets et réciproquement avoir un procédé pour retrouver la suite de caractères à partir d'une suite d'octets bien formée, ces données constituent ce que l'on appelle un encodage.

(source <https://www.tuteurs.ens.fr/faq/utf8.html>)

l'Unicode prévoit principalement 3 représentations :

- UTF-32 : 4 octets, soit 32 bits par caractères ou symboles. Codage exhaustif.
- UTF-16 : 2 à 4 octets par caractères ou symboles
- UTF-8 : 1 à 4 octets par caractères ou symboles.

Le logiciel Bloc-notes de Windows, permet de sauvegarder un texte sous différents encodages.

Analysons l'encodage d'un fichier texte en UTF-8

Nous avons réalisé une sauvegarde en encodage appelé 'UTF-8' sous le fichier **EncodageUtf8.txt**

- **Ouvrez** avec le Bloc-notes ce fichier. Remarquez qu'il contient des caractères accentués.

Visualisation du contenu du fichier avec l'éditeur [Hexed](#).

	EncodageUtf8.txt ×	
00000000	EF BB BF 53 70 C3 A9 63 69 61 6C 69 74 C3 A9 20	Spécialité.
00000016	4E 53 49 20 0D 0A 41 6E 6E C3 A9 65 20 32 30 31	NSI...Année.201
00000032	39 2D 32 30 32 30 0D 0A +	9-2020..

- Indiquez par quelles valeurs sont codés les symboles suivants ?

Lettres	A	n	n	é	e
Valeurs					

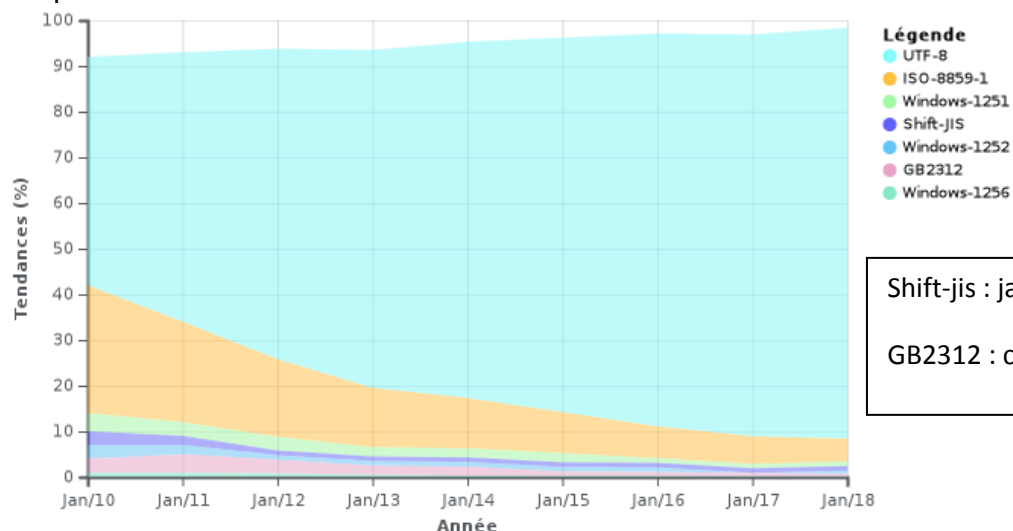
Selon la norme UTF-8, le caractère accentué 'é' visible sous le Bloc-notes a été codé à l'aide de deux octets. Le logiciel [Hexed](#) les interprète individuellement et les transpose en caractères spéciaux.

3.3 L'UTF-8

L'UTF-8 est un codage de l'Unicode dont les valeurs sont similaires à celles de la table ASCII (0 à 127). Son intérêt est un code à longueur variable de 1 à 4 octets permettant d'optimiser la taille des fichiers.

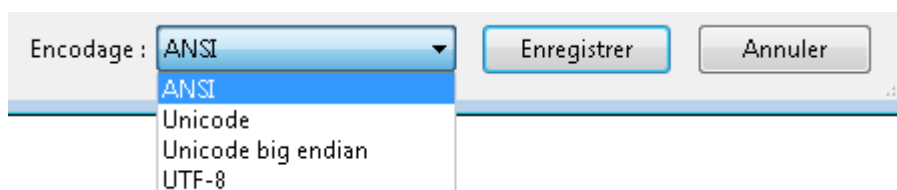
La norme UTF-8 décrit une méthode pour passer du Point de code à sa représentation par une suite d'octets. (Voir l'exemple en annexe 3)

L'UTF-8 tend à s'imposer sur le Web.



<https://fr.wikipedia.org/wiki/UTF-8>

Comment le Bloc-notes identifie l'encodage utilisé



Vous aurez remarqué que le fichier généré par le logiciel Bloc-notes de Windows en encodage UTF-8 contient à son début 3 valeurs supplémentaires* lui permettant d'identifier le type de codage utilisé.

*En [Unicode](#), l'**indicateur d'ordre des octets** ou BOM (pour l'anglais *byte order mark*) est une donnée qui indique l'utilisation d'un encodage Unicode ainsi que l'ordre des octets, généralement situé au début de certains fichiers texte.

(Source : [Indicateur d'ordre des octets](#))

Séquence éventuellement présente en début de fichier

Codage	Séquence d'octets En notation hexadécimale	Séquence d'octets En notation décimale
UFT-8	EF BB BF	239 187 191
UTF-16 Little Endian	FF FE	255 254

(Voir le [Boutisme](#) : **endianness** en anglais)

Annexe 1 - Tableau des caractères ASCII

caractère	code ASCII	code Hexadécimal		caractère	code ASCII	code Hexadécimal
NUL (Null)	0	00		G	71	47
SOH (Start of heading)	1	01		H	72	48
STX (Start of text)	2	02		I	73	49
ETX (End of text)	3	03		J	74	4A
EOT (End of transmission)	4	04		K	75	4B
ENQ (Enquiry)	5	05		L	76	4C
ACK (Acknowledge)	6	06		M	77	4D
BEL (Bell)	7	07		N	78	4E
BS (Backspace)	8	08		O	79	4F
TAB (tabulation horizontale)	9	09		P	80	50
LF (Line Feed, saut de ligne)	10	0A		Q	81	51
VT (Vertical tabulation, tabulation verticale)	11	0B		R	82	52
FF (Form feed)	12	0C		S	83	53
CR (Carriage return, retour à la ligne)	13	0D		T	84	54
SO (Shift out)	14	0E		U	85	55
SI (Shift in)	15	0F		V	86	56
DLE (Data link escape)	16	10		W	87	57
DC1 (Device control 1)	17	11		X	88	58
DC2 (Device control 2)	18	12		Y	89	59
DC3 (Device control 3)	19	13		Z	90	5A
DC4 (Device control 4)	20	14		[91	5B
NAK (Negative acknowledgement)	21	15		\	92	5C
SYN (Synchronous idle)	22	16]	93	5D
ETB (End of transmission block, fin de bloc de transmission)	23	17		^	94	5E
CAN (Cancel, annulation)	24	18		_	95	5F
EM (End of medium, fin du médium)	25	19		`	96	60
SUB (Substitute, substitut)	26	1A		a	97	61
ESC (Escape, caractère d'échappement)	27	1B		b	98	62
FS (File separator, séparateur de fichier)	28	1C		c	99	63
GS (Group separator, séparateur de groupe)	29	1D		d	100	64
RS (Record separator, séparateur d'enregistrement)	30	1E		e	101	65
US (Unit separator, séparateur d'enregistrement)	31	1F		f	102	66
SP (Space, espace)	32	20		g	103	67
!	33	21		h	104	68
*	34	22		i	105	69
#	35	23		j	106	6A
\$	36	24		k	107	6B
%	37	25		l	108	6C
&	38	26		m	109	6D
'	39	27		n	110	6E
(40	28		o	111	6F
)	41	29		p	112	70
*	42	2A		q	113	71
+	43	2B		r	114	72
,	44	2C		s	115	73
-	45	2D		t	116	74
.	46	2E		u	117	75
/	47	2F		v	118	76
0	48	30		w	119	77
1	49	31		x	120	78
2	50	32		y	121	79
3	51	33		z	122	7A
4	52	34		{	123	7B
5	53	35			124	7C
6	54	36		}	125	7D
7	55	37		~	126	7E
8	56	38		Touche suppression	127	7F
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				

Annexe 2 - Tableau des caractères UnicodeLatin étendu A

HEX		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	DEC	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
100	256	Ā	ā	Ă	ă	Ą	ą	Ć	ć	Ĉ	ĉ	Ċ	ċ	Č	č	Ď	ď
110	272	Đ	đ	Ē	ē	Ĕ	ĕ	Ė	ė	Ę	ę	Ė	ė	Ġ	ġ	Ģ	ģ
120	288	Ĝ	ĝ	Ģ	ģ	Ĥ	ĥ	Ħ	ħ	Ĩ	ĩ	Ī	ī	Ĵ	ĵ	Ĳ	ĳ
130	304	İ	ı	Ĳ	ĳ	Ĵ	ĵ	ķ	ķ	ķ	ķ	Ļ	ļ	Ł	ł	Ł	ł
140	320	Ł	ł	ł	ł	ł	ł	ł	ł	ł	ł	ł	ł	ł	ł	ł	ł
150	336	Œ	œ	Œ	œ	Œ	œ	Œ	œ	Œ	œ	Œ	œ	Œ	œ	Œ	œ
160	352	Š	š	Š	š	Š	š	Š	š	Š	š	Š	š	Š	š	Š	š
170	368	Ÿ	ÿ	Ÿ	ÿ	Ÿ	ÿ	Ÿ	ÿ	Ÿ	ÿ	Ÿ	ÿ	Ÿ	ÿ	Ÿ	ÿ

Latin étendu B

HEX		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	DEC	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
180	384	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ
190	400	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ
1A0	416	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ
1B0	432	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ
1C0	448	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ
1D0	464	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ
1E0	480	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ
1F0	496	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ
200	512	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ
210	528	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ
220	544	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ
230	560	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ
240	576	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ	Ḃ	ḃ

Grec et copte

HEX		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	DEC	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
U+0370	880	Ͱ	ͱ	Ͳ	ͳ	ʹ	͵	Ͷ	ͷ	͸	͹	ͺ	ͻ	ͼ	ͽ	Ϳ	Ϳ
U+0380	896	Ͱ	ͱ	Ͳ	ͳ	ʹ	͵	Ͷ	ͷ	͸	͹	ͺ	ͻ	ͼ	ͽ	Ϳ	Ϳ
U+0390	912	Ͱ	ͱ	Ͳ	ͳ	ʹ	͵	Ͷ	ͷ	͸	͹	ͺ	ͻ	ͼ	ͽ	Ϳ	Ϳ
U+03A0	928	Ͱ	ͱ	Ͳ	ͳ	ʹ	͵	Ͷ	ͷ	͸	͹	ͺ	ͻ	ͼ	ͽ	Ϳ	Ϳ
U+03B0	944	Ͱ	ͱ	Ͳ	ͳ	ʹ	͵	Ͷ	ͷ	͸	͹	ͺ	ͻ	ͼ	ͽ	Ϳ	Ϳ
U+03C0	960	Ͱ	ͱ	Ͳ	ͳ	ʹ	͵	Ͷ	ͷ	͸	͹	ͺ	ͻ	ͼ	ͽ	Ϳ	Ϳ
U+03D0	976	Ͱ	ͱ	Ͳ	ͳ	ʹ	͵	Ͷ	ͷ	͸	͹	ͺ	ͻ	ͼ	ͽ	Ϳ	Ϳ
U+03E0	992	Ͱ	ͱ	Ͳ	ͳ	ʹ	͵	Ͷ	ͷ	͸	͹	ͺ	ͻ	ͼ	ͽ	Ϳ	Ϳ
U+03F0	1008	Ͱ	ͱ	Ͳ	ͳ	ʹ	͵	Ͷ	ͷ	͸	͹	ͺ	ͻ	ͼ	ͽ	Ϳ	Ϳ

Annexe 3 - Comment est converti un point de Code en code UTF-8 ?

Extrait de la spécification de la norme : [Codage Utf8 Wikipédia](#)

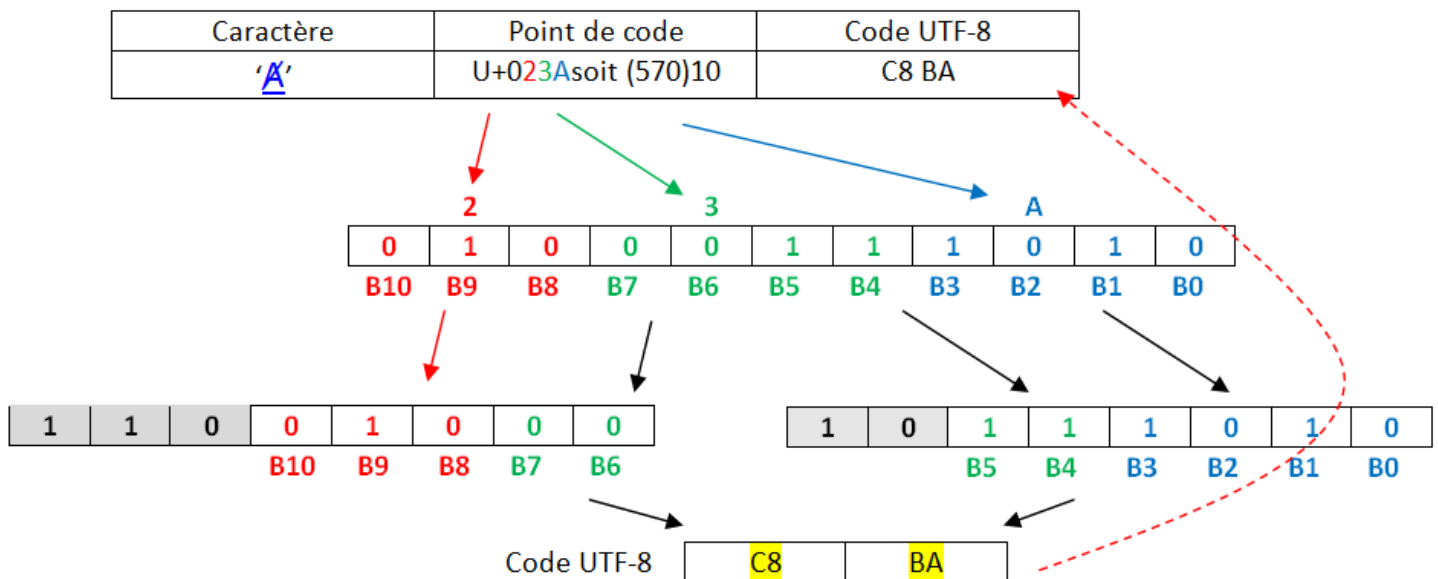
Définition du nombre d'octets utilisés dans le codage (uniquement les séquences valides)

Caractères codés	Représentation binaire UTF-8	Premier octet valide (hexadécimal)	Signification
U+0000 à U+007F	0xxxxxxx	00 à 7F	1 octet, codant 7 bits
U+0080 à U+07FF	10xxxxxx 10xxxxxx	C2 à DF	2 octets, codant 11 bits

Un exemple, à partir de l'extrait de la table Unicode(version hexa), rappelez le Point de Code du caractère 'À' (valeur numérique) : **U+023A** $(23A)_{16} = (570)_{10}$

Le caractère appartient à l'intervalle U+0080 à U+07FF, donc les **11 bits** de son code seront répartis sur **deux octets** en code UTF-8

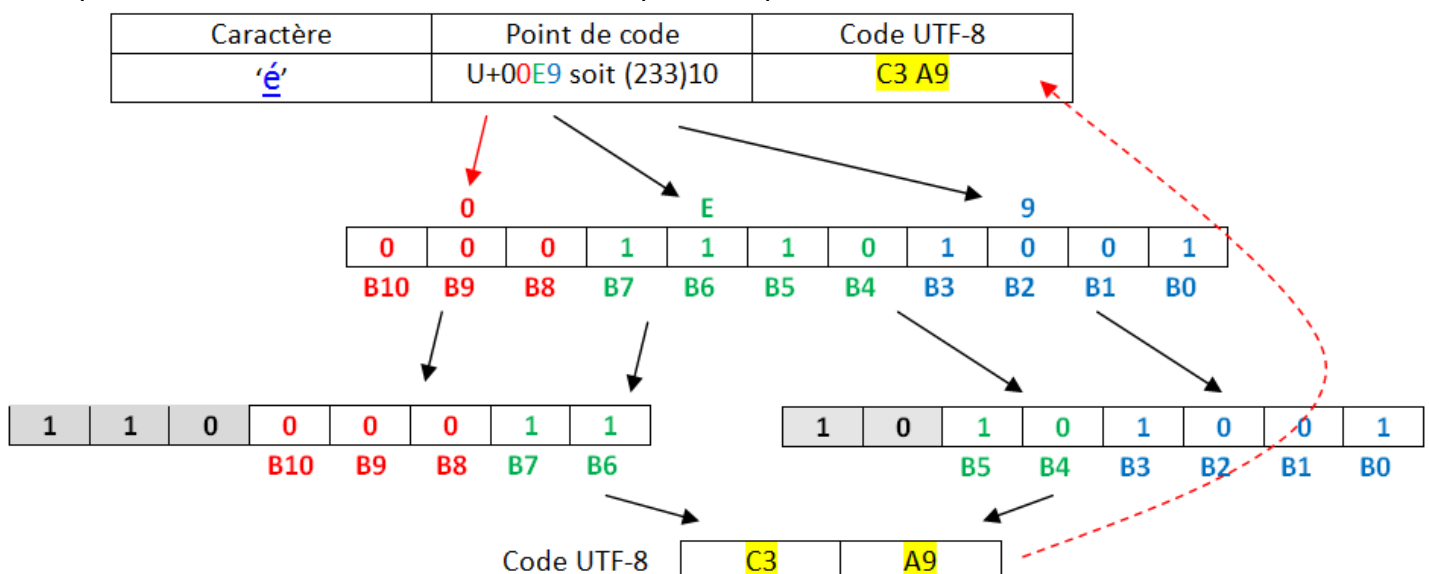
- Convertir le code en binaire sur 11 bits
- Répartir les 11bits sur deux octets selon la disposition spécifiée dans la norme



Un outil pour vérifier : <http://hapax.qc.ca/conversion.fr.html>

Deuxième exemple : Lettre 'é' :

- Répartir les 11bits sur deux octets selon la disposition spécifiée dans la norme



TP codage ASCII d'après E. BANSIÈRE – P. JONIN