L'encodage de texte

1. Qu'est-ce qu'un encodage ?

Le jeu de caractères

Le codage des caractères est une convention qui permet, à travers un codage connu de tous, de transmettre de l'information textuelle. On définit alors un jeu de caractères (charset en anglais) appelé aussi répertoire.
A chaque caractère est assigné un numéro unique appelé code. On obtient alors un jeu de caractères codés et on peut alors créer un tableau de correspondance. Par exemple[1] :

A	B	...	Z	0	1	...	9	.	,	‘	?	!	-	espace
0	1		25	26	27		35	36	37	38	39	40	41	42

Le codage du message “**LA NSI, C'EST BIEN !**” peut s'effectuer par l'utilisation de la table de correspondance :

L	A	esp	N	S	I	esp	C	‘	E	S	T	esp	B	I	E	N	esp	!
11	0	42	13	18	8	42	2	38	4	18	19	42	1	8	4	13	42	40

Cependant attention, l'écriture du message ne peut pas être effectuée comme cela :

1104213188422384181942184134240

Il y a en effet ambiguité sur la longueur des caractères. La première lettre est-elle un B (1) ou un L (11) ? Et si la première lettre est un B (1) la deuxième est elle encore un B (1) ou un K (10)? etc...

Il est donc nécessaire de mettre en place un encodage. Ici un encodage simple est de coder l'information sur deux digits pour éviter toute confusion. Ainsi le message devient :

11004213180842023804181942010804134240

De cette même façon, vous êtes donc en mesure de décoder le message précédent :

01141309142017421104184204110421041836 -> BONJOUR LES ELEVES.

Ce simple exemple montre qu'une stratégie d'encodage est nécessaire.

Repères historiques

On n'a pas pas attendu l'informatique pour transmettre des messages autrement que par écrit. Le code Morse en est un parfait exemple (figure 1). Le braille est également un autre exemple (figure 2).

1 https://zestedesavoir.com/tutoriels/1114/comprendre-les-encodages/1-theorie/

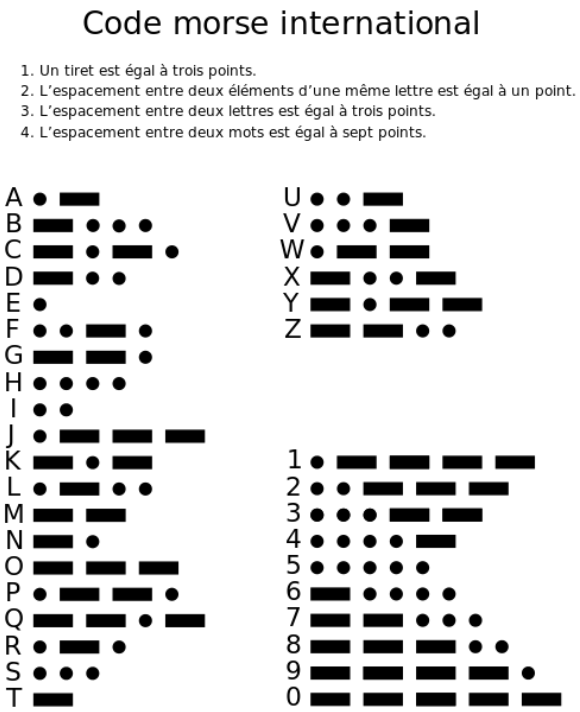


Figure 1: Table d'encodage du morse

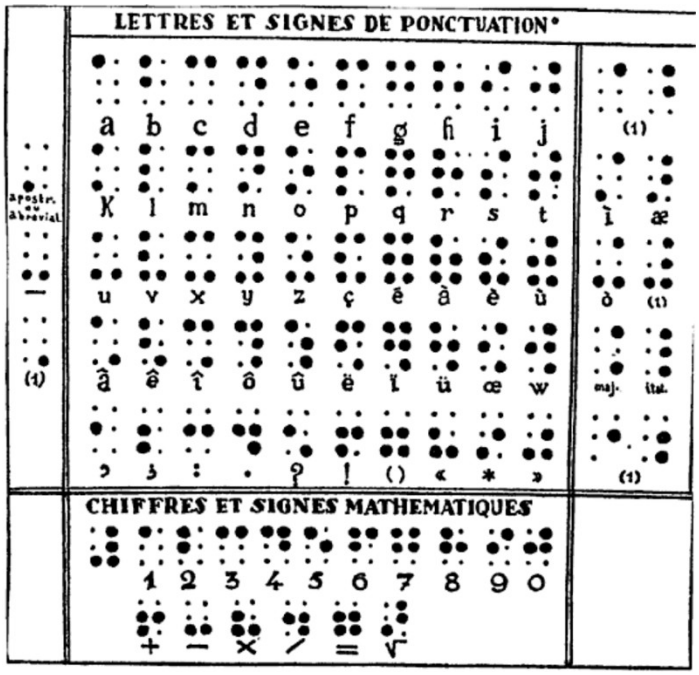


Figure 2: Table d'encodage du braille

Question 1) **Calculer** le nombre de caractères codable par les 6 points de touche d'une cellule braille

Il y a 6 caractères prenant 2 états possible (base 2). Il y a donc 2^6 caractères codables sur une cellule braille

2.Les encodages utilisés en informatique

Le code ASCII

L'un des premiers encodages historiques est l'ASCII (American Standard Code for Information Interchange, en français, le code américain normalisé pour l'échange d'informations). Cet encodage s'est imposé au début des années 1960. Le code ASCII (figure 3) contient 128 caractères qui suffisent à écrire des textes en anglais. Cependant ce code ne convient pas pour les autres langues.

Dec	Hex	Name	Char	Ctrl-char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	Null	NUL	CTRL-@	32	20	Space	64	40	@	96	60	`
1	1	Start of heading	SOH	CTRL-A	33	21	!	65	41	A	97	61	a
2	2	Start of text	STX	CTRL-B	34	22	"	66	42	B	98	62	b
3	3	End of text	ETX	CTRL-C	35	23	#	67	43	C	99	63	c
4	4	End of xmit	EOT	CTRL-D	36	24	$	68	44	D	100	64	d
5	5	Enquiry	ENQ	CTRL-E	37	25	%	69	45	E	101	65	e
6	6	Acknowledge	ACK	CTRL-F	38	26	&	70	46	F	102	66	f
7	7	Bell	BEL	CTRL-G	39	27	'	71	47	G	103	67	g
8	8	Backspace	BS	CTRL-H	40	28	(	72	48	H	104	68	h
9	9	Horizontal tab	HT	CTRL-I	41	29	)	73	49	I	105	69	i
10	0A	Line feed	LF	CTRL-J	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	VT	CTRL-K	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	FF	CTRL-L	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage feed	CR	CTRL-M	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	SO	CTRL-N	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	SI	CTRL-O	47	2F	/	79	4F	O	111	6F	o
16	10	Data line escape	DLE	CTRL-P	48	30	0	80	50	P	112	70	p
17	11	Device control 1	DC1	CTRL-Q	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	DC2	CTRL-R	50	32	2	82	52	R	114	72	r
19	13	Device control 3	DC3	CTRL-S	51	33	3	83	53	S	115	73	s
20	14	Device control 4	DC4	CTRL-T	52	34	4	84	54	T	116	74	t
21	15	Neg acknowledge	NAK	CTRL-U	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	SYN	CTRL-V	54	36	6	86	56	V	118	76	v
23	17	End of xmit block	ETB	CTRL-W	55	37	7	87	57	W	119	77	w
24	18	Cancel	CAN	CTRL-X	56	38	8	88	58	X	120	78	x
25	19	End of medium	EM	CTRL-Y	57	39	9	89	59	Y	121	79	y
26	1A	Substitute	SUB	CTRL-Z	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	ESC	CTRL-[	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	FS	CTRL-\	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	GS	CTRL-]	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	RS	CTRL-^	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	US	CTRL-_	63	3F	?	95	5F	_	127	7F	DEL

Figure 3: Table ASCII

Généralement, l'encodage est présenté en hexadécimal. Par exemple le code 0x41, correspond à la lettre « A ».

Question 2) **Déterminer** la signification du code suivant codé en ASCII

4C 69 66 65 20 69 6D 69 74 61 74 65 73 20 61 72 74

L i f e i m i t a t e s a r t

Question 3) **Déterminer** la taille (en octets) de la phrase «Roger is an alien !».

La phrase comporte 19 caractères. Chaque caractère est codé sur 1 octet. La phrase a donc une taille de 19 octets

Question 4) **Indiquer** si l'encodage ASCII permet de coder la phrase «Le cinquième élément est un bon film»

LA phrase en peut pas être encodée en ASCII puisqu'elle comporte des caractères non reconnus (é)

Le codage ASCII n'est maintenant que très rarement utilisé et ce pour des programmes spécifiques et relativement anciens.

Le codage ISO-8859-1

Le codage ASCII précédemment présenté utilise 7 bits pour l'encodage des caractères. Avec l'avancée techniques et des méthodes de contrôles, le 8ème bit a pu être utilisé pour coder plus de caractères.

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	{	l	}	~	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

Figure 4: Codage ISO-8859-1 (latin 1)

Utiliser le 8ème bit permet de passer à 256 caractères possibles, cela permet notamment d'obtenir beaucoup des caractères de la langue française. L'encodage principalement utilisé est l'iso-8859-1 ou latin-1 (il manque cependant par exemple le caractère œ et €). La table d'encodage est représentée en figure 4, on constate qu'elle englobe la table ASCII.

Les tables iso-8859 sont aux nombres de 16. Une table également exploitable en français est l'iso-8859-15 (ou latin-9). Cette table contient notamment les deux caractères

Différences ISO 8859-15 — ISO 8859-1								
Position	0xA4	0xA6	0xA8	0xB4	0xB8	0xBC	0xBD	0xBE
8859-1	¤	¦	¨	´	¸	¼	½	¾
8859-15	€	Š	š	Ž	ž	Œ	œ	Ÿ

Figure 5: Caractères différents sur la table latin-9

manquants décrits précédemment. Les caractères différents entre les deux tables sont indiqués sur la figure 5

L'encodage Windows-1252

Windows s'est basé sur latin-1 pour mettre au point son jeu de caractères et son encodage appelé Windows-1252 (qu'on appelle aussi CP1252 ou ANSI). Cet encodage a l'avantage de permettre l'utilisation des caractères œ et €.
Cependant la multiplication des encodages allant souvent de paire avec une modification des jeux de caractères pose un vrai problème dans la transmission de données d'un continent à l'autre. C'est à ce moment là qu'intervient l'Unicode.

L'Unicode

L'Unicode est un jeu de caractères universel qui contient l'ensemble des alphabets et des caractères spéciaux. Le développement de cette norme s'est fait au début des années 1990 et en est actuellement à sa septième version. Elle recense environ 110 000 caractères. On y trouve donc les alphabets latin, cyrillique, grec, arabe etc.
Pour coder ce jeu de caractères, il faut alors **4 octets**.

Question 5) **Calculer** le nombre de caractères différents que peut posséder la table de caractère Unicode

La table de caractère Unicode code les caractères sur 4 octets soit 32 bits. Elle possède donc au maximum 2^{32} caractères soit 4 294 967 296 caractères différents

UTF-8

Jusqu'à présent, on pouvait confondre jeu de caractères et encodage, mais c'est à ce moment là qu'intervient l'UTF-8. En effet, coder une lettre sur 4 octets n'est vraiment pas utile et alourdit inutilement le fichier. Prenons en exemple la lettre « A », il s'agit du 66ème, son code en hexadécimal est 0x41. Son codage en Unicode est alors 00000000 00000000 00000000 01000001. Cela fait quand même beaucoup de 0 pour rien du tout.
L'UTF-8 s'appuie sur le jeu de caractères de l'Unicode mais l'encodage s'effectue à longueur variable.
L'intérêt de l'UTF-8 est d'utiliser un nombre d'octets minimal afin d'encoder les caractères. Au final, on peut déterminer le nombre d'octets nécessaires en UTF-8 pour faire l'encodage du caractère suivant ce tableau :

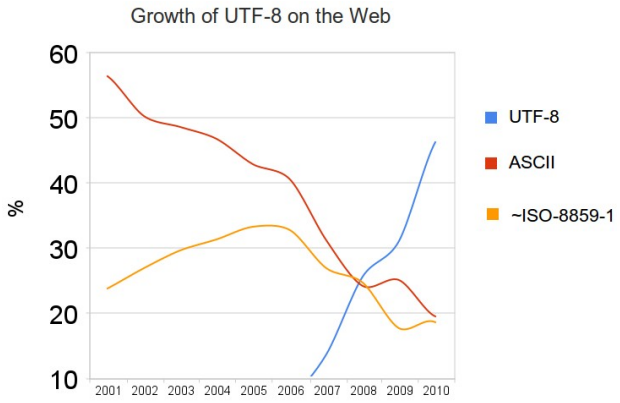
Définition du nombre d'octets utilisés

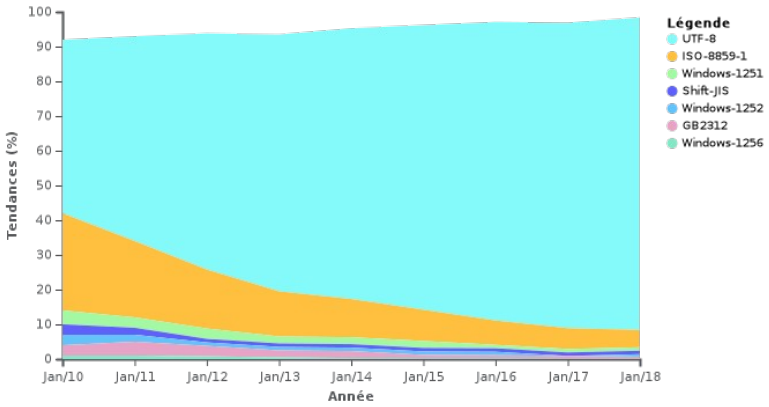
Représentation binaire UTF-8	Signification
0xxxxxxx	1 octet codant 1 à 7 bits
110xxxxx 10xxxxxx	2 octets codant 8 à 11 bits
1110xxxx 10xxxxxx 10xxxxxx	3 octets codant 12 à 16 bits
11110xxx 10xxxxxx 10xxxxxx 10xxxxxx	4 octets codant 17 à 21 bits

Comparaison de l'Unicode et de l'UTF-8

Le développement de l'encodage UTF-8 est incontestable au cours du temps. En effet, son universalité rend son utilisation très pratique et accessible à tous. En 2010, 50% environ des sites Web utilisaient un encodage UTF-8, de nos jours c'est plus de 90%, comme le montre ces deux graphiques[2].

2 Wikipédia : https://fr.wikipedia.org/wiki/UTF-8


Growth of UTF-8 on the Web
%
60
50
40
30
20
10
2001 2002 2003 2004 2005 2006 2007 2008 2009 2010
UTF-8
ASCII
~ISO-8859-1


Tendances (%)
100
90
80
70
60
50
40
30
20
10
0
Jan/10 Jan/11 Jan/12 Jan/13 Jan/14 Jan/15 Jan/16 Jan/17 Jan/18
Année
Légende
UTF-8
ISO-8859-1
Windows-1251
Shift-JIS
Windows-1252
GB2312
Windows-1256