Codage numérique : du nombre au pixel - Cours 2

Codage des textes

L1 – Université de Lorraine
B. Girau et N. de Rugy Altherre

Transparents disponibles sur la plateforme de cours en ligne

Définition d'un texte :

- Succession (suite) de mots qui porte un sens et utilise les structures d'une langue (vocabulaire, conjugaison, etc.).
- Point de vue informatique :
  - par opposition aux « données binaires » (et pourtant le texte est codé sur des chiffres binaires)
  - texte brut : ensemble de caractères (alphabet, ponctuation, retours à la ligne, etc.)
  - fichier texte versus fichier formaté (avec indications de présentation du texte)
  - éditeur de texte (emacs, etc.) vs traitement de texte (word, etc.)

Le codage du texte dépend du support :

- télégraphe $\rightarrow$ morse
- papier $\rightarrow$ alphabet
- telex $\rightarrow$ baudot
- ordinateur $\rightarrow$ ASCII, latin-1, UTF,...

Morse

- Technique simple de communication définie par Samuel Morse (1791-1872, U.S.).
- Code composé de points et de traits.
- Chaque caractère a son propre code.
- Transmission via signaux sonores, lumineux, électriques (point = court, trait = long).
- Pourrait très bien être utilisé en informatique (point = 0, trait =1 par exemple).
- Il faut bien séparer les codes de chaque caractère (”silence”) : pas pratique en informatique.

Morse

A •−	J •−−−	S •••	2 ••−−−
B −•••	K −•−	T −	3 •••−−
C −•−•	L •−••	U ••−	4 ••••−
D −••	M −−	V •••−	5 •••••
E •	N −•	W •−−	6 −••••
F ••−•	O −−−	X −••−	7 −−•••
G −−•	P •−−•	Y −•−−	8 −−−••
H ••••	Q −−•−	Z −−••	9 −−−−•
I ••	R •−•	1 •−−−−	0 −−−−−

Alphabet

- Ensemble de caractères.
- Propre à des groupes de langues.
- L'alphabet informatique : '0' et '1'.

Alphabet

Alphabet grec.

Αα alpha (άλφα)	Ββ bêta (βήτα)	Γγ gamma (γάμμα)	Δδ delta (δέλτα)
Εε epsilon (έψιλον)	Ζζ zêta (ζήτα)	Ηη êta (ήτα)	Θθ thêta (θήτα)
Ιι iota (ιώτα)	Κκ kappa (κάππα)	Λλ lambda (λάμβδα)	Μμ mu (μυ)
Νν nu (νυ)	Ξξ ksi (ξι)	Οο omicron (όμικρον)	Ππ pi (πι)
Ρρ rho (ρω)	Σσ sigma (σίγμα)	Ττ tau (ταυ)	Υυ upsilon (ύψιλον)
Φφ phi (φι)	Χχ khi (χι)	Ψψ psi (ψι)	Ωω oméga (ωμέγα)

Alphabet

Alphabet japonais (un des trois utilisés).

Katakana (カタカナ)

	wa ワ	ra ラ	ya ヤ	ma マ	pa パ	ba バ	ha ハ	na ナ	da ダ	ta タ	za ザ	sa サ	ga ガ	ka カ	a ア
		ri リ		mi ミ	pi ピ	bi ビ	hi ヒ	ni ニ	di ヂ	chi チ	ji ジ	shi シ	gi ギ	ki キ	i イ
		ru ル	yu ユ	mu ム	pu プ	bu ブ	fu フ	nu ヌ	du ヅ	tsu ツ	zu ズ	su ス	gu グ	ku ク	u ウ
		re レ		me メ	pe ペ	be ベ	he ヘ	ne ネ	de デ	te テ	ze ゼ	se セ	ge ゲ	ke ケ	e エ
n ン	(o) ヲ	ro ロ	yo ヨ	mo モ	po ポ	bo ボ	ho ホ	no ノ	do ド	to ト	zo ゾ	so ソ	go ゴ	ko コ	o オ

Alphabet

La différence entre les alphabets grec et japonais ?

...

Alphabet

La différence entre les alphabets grec et japonais ?

...

Un caractère japonais (Katakana) représente une *syllabe* alors qu'un caractère grec représente un *phonème* (plus ou moins).

Alphabet

En informatique, coder un texte consiste à coder chacun de ses caractères (un peu comme le Morse).

Mais il faut spécifier quel alphabet est utilisé/encodé (i.e. quel ensemble de caractères, aussi appeler jeu de caractères, *charset* en anglais).

Le codage consiste donc juste à exprimer un alphabet sous la forme de mots utilisant l'alphabet binaire.

Les ordinateurs ne sont pas les seuls à faire cela . . .

Code Baudot

Le télex est un réseau de communication entre téléscripteurs. Il commence en 1930, connait son apogée dans les années 90 et disparaît au début du XXIème siècle (Orange arrête les derniers abonnements en 2017). Le codage utilisé est le code Baudot.

Code Baudot

Code Baudot

Codage sur 5 bits (32 combinaisons). Ce qui ne permet pas d'encoder les lettres (26), les chiffres (10) et les caractères de ponctuation. Deux jeux sont donc créés : un pour les lettres (Lower Case), l'autre pour les chiffres et caractères (Upper Case). Deux caractères permettent de passer de l'un à l'autre.

Rappel : Un bit stocke soit 0 soit 1. Un octet est un multiplet de 8 bits. Il y a $2^8 = 256$ octets différents.

Lettre	Chiffre	Code	Lettre	Chiffre	Code
A	-	00011	Q	1	10111
B	?	11001	R	4	01010
C	:	01110	S	'	00101
D	Qui ?	01001	T	5	10000
E	3	00001	U	7	00111
F	(1)	01101	V	=	11110
G	(1)	11010	W	2	10011
H	(1)	10100	X	/	11101
I	8	00110	Y	6	10101
J	Ring	01011	Z	+	10001
K	(	01111	CR (retour chariot)		01000
L	)	10010	LF (saut ligne)		00010
M	.	11100	**Lettre**		11111
N	,	01100	**Chiffre**		11011
O	9	11000	SP (espace)		00100
P	0	10110	(inutilisé)		00000

1) Inutilisés en TELEX, on peut y trouver des codes nationaux (É, %, H en France).

Source : yves LESCOP

Il est créé dans les années 1960 pour assurer la transmission de textes vers des terminaux ou des imprimantes.

Code ASCII

Le code ASCII (*American Standard Code for Information Interchange*) est une norme pour le codage de caractère. Il travaille sur 7 bits et encode 95 caractères.

Il est créé dans les années 1960 pour assurer la transmission de textes vers des terminaux ou des imprimantes.

Code ASCII

Le code ASCII (*American Standard Code for Information Interchange*) est une norme pour le codage de caractère. Il travaille sur 7 bits et encode 95 caractères.

Problème : il encode bien l'anglais, mais pas les langues utilisants des caractères spécifiques (français, arabe, chinois, etc.).

Dec	Bin	Hex	Char	Dec	Bin	Hex	Char	Dec	Bin	Hex	Char	Dec	Bin	Hex	Char
0	0000 0000	00	[NUL]	32	0010 0000	20	space	64	0100 0000	40	@	96	0110 0000	60	`
1	0000 0001	01	[SOH]	33	0010 0001	21	!	65	0100 0001	41	A	97	0110 0001	61	a
2	0000 0010	02	[STX]	34	0010 0010	22	"	66	0100 0010	42	B	98	0110 0010	62	b
3	0000 0011	03	[ETX]	35	0010 0011	23	#	67	0100 0011	43	C	99	0110 0011	63	c
4	0000 0100	04	[EOT]	36	0010 0100	24	$	68	0100 0100	44	D	100	0110 0100	64	d
5	0000 0101	05	[ENQ]	37	0010 0101	25	%	69	0100 0101	45	E	101	0110 0101	65	e
6	0000 0110	06	[ACK]	38	0010 0110	26	&	70	0100 0110	46	F	102	0110 0110	66	f
7	0000 0111	07	[BEL]	39	0010 0111	27	'	71	0100 0111	47	G	103	0110 0111	67	g
8	0000 1000	08	[BS]	40	0010 1000	28	(	72	0100 1000	48	H	104	0110 1000	68	h
9	0000 1001	09	[TAB]	41	0010 1001	29	)	73	0100 1001	49	I	105	0110 1001	69	i
10	0000 1010	0A	[LF]	42	0010 1010	2A	*	74	0100 1010	4A	J	106	0110 1010	6A	j
11	0000 1011	0B	[VT]	43	0010 1011	2B	+	75	0100 1011	4B	K	107	0110 1011	6B	k
12	0000 1100	0C	[FF]	44	0010 1100	2C	,	76	0100 1100	4C	L	108	0110 1100	6C	l
13	0000 1101	0D	[CR]	45	0010 1101	2D	-	77	0100 1101	4D	M	109	0110 1101	6D	m
14	0000 1110	0E	[SO]	46	0010 1110	2E	.	78	0100 1110	4E	N	110	0110 1110	6E	n
15	0000 1111	0F	[SI]	47	0010 1111	2F	/	79	0100 1111	4F	O	111	0110 1111	6F	o
16	0001 0000	10	[DLE]	48	0011 0000	30	0	80	0101 0000	50	P	112	0111 0000	70	p
17	0001 0001	11	[DC1]	49	0011 0001	31	1	81	0101 0001	51	Q	113	0111 0001	71	q
18	0001 0010	12	[DC2]	50	0011 0010	32	2	82	0101 0010	52	R	114	0111 0010	72	r
19	0001 0011	13	[DC3]	51	0011 0011	33	3	83	0101 0011	53	S	115	0111 0011	73	s
20	0001 0100	14	[DC4]	52	0011 0100	34	4	84	0101 0100	54	T	116	0111 0100	74	t
21	0001 0101	15	[NAK]	53	0011 0101	35	5	85	0101 0101	55	U	117	0111 0101	75	u
22	0001 0110	16	[SYN]	54	0011 0110	36	6	86	0101 0110	56	V	118	0111 0110	76	v
23	0001 0111	17	[ETB]	55	0011 0111	37	7	87	0101 0111	57	W	119	0111 0111	77	w
24	0001 1000	18	[CAN]	56	0011 1000	38	8	88	0101 1000	58	X	120	0111 1000	78	x
25	0001 1001	19	[EM]	57	0011 1001	39	9	89	0101 1001	59	Y	121	0111 1001	79	y
26	0001 1010	1A	[SUB]	58	0011 1010	3A	:	90	0101 1010	5A	Z	122	0111 1010	7A	z
27	0001 1011	1B	[ESC]	59	0011 1011	3B	;	91	0101 1011	5B	[	123	0111 1011	7B	{
28	0001 1100	1C	[FS]	60	0011 1100	3C	<	92	0101 1100	5C	\	124	0111 1100	7C	|
29	0001 1101	1D	[GS]	61	0011 1101	3D	=	93	0101 1101	5D	]	125	0111 1101	7D	}
30	0001 1110	1E	[RS]	62	0011 1110	3E	>	94	0101 1110	5E	^	126	0111 1110	7E	~
31	0001 1111	1F	[US]	63	0011 1111	3F	?	95	0101 1111	5F	_	127	0111 1111	7F	[DEL]

Remarque : entre une majuscule et une minuscule, seul le bit 5 change.

ISO 8859-1

C'est une extension de la norme ASCII destinée à intégrer des symboles courants de l'alphabet latin ainsi que les caractères accentués de plusieurs langues. Il est codé sur 8 bits.

ISO/CEI 8859-1															
	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	*positions inutilisées*															
1x																
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	}	~	
8x	*positions inutilisées*															
9x																
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

Source : wikipedia

Unicode

- En fait plusieurs extensions d'ASCII ont été définies, non compatibles (y compris pour un même alphabet).
- Travail de normalisation normalement assuré par ISO (organisation internationale de normalisation).
- Retard d'ISO : des éditeurs de logiciels décident de se regrouper pour créer la norme Unicode (en fait Unicode 1.0), sur 16 bits.
- Volonté : avoir enfin un système d'encodage universel (et partagé).

D'Unicode à UTF

Unicode et ISO finissent par travailler ensemble.

Unicode

Le standard Unicode est produit par une organisation a but non lucratif (Consortium Unicode) ayant pour objectif d'encoder tout caractère utilisé par une langue humaine. Plus d'un million de caractères sont encodés de façon unique.

Par exemple le 65ème caractère du répertoire Unicode est la lettre A. Le 8365ème caractère est le symbole euro €. Le 723ème est la lettre syriaque semkath.

Des caractères sont rajoutés chaque année (par exemple de nouveaux emojis).

D'Unicode à UTF

Principe : on sépare les points de code (≪ vrai ≫ code Unicode des caractères, potentiellement en 32 bits, en réalité sur 21 bits) de leur encodage en machine.
Différentes méthodes d'encodage machine pour les caractères Unicode sont définies et normées. Ce sont les normes UTF, *Unicode Transformation Format*.

UTF-8 est une norme pour encoder les caractères. UFT-16 et UTF-32 sont d'autres normes.
Chaque norme précise comment obtenir algorithmiquement le code Unicode à partir du code machine UTF utilisé.

UTF-*n* indique que la taille de l'encodage est *au minimum* de *n* bits, mais ce peut être une taille variable.

UTF-8

UTF-8 a une taille variable : il encode un caractère sur 1 à 4 octets

Les 128 premiers caractères du répertoire unicode sont représentés par 1 octet ; ceux entre le 129ème et le 2047ème caractère sont représentés par 2 octets, etc.

Certains bits dans chaque octet ne sont pas utilisés pour représenter le caractère mais pour automatiser la lecture (par une machine) de l'encodage.

Une machine lisant un code UTF-8 commençant par un 0 sait que le premier caractère du code est encodé sur 1 seul octet et est l'un des 128 premiers caractères.

UTF-8

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	110xxxxx 10xxxxxx	C2 à DF	2 octets, codant 11 bits
U+0800 à U+0FFF	1110 0000 10 1xxxxx 10xxxxxx	E0 (le 2e octet est restreint de A0 à BF)	3 octets, codant 16 bits
U+1000 à U+1FFF	1110 0001 10xxxxxx 10xxxxxx	E1	
U+2000 à U+3FFF	1110 001x 10xxxxxx 10xxxxxx	E2 à E3	
U+4000 à U+7FFF	1110 01xx 10xxxxxx 10xxxxxx	E4 à E7	
U+8000 à U+BFFF	1110 10xx 10xxxxxx 10xxxxxx	E8 à EB	
U+C000 à U+CFFF	1110 1100 10xxxxxx 10xxxxxx	EC	
U+D000 à U+D7FF	1110 1101 10 0xxxxx 10xxxxxx	ED (le 2e octet est restreint de 80 à 9F)	
U+E000 à U+FFFF	1110 111x 10xxxxxx 10xxxxxx	EE à EF	
U+10000 à U+1FFFF	11110 000 10 01xxxx 10xxxxxx 10xxxxxx	F0 (le 2e octet est restreint de 90 à BF)	4 octets, codant 21 bits
U+20000 à U+3FFFF	11110 000 10 1xxxxx 10xxxxxx 10xxxxxx		
U+40000 à U+7FFFF	11110 001 10xxxxxx 10xxxxxx 10xxxxxx	F1	
U+80000 à U+FFFFF	11110 01x 10xxxxxx 10xxxxxx 10xxxxxx	F2 à F3	
U+100000 à U+10FFFF	11110 100 10 00xxxx 10xxxxxx 10xxxxxx	F4 (le 2e octet est restreint de 80 à 8F)	

Source : wikipedia

UTF-8

Exemples de codage UTF-8

Type	Caractère	Point de code (hexadécimal)	Valeur scalaire		Codage UTF-8	
			décimal	binaire	binaire	hexadécimal
Contrôles	[NUL]	U+0000	0	*0000000*	0*0000000*	00
	[US]	U+001F	31	*0*011111	0*0*011111	1F
Texte	[SP]	U+0020	32	*0*100000	0*0*100000	20
	A	U+0041	65	1000001	01000001	41
	~	U+007E	126	1111110	01111110	7E
Contrôles	[DEL]	U+007F	127	1111111	01111111	7F
	[PAD]	U+0080	128	*00*010 000000	11*000*010 10000000	C2 80
	[APC]	U+009F	159	*00*010 011111	11*000*010 10011111	C2 9F
Texte	[NBSP]	U+00A0	160	*00*010 100000	11*000*010 10100000	C2 A0
	é	U+00E9	233	*00*011 101001	11*000*011 10101001	C3 A9
	[illegible]	U+07FF	2047	11111 111111	11011111 10111111	DF BF
	[illegible]	U+0800	2048	*0000* 100000 000000	1110*0000* 10100000 10000000	E0 A0 80
	€	U+20AC	8 364	*00*10 000010 101100	1110*0*010 10000010 10101100	E2 82 AC
	[illegible]	U+D7FF	55 295	1101 011111 111111	11101101 10011111 10111111	ED 9F BF
Demi-codets		U+D800	*(néant)*		*(codage interdit)*	
		U+DFFF				
Usage privé	[illegible]	U+E000	57 344	1110 000000 000000	11101110 10000000 10000000	EE 80 80
	[illegible]	U+F8FF	63 743	1111 100011 111111	11101111 10100011 10111111	EF A3 BF
Texte		U+F900	63 744	1111 100100 000000	11101111 10100100 10000000	EF A4 80
	[illegible]	U+FDCF	64 975	1111 110111 001111	11101111 10110111 10001111	EF B7 8F
Non-caractères		U+FDD0	64 976	1111 110111 010000	11101111 10110111 10010000	EF B7 90
		U+FDEF	65 007	1111 110111 101111	11101111 10110111 10101111	EF B7 AF
Texte	صلے	U+FDF0	65 008	1111 110111 110000	11101111 10110111 10110000	EF B7 B0
	[illegible]	U+FFFD	65 533	1111 111111 111101	11101111 10111111 10111101	EF BF BD
Non-caractères		U+FFFE	65 534	1111 111111 111110	11101111 10111111 10111110	EF BF BE
		U+FFFF	65 535	1111 111111 111111	11101111 10111111 10111111	EF BF BF
	[illegible]	U+10000	65 536	*000* 010000 000000 000000	11110*000* 10010000 10000000 10000000	F0 90 80 80

Source : wikipedia

UTF-8

Deux cas sont possibles lors de la traduction d'un code binaire en texte en fonction de la place du premier 0 :

- S'il est au début du code binaire, la première lettre du texte est encodée par 1 octet.
- S'il y a k `1` avant le premier `0`, la première lettre sera encodée par k octets. Chaque octet (sauf le premier) commence par 10.

Remarque : en UTF-8, les codes ne commencent donc jamais par `10`...

UTF-8

Version simplifiée (sans tenir compte des séquences invalides) :

0xxxxxxx	1 octet (1 à 7 bits codés)
110xxxxx 10xxxxxx	2 octets (8 à 11 bits codés)
1110xxxx 10xxxxxx 10xxxxxx	3 octets (12 à 16 bits codés)
11110xxx 10xxxxxx 10xxxxxx 10xxxxxx	4 octets (17 à 21 bits codés)

UTF-16

Codes de 2 à 4 octets.

Principalement utilisé par Java et Windows (ce qui n'empêche pas un stockage en UTF-8 plus compact).

Différentes variantes existent selon l'ordre des octets

Version simplifiée en bigendian.

xxxxxxxx xxxxxxxx	2 octets (1 à 16 bits codés)
110110yy yyxxxxxx 110111xx xxxxxxxx	4 octets (17 à 21 bits codés)

Astuce : dans le second cas, les 5 bits de poids fort du point de code correspondent à $yyyy + 1$.

UTF-32

S'utilise si vraiment il n'y a aucun problème de taille fichier.
Avantage : tous les codes ont la même taille, 32 bits.

Attention, là encore il existe différentes variantes selon l'ordre des octets. En bigendian, le code UTF-32 d'un point de code est bien le code binaire de l'entier correspondant.

Conversions et encodages de texte

Pour retrouver le point de code (« vrai codage ») d'un caractère en UTF-*n*, et donc afficher à l'écran le symbole correspondant, on applique un véritable algorithme.

De même pour encoder un point de code. Chaque transformation UTF est donc réversible.

Il est également possible de concevoir des algorithmes qui passent directement d'un format UTF à un autre sans passer par le point de code (conversion).