

# Représentation des données

---

# Représentation des données

---

## *Plan de la phase*

*Introduction*

*Code ASCII*

*Code ISO 8859*

*Code EBCDIC*

*Code UNICODE*

*Code BASE64*

*Exercices*

# Représentation des données

---

## *Introduction*

Les informations que doit traiter un ordinateur sont composées :

- nombres
- chiffres
- symboles

Or comme nous l'avons vu, les systèmes informatiques ne reconnaissent que les deux états binaires 0 et 1.

On doit alors présenter l'information à traiter, quelle quel soit, de manière à ce qu'elle puisse être utilisable par la machine .

Pour, cela on doit coder ces informations afin qu'assimilables par l'homme elles le deviennent par la machine.

# Représentation des données

## Code ASCII

- Le code **ASCII (American Standard Code Information Interchange)** est le code le plus **ancien** des codes informatique.
- Normaliser** ISO 7 bits, nommé aussi US-ASCII ou ASCII Standard.
- Code sur **7 bits (128 symboles)**, c'est le charset, définit au départ pour coder les caractères anglo-saxon.

### Utilisation du tableau :

Pour coder la lettre A en ASCII :

- colonne 4
- ligne 1

Donc le code ASCII de la lettre A est **41H**

En décimal se serait **65**.

Le maintien de la touche **Alt** et la frappe de la valeur décimal du caractère recherché permet d'atteindre ce caractère.

Par exemple **Alt 65** donne la lettre

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

Quel est le caractère codé par **0x2A** ?

Certains **caractères** sont **non imprimables** mais utilisés lors de **transmission** de données entre ordinateur – ordinateur ou ordinateur - imprimante. (**EOT**, **ENQ**, **ACK**)

# Représentation des données

## Code ASCII - étendu

(1970) Pour permettre l'utilisation de caractères accentués (étendus) utilisés dans les langues différentes de l'anglais, le codage est actuellement sur 8 bits et permet 256 caractères différents. Ce code ASCII étendu est à la base de tous les codes utilisés aujourd'hui. De nombreuses langues comportent des symboles qu'il est impossible de résumer en 256 caractères. Il existe pour cette raison des variantes de code ASCII incluant des caractères et symboles régionaux.

128	Ç	144	É	161	í	177	☐	193	⊥	209	〒	225	β	241	±
129	ü	145	æ	162	ó	178	☐	194	⊤	210	⊥	226	Γ	242	≥
130	é	146	Æ	163	ú	179		195	⊥	211	⊥	227	π	243	≤
131	â	147	ô	164	ñ	180	⊥	196	—	212	⊥	228	Σ	244	∫
132	ä	148	ö	165	Ñ	181	⊥	197	+	213	⊥	229	σ	245	∫
133	à	149	ò	166	²	182	⊥	198	⊥	214	⊥	230	μ	246	+
134	å	150	û	167	°	183	⊥	199	⊥	215	⊥	231	τ	247	≈
135	ç	151	ù	168	¿	184	⊥	200	⊥	216	⊥	232	Φ	248	°
136	ê	152	—	169	—	185	⊥	201	⊥	217	⊥	233	⊙	249	·
137	ë	153	Ö	170	⊥	186	⊥	202	⊥	218	⊥	234	Ω	250	·
138	è	154	Ü	171	½	187	⊥	203	⊥	219	■	235	δ	251	√
139	ï	156	£	172	¼	188	⊥	204	⊥	220	■	236	∞	252	—
140	î	157	¥	173	¡	189	⊥	205	=	221	■	237	φ	253	²
141	ì	158	—	174	«	190	⊥	206	⊥	222	■	238	ε	254	■
142	Ä	159	f	175	»	191	⊥	207	⊥	223	■	239	∩	255	
143	Å	160	á	176	☐	192	⊥	208	⊥	224	α	240	≡		

# Représentation des données

## Code ISO 8859

Pour coder les **caractères** utilisés dans **toutes** les **langues**, il a fallu trouver autre chose.  
**L'ISO à créé** la gamme des codes s'étend de **ISO 8859-1 à ISO 8859-16**.  
 Norme utilisée par de nombreux système d'exploitation (Unix, Windows).

### ISO 8859-xx :

- Reprend le code ASCII
- Extension pour chaque langue au moyen des caractères supérieur à 128.
- Les caractères de 0 à 31 (comme en ASCII) sont des caractères de contrôle.
- Les caractères de 128 à 159 sont réservés : les constructeurs utilisent cette plage pour leurs codes (**note 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	ENO	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	ı	ç	£	€	¥	—	§	"	©	"	«	¬		®	™
Bx	°	±	—	—	ˆ	µ	¶	·	,	—	°	»	—	—	—	¿
Cx	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
Dx	—	Ñ	Ò	Ó	Ô	Õ	—	Ö	Ù	Ú	Û	Ü	—	—	—	ß
Ex	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
Fx	—	ñ	ò	ó	ô	õ	÷	ø	ù	ú	û	ü	—	—	—	ÿ

### ISO 8859-1 (Latin-1) :

Etend ASCII standard avec les caractères accentués utiles aux langues européennes.

ISO 8859-15 (Latin 9) ajoute le caractère de l'euro (€).

Les codes ISO 8859-xx permettent de gérer qu'un alphabet à la fois.

# Représentation des données

---

## *EBCDIC*

L'Extended Binary Coded Decimal Interchange Code (EBCDIC) est un mode de codage des caractères sur 8 bits créé par IBM à l'époque des cartes perforées. Il existe au moins 6 versions différentes bien documentées (et de nombreuses variantes parfois créées par des concurrents d'IBM), incompatibles entre elles. Ce mode de codage a été critiqué pour cette raison, mais aussi parce que certains caractères de ponctuation ne sont pas disponibles dans certaines versions. Ces disparités ont parfois été interprétées comme un moyen pour IBM de conserver ses clients captifs. EBCDIC est encore utilisé dans les systèmes AS/400 d'IBM ainsi que sur les mainframes sous MVS, VM ou DOS/VSE.

# TODO

# Représentation des données

---

## *Unicode(note 1)*

**ISO travaille** depuis 1988 sur un **code « universel »** (UNiversal CODE).  
Ces travaux référencés sous **ISO/IEC 10 646** se **présentent** sous **deux formes** :

- **32 bits** (USC-4)
- **16 bits** (USC-2)

Unicode 1.0.0 :

- Sous ensemble de départ conforme à la norme ISO 10646 (UCS-2).
- codage sur 16 bits (65 535 caractères).

Unicode 2.0 [1993] :

- recense 38 885 caractères conforme à la norme ISO/IEC 10646-1:1993.
- codage sur 16 bits (65 535 caractères).

Unicode 3.0 [2000] :

- recense 49 194 caractères conforme à la norme ISO/IEC 10646-1:2000.
- couvre les langages : États-Unis, Europe, Moyen-Orient, Afrique, Inde, Asie et Pacifique
- la version Unicode 3.1 introduit 44 946 nouveaux symboles (**note 2**)
- la version 3.2 [2003] code 95 221 caractères

Unicode 5.1 [2008] :

- 100,713 caractères

Unicode 6.0 [11 octobre 2010] :

- 109449 caractères

Unicode 6.0 [ 2012] :

- dernière version
- 110116 caractères

Toute application conforme à Unicode est donc conforme à l'ISO/CEI 10 646.



# Représentation des données

## Unicode

Unicode se traduit sous trois formes :

- **UTF-8** (RFC 3629) :
  - codage de taille variable (moins coûteux en mémoire)
  - prend en charge ASCII étendus et UCS-2 (Unicode sur 16 bits)
  - utilisé pour les applications Unix et Internet
- **UTF-16**
  - la majorité des caractères peuvent être représentés.
- **UTF-32** synonyme d'UCS-4
  - code les caractères sur une même taille.



	A	κ	好	不
Code point	U+0041	U+05D0	U+597D	U+233B4
UTF-8	41	D7 90	E5 A5 BD	F0 A3 8E B4
UTF-16	00 41	05 D0	59 7D	D8 4C DF B4
UTF-32	00 00 00 41	00 00 05 D0	00 00 59 7D	00 02 33 B4

- Les premiers 65,536 caractères forment le PMB (Plan Multilingue de Base).
- Près de 1 millions de caractères sont référencés dans le jeu de caractères Unicode appelés supplementary characters.

# Représentation des données

## Unicode - Fonctionnement de UTF-8

### Fonctionnement de UTF-8

Chaque **caractères** Unicode est **repéré** par un **nombre (point de code)** compris entre **0 et  $2^{31}$**  (0x0 – 0x7FFFFFFF) et un **nom**.

- UTF-8 est un **code variable**, on transmet 8 bits, si le caractères peut être codé sur 8 bits (ASCII) et non pas 31 bits comme avec UTF-32.
- Choisie pour l'internationalisation des protocoles d'Internet.
- Les bits de poids fort d'un octet indique la position de celui-ci dans la suite.
- Le premier octet permet de savoir si la suite est composé de 2 ou 3... octets.

L'espace total de codage a été découpé en pages :

#### - **La première page** :

- code les caractères ASCII
- représenté sur 1 octet, le premier toujours à 0 indique la première page

#### - **La deuxième page** :

- code les caractères accentués
- codés sur 2 octets
- les 3 bits de poids fort du 1er octet sont positionnés à 110
  - permet de connaître la page.
- les 2 bits de poids fort du 2 octet sont positionnés à 10
  - indique qu'ils appartiennent a une suite d'octet.

#### - **La troisième page** :

- code les caractères « exotiques » sur 3 octets.

#### - **La dernière page** :

- occupe 6 octets.

Caractères Unicode		Octets en UTF-8
0-127	0x0000 0000 – 0x0000 007F	0xxxxxxx
128-2047	0x0000 0080 – 0x0000 07FF	110xxxxx 10xxxxxx
2048-65535	0x0000 0800 – 0x0000 FFFF	1110xxxx 10xxxxxx 10xxxxxx
	0x0001 0000 – 0x0001 FFFF	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx
	0x0020 0000 – 0x03FF FFFF	111110xx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx
	0x0400 0000 – 0x7FFF FFFF	1111110x 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx

# Représentation des données

## Unicode – Fonctionnement de UTF-8

**Exemple :** pour écrire le mot « déjà » on utilisera la séquence d'octets suivante :  
**0x64 C3 A9 6A C3 A0**

	Hexadécimal	1 <sup>er</sup> octet	2 <sup>e</sup> octet
d	0x64	0110 0100	
é	0xC3 – 0xA9	1100 0011	1010 1001
j	0x6A	0110 1010	
à	0xC3 – 0xA0	1100 0011	1010 0000

- « d » caractère de la 1er plage : l'octet commence par un 0.
- « è » caractère de la 2e plage : il nous faut 2 octets.
  - 1er octet commence par 110 indique :
    - un octet suit.
    - caractère de 2e plage.
    - les bits qui restent code le caractère (00000)
  - 2 caractère commence par 10 indique :
    - que c'est un octet de suite
    - les bits qui restent code le caractère (000000)

Il faut placer la valeur codante, dans le tableau la lettre « è » soit 0xE9 ou 1110 1001 dans la suite binaire  
 00000 000000.

2<sup>e</sup> octet : 1010 1001 → 0xA9

1<sup>er</sup> octet : 1100 0011 → 0xC3

on trouve de même pour « j » = 0x6A et pour « à » = 0xC3A0  
 Donc « déjà » est codé sur 6 octets. (comparer au 16 octets pour un code UTF-32)

000	001	002	003	004	005	006	007	008	009	00A	00B	00C	00D	00E	00F
0	NUL	SOH	STX	ETX	SP	0	@	P	`	p					
1	DEL	CD1	!	1	A	Q	a	q							
2	DTX	CD2	"	2	B	R	b	r							
3	FTX	CD3	#	3	C	S	c	s							
4	FTR	CD4	\$	4	D	T	d	t							
5	DEM	ACN	%	5	E	U	e	u							
6	ACC	SYN	&	6	F	V	f	v							
7	SDN	FMT	'	7	G	W	g	w							
8	EFF	ANN	(	8	H	X	h	x							
9	TAB	FS	)	9	I	Y	i	y							
A	PAL	SUB	*	:	J	Z	j	z							
B	TAV	ECH	+	;	K	[	k	{							
C	SDP	SP	,	<	L	\	l								
D	RC	SG	-	=	M	]	m	}							
E	HC	SA	.	>	N	^	n	~							
F	EC	SSA	/	?	O	_	o	sup							

Unicode Jeu de base Latin Unicode compléments

# Représentation des données

## *Unicode – Fonctionnement de UTF-8*

```
HTTP/1.1 200 OK
Date: Wed, 05 Nov 2003 10:46:04 GMT
Server: Apache/1.3.28 (Unix) PHP/4.2.3
Content-Location: CSS2-REC.en.html
Vary: negotiate,accept-language,accept-charset
TCN: choice
P3P: policyref=http://www.w3.org/2001/05/P3P/p3p.xml
Cache-Control: max-age=21600
Expires: Wed, 05 Nov 2003 16:46:04 GMT
Last-Modified: Tue, 12 May 1998 22:18:49 GMT
ETag: "3558cac9;36f99e2b"
Accept-Ranges: bytes
Content-Length: 10734
Connection: close
Content-Type: text/html; charset=utf-8
Content-Language: en
```

<http://www.w3.org/International/O-HTTP-charset.fr.php>

**Voir note**

**Faire démo : wireshark**

# Représentation des données

---

## *Exercices*

1. Le vidage d'un fichier fait apparaître les information suivantes en ASCII :  
4C 65 20 42 54 53 20 65 73 74 20 75 6E 65 20 22 65 78 63 65 6C 6C 65 6E 74 65 22 20 46 6F 72  
6D 61 74 69 6F 6E 2E.  
Procéder à leur conversion en texte.
2. Coder le texte suivant en utilisant le code ASCII sous sa forme hexadécimale :  
Technologie 2003 « Leçon sur les CODES ».
3. Coder en Unicode UTF-8, sous forme hexadécimale, le texte : « Année 2007 ».

# Représentation des données

---

## *Exercices*

Le type **MIME** (*Multipurpose Internet Mail Extension*) est un standard ( 1991) qui propose pour transférer les données cinq formats de codage. **MIME** s'appuie entre autre, sur un code particulier dit « base64 »

- Définissez le rôle du protocole MIME.
- Quels sont les protocoles de communications utilisateurs de MIME ?
- Comment s'effectue le codage des données en base 64 ?

**Exemple** : supposons que nous ayons à transmettre l'extrait du fichier suivant : **0xE3 85 83...** Ces valeurs hexadécimales pouvant correspondre à une vidéo, un fichier mp3... Donner en hexadécimal les valeurs transmises.