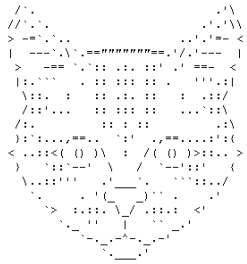
5

ReprÃ©sentation des textes

Extrait du programme

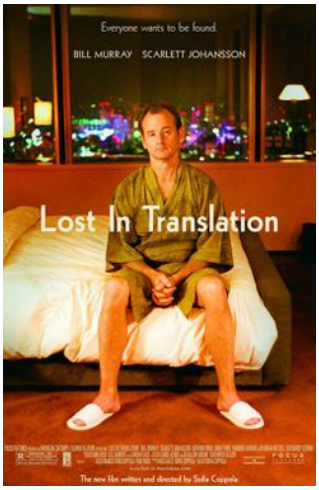
ThÃ¨me : Types et valeurs de base



Contenus : ReprÃ©sentation d'un texte en machine.
Exemples des encodages ASCII, ISO-8859-1, Unicode

CapacitÃ©s attendus : Identifier l'intÃ©rÃªt des diffÃ©rents systÃ¨mes d'encodage.
Convertir un fichier texte dans diffÃ©rents formats d'encodage.

Commentaires : Aucune connaissance prÃ©cise des normes d'encodage n'est exigible.



Mais pourquoi ai-je reÃ§u le message suivant ? :

```
Ceci est un texte accentuÃ© enregistrÃ© en utf8 avec des Ã© des Ã¨ des Ã§ des Ã  et mÃªme des Ã¹
```

I. Codage ASCII

a) Code ASCII

Dans les annÃ©es 1950, il existait un nombre important d'encodages de caractÃ¨res dans les ordinateurs, les imprimantes ou les lecteurs de carte. Tous ces encodages Ã©taient incompatibles les uns avec les autres ce qui rendait les Ã©changes particuliÃ¨rement difficiles car il fallait utiliser des programmes pour convertir les caractÃ¨res d'un encodage dans un autre.
Au dÃ©but des annÃ©es 1960, l'*ANSI* (American National Standards Institue) propose une norme de codage de caractÃ¨res appelÃ©e ***ASCII*** (American Standard Code for Information Interchange). Cette norme dÃ©finit un jeu de 128 caractÃ¨res. En effet, Ã l'Ã©poque les ordinateurs fonctionnaient en 8 bits, et, 1 bit de paritÃ© Ã©tant conservÃ© pour la dÃ©tection des erreurs de transmission, soit $2^7 = 128$ caractÃ¨res, ce qui Ã©tait trÃ¨s largement suffisant pour coder les lettres majuscules, minuscules, chiffres et ponctuations.
Le document ci-dessous date de 1972 (*source WikipÃ©dia*)

USASCII code chart

$b_7 b_6 b_5$ → / $b_4 b_3 b_2 b_1$ ↓	000	001	010	011	100	101	110	111
	0	1	2	3	4	5	6	7
0000	0	NUL	DLE	SP	0	@	P	`	p
0001	1	SOH	DC1	!	1	A	Q	a	q
0010	2	STX	DC2	"	2	B	R	b	r
0011	3	ETX	DC3	#	3	C	S	c	s
0100	4	EOT	DC4	$	4	D	T	d	t
0101	5	ENQ	NAK	%	5	E	U	e	u
0110	6	ACK	SYN	&	6	F	V	f	v
0111	7	BEL	ETB	'	7	G	W	g	w
1000	8	BS	CAN	(	8	H	X	h	x
1001	9	HT	EM	)	9	I	Y	i	y
1010	10	LF	SUB	*	:	J	Z	j	z
1011	11	VT	ESC	+	;	K	[	k	{
1100	12	FF	FS	,	<	L	\	l	|
1101	13	CR	GS	-	=	M	]	m	}
1110	14	SO	RS	.	>	N	^	n	~
1111	15	SI	US	/	?	O	_	o	DEL

Pour lire le codage de chaque caractÃ¨re, on commence par rÃ©cupÃ©rer le code binaire de la colonne correspondante, auquel on concatÃ¨ne [1] le code binaire de la ligne correspondante.
Ainsi, pour le caractÃ¨re `A`, on est sur la colonne `100` et sur la ligne `0001`. Le code binaire ASCII du `A` est donc `100 0001`. Soit en dÃ©cimale : $1 \times 2^6 + 0 \times 2^5 + ... + 0 \times 2^1 + 1 \times 2^0 = 65$.
On peut faire de mÃªme pour le codage en hexadÃ©cimal du caractÃ¨re `A`. On est sur la colonne `4` et sur la ligne `1`. Le codage hexadÃ©cimal du caractÃ¨re `A` est donc `41`, soit en dÃ©cimale : $4 \times 16^1 + 1 \times 16^0 = 65$.

Exercice 1
Donner le codage binaire, hexadÃ©cimal puis dÃ©cimal des caractÃ¨res suivants :

1. **a** :
   - binaire .....
   - hexadÃ©cimal .....
   - dÃ©cimal .....

1. Si, si, ce mot existe.

2. **9** :
   - binaire .....
   - hexadÃ©cimal .....
   - dÃ©cimal .....

Dans la suite, nous ne donnerons que les codages en hexadÃ©cimal, ce qui est beaucoup plus court et pratique.
On remarque que la table ***ASCII*** contient plusieurs catÃ©gories de caractÃ¨res :

- les chiffres de 0 Ã 9 (entre `30` et `39`)
- les lettres de l'alphabet latin en majuscules (entre `41` et `5A`)
- les lettres de l'alphabet latin en minuscules (entre ..... et .....)
- des signes de ponctuations (comme la virgule qui vaut `2C`, le crochet [qui vaut .....)
- des signes opÃ©rateurs arithmÃ©tiques (comme le + qui vaut `2B`, le = qui vaut .....)
- des caractÃ¨res spÃ©ciaux (entre `00` et `20`)

b) CarcatÃ¨res spÃ©ciaux

Voici quelques dÃ©tails sur certains caractÃ¨res spÃ©ciaux :

CaractÃ¨re	Code hexa	Signification
HT	09	Tabulation horizontale
LF	0A	Nouvelle ligne
CR	0D	Retour chariot
FF	0C	Nouvelle page
SP	20	Espace
BS	...	Suppression
DEL	7F	Effacement

Exercice 2
DÃ©coder le message suivant Ã©crit en code ASCII suivants donnÃ©s en hexadÃ©cimal :
4F 75 69 20 43 48 45 46 21 (0D) 0A 42 69 65 6E 20 43 48 45 46 21
.....
.....

c) Python

En Python, on peut obtenir directement le code ***ASCII*** en dÃ©cimal d'un caractÃ¨re avec la fonction `ord()` et le caractÃ¨re correspondant Ã un code avec la fonction `chr()` :

```
>>> ord('A')
65
>>> bin(ord('A')) #code en binaire
'0b1000001'
>>> hex(ord('A')) #code en hexadÃ©cimal
'0x41'
>>> chr(65)
'A'
>>> chr(0b1000001) #en binaire
'A'
>>> chr(0x41) #en hexadÃ©cimal
'A'
```

On peut aussi Ã©crire directement en hexadÃ©cimal en rajoutant \x devant le code en hexadÃ©cimal. On appelle cette technique la technique du caractÃ¨re Ã©chappÃ©e car le caractÃ¨re \ s'appelle aussi le caractÃ¨re d'Ã©chappement.

```
>>> print('\x4F\x75\x69\x20\x43\x48\x45\x46\x21\x0A\x42\x69\x65\x6E\x20\x43\x48\x45\
    x46\x21')
Oui CHEF!
Bien CHEF!
```

Il existe quelques raccourcis pour les caractÃ¨res spÃ©ciaux. Le tableau ci-dessous en propose quelques-un :

Raccourcis	CaractÃ¨re	Signification
\t	HT	Tabulation horizontale
\n	LF	Nouvelle ligne
\r	CR	Retour chariot
\f	FF	Nouvelle page
\b	BS	Suppression

Ainsi, on a :

```
>>> print('Biem\bn!\npetit\tscarabÃ©e')
Bien!
petit    scarabÃ©e
```

II. Les normes ISO 8859

Les caractÃ¨res imprimables de la table ***ASCII*** se sont vites avÃ©rÃ©s insuffisants pour transmettre des textes dans les autres langues que l'anglais. En effet, rien qu'en considÃ©rant les langues reposant sur un alphabet latin, il manque dans la table ***ASCII*** de nombreux caractÃ¨res comme les lettres accentuÃ©es (Ã©, Ã¨, Ã , Ã¹, ...), *les symboles de monnaies* (e, ...). *Pour rem©dier ce prob*Ene, *l'* ***ISO*** (*Organisation de Normalisation*) *apropos© la norme* ***ISO 8859***, *une extension de l'* ***ASCII*** *qu les* 128 *premiers caracE ressont ceux de la norme* ***ASCII***; *les* 128 *suivants sont ceux sp©cifiques la table*n. *Les caracE resident iques o*

Code ISO	Nom	Zone
8859-1	latin-1	Europe occidental
8859-2	latin-2	Europe centrale ou de l'est
8859-3	latin-3	Europe du sud
8859-4	latin-4	Europe du nord
8859-5		Cyrillique
8859-6		Arabe
8859-7		Grec
8859-8		HÃ©breu
8859-9	latin-5	Turc, Kurde
8859-10	latin-6	RÃ©vision du latin-4
8859-11		ThaÃ¯
8859-12		Devanagari[3] (projet abandonnÃ©)
8859-13	latin-7	Balte
8859-14	latin-8	Celtique
8859-15	latin-9	RÃ©vision du latin-1 (avec €)
8859-16	latin-10	Europe du sud-est

2. Il y en a 16 en tout, dont 10 uniquement pour les langues latines.
3. Ãcriture utilisÃ©e pour le sanskrit, le prÃ¢krit, le hindi, le nÃ©palais, le marathi et plusieurs autres langues indiennes.