

Chapitre 6 : Représentation d'un texte

1. [Algèbre de Boole](#) [Voir chapitre 1]

L'**algèbre booléenne** (ou **algèbre de Boole**), est la partie des mathématiques, de la logique et de l'électronique qui s'intéresse à une approche algébrique de la logique, vue en termes de variables : elle concerne les opérations et les fonctions utilisant des variables logiques. Elle fut lancée en 1854 par le mathématicien britannique George Boole. En Python, ces variables logiques sont de type « **booléen** ».

1.1. [Les tables de vérité](#)

▪ L'opérateur « ET »

L'opérateur booléen « **and** » (« et » en français) est un opérateur de conjonction (opération liant deux valeurs logiques, généralement les valeurs de deux propositions), noté « & », « AND », « · » ou « $\wedge$ » selon les langages.

Cette conjonction est définie par : **a ET b est VRAI si et seulement si a est VRAI et b est VRAI.**

a	b	a and b
False	False	False
False	True	False
True	False	False
True	True	True

▪ L'opérateur « OR »

L'opérateur booléen « **or** » (« ou » en français) est un opérateur de disjonction (opération comparant deux valeurs logiques), noté « | », « OR », « + » ou « $\vee$ » selon les langages.

Cette disjonction est définie par : **a OU b est VRAI si et seulement si a est VRAI ou b est VRAI.**

a	b	a or b
False	False	False
False	True	True
True	False	True
True	True	True

→ Si a est vrai et que b est vrai aussi, alors a OU b est vrai.

▪ L'opérateur « NOT »

L'opérateur booléen « **not** » (« non » en français) est un opérateur de négation, noté (pour un opérateur a) « non- a », « non a », « not a », « $\sim a$ », « $\bar{a}$ » ou « ! » selon les langages.

Cette négation est définie par : **la négation de a est VRAIE si et seulement si a est FAUX ou la négation de a est FAUX si et seulement si a est VRAI.**

a	not a
False	True
True	False

▪ L'opérateur « XOR »

L'opérateur booléen « **xor** » (« ou exclusif » en français) est un opérateur de disjonction exclusive ($XOR \Leftrightarrow eXclusive OR$), noté (pour un opérateur a) « $\oplus$ », « $\underline{\vee}$ », « not a », « $\sim a$ », « $\bar{a}$ » ou « ! » selon les langages.

Cette négation exclusive est définie par : **a XOR b est VRAIE si et seulement si a ou b est VRAI mais pas les deux.**

a	b	a or b
False	False	False
False	True	True
True	False	True
True	True	False

1.2. Les expressions booléennes

Comme en mathématiques, il faut tenir compte des parenthèses et des priorités des opérateurs booléens pour évaluer une expression booléenne : les opérateurs booléens prioritaires sont « **not** » puis « **and** » en enfin « **or** » ou « **xor** ».

Exemple :

Évaluer la valeur booléenne : **A = True and True or (not False and False)**

Démarche :

❶ On commence par évaluer ce qui est entre parenthèses : **not False** vaut **True**

True and False vaut **False**

⇒ **A = True and True or False**

❷ On évalue ensuite le reste en utilisant les priorités des opérateurs : **True and True** vaut **True**

True or False vaut **True**

⇒ **A = True**

On peut aussi dresser une table de vérité pour l'expression booléenne à trouver.

Exemple : Dresser la table de vérité de l'expression « **a or b and not a** » ?

Priorités : « not a » puis « b and » puis « a or ».

a	b	not a	b and (not a)	a or (b and not a)
False	False	True	False	False
False	True	True	True	True
True	False	False	False	True
True	True	False	False	True

Remarque : dans certains cas, l'interpréteur Python peut court-circuiter une partie d'une expression booléenne qui se situe à la droite d'un opérateur booléen si l'évaluation de cette expression (de droite) ne change rien au résultat.

```
1 >>>def test(retour, texte):
2     print(texte)
3     return retour
4
5 >>> a = test(True, "1") or test(False, "2")
6 1
```

→ L'interpréteur Python affiche le retour du premier appel à la fonction « `test()` », qui est « 1 », mais pas le second qui serait « 2 ».

En effet, le premier appel à la fonction « `test()` » renvoie « `True` », donc l'interpréteur Python n'évalue pas l'expression après l'opérateur booléen « `or` » car si c'est « `False` » alors le résultat de « `True or False` » sera « `True` » et si c'est « `True` » alors le résultat de « `True or True` » sera « `True` » donc la valeur booléenne après l'opérateur « `or` » ne change rien au résultat de l'opération de disjonction booléenne : c'est celui du premier test.

2. Représentation d'un texte en machine

2.1. La norme ASCII

Le codage américain **ASCII** est une norme de codage de caractères en informatique, ancienne et connue pour son influence incontournable sur les codages de caractères qui lui ont succédé. Elle était la plus largement compatible pour ce qui est des caractères latins non accentués.

La norme d'encodage ASCII contient les caractères nécessaires pour écrire en langue anglaise mais l'absence des caractères des langues étrangères à l'anglais, notamment les caractères accentués, rend ce standard insuffisant à lui seul pour des textes étrangers (par exemple en langue française), ce qui rend nécessaire l'utilisation d'autres encodages.

L'ASCII définit seulement **128 caractères** numérotés de **0 à 127** et codés en binaire de 0000000 à 1111111 :

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

⇒ Sept bits ($2^7 = 128$) suffisent donc pour représenter un caractère codé en ASCII. Toutefois, les ordinateurs travaillant presque tous sur un multiple de huit bits (multiple d'un octet) depuis les années 1970, chaque caractère d'un texte en ASCII est donc souvent stocké dans un octet : le 8^e bit est 0.

Le jeu de caractères ASCII comprend :

- Les 26 lettres de l'alphabet latin en versions majuscules et minuscules ;
- Les 10 chiffres ;
- Les symboles de ponctuation y compris l'espace ;
- Les couples de parenthèses, crochets, accolades et quelques autres symboles divers.

→ **C'est donc un encodage simple mais limité en caractères.**

2.2. [La norme ISO-8859-1](#)

La norme **ISO 8859-1**, qui est souvent appelée Latin-1 ou Europe occidentale, forme la première partie de la norme internationale ISO/CEI 8859, qui est une norme de l'*Organisation internationale de normalisation* pour le codage des caractères en informatique.

Cette norme définit ce qu'on appelle l'alphabet latin numéro 1, qui consiste en 191 caractères de l'alphabet latin, chacun d'entre eux étant **codé sur un octet** (soit **8 bits**). La norme d'encodage ISO 8859-1 reprend le codage des caractères imprimables ASCII.

[Source : Wikipédia]

→ Elle permet d'encoder tous les caractères des principales langues européennes.

2.3. [La norme Unicode UTF-8](#)

Unicode est un standard informatique qui **permet d'encoder tous les caractères de toutes les langues**, il est universel et extensible. Il est développé par le Consortium Unicode, une organisation privée sans but lucratif qui coordonne le développement du standard Unicode.

La table complète des caractères Unicode peut être consultée sur le site <http://unicode-table.com>. Elle est constituée d'un répertoire de 144 697 caractères : tout caractère de n'importe quel système d'écriture possède un nom et un identifiant numérique, et ce de manière unifiée, quels que soient la plate-forme informatique ou le logiciel utilisé.

→ Ce standard nécessite l'utilisation de **4 à 6 octets** pour encoder un caractère.

L'encodage **UTF-8** est un encodage de type Unicode qui permet d'encoder tous les caractères de toutes les langues, il est actuellement en train de s'imposer face aux autres encodages en raison de son universalité. Cet encodage est de taille variable, ce qui lui permet d'être moins gourmand en espace mémoire qu'Unicode.

Remarques :

- Les 128 premiers caractères de la table UTF-8 sont compatibles avec le codage ASCII : le codage UTF-8 d'un texte ne comportant que des caractères présents dans la table ASCII sera le même que le codage ASCII de ce texte ;
- En python 3, tous les fichiers doivent être codés en dans le système d'encodage UTF-8, et les chaînes de caractères string sont encodées à la norme UTF-8, contrairement aux bytes qui ne doivent contenir que des caractères à la norme ASCII ;
- Pour obtenir un affichage correct des caractères spéciaux dans une page html, il faut déclarer l'encodage UTF-8 dans l'en-tête du fichier html (section `<head>`) en utilisant la balise `<meta>`.

Programme :

Contenus	Capacités attendues	Commentaires
Valeurs booléennes : 0, 1. Opérateurs booléens : and, or, not. Expressions booléennes	Dresser la table d'une expression booléenne.	Le ou exclusif (xor) est évoqué. Quelques applications directes comme l'addition binaire sont présentées. L'attention des élèves est attirée sur le caractère séquentiel de certains opérateurs booléens.
Représentation d'un texte en machine. Exemples des encodages ASCII, ISO-8859-1, Unicode	Identifier l'intérêt des différents systèmes d'encodage. Convertir un fichier texte dans différents formats d'encodage.	Aucune connaissance précise des normes d'encodage n'est exigible.

Sitographie :

- [Algèbre de Boole \(logique\)](#) (Wikipedia.org)
- <https://www.lyceum.fr/1g/nsi/2-representation-des-donnees-types-et-valeurs-de-base/4-representation-des-textes>
- <https://fr.wikipedia.org/wiki/UTF-8>
- <https://fr.wikipedia.org/wiki/Unicode>