

Le codage des caractères est une convention qui permet, à travers un codage connu de tous, de transmettre de l'information textuelle, là où aucun support ne permet l'écriture scripturale.

1.ASCII

L'**American Standard Code for Information Interchange** (Code américain normalisé pour l'échange d'information), plus connu sous l'**acronyme ASCII** ([\[aski:\]](https://fr.wikipedia.org/wiki/ASCII)), est une **norme informatique** de **codage de caractères** apparue dans les **années 1960**. C'est la norme de codage de caractères la plus influente à ce jour. Code ASCII de base :

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	:	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Source : www.LookUpTables.com

Code ASCII étendu (valeur hexadécimales lignes dizaines colonnes unités)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STH	ETH	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	CD2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL
8	€	□	,	f	„	…	†	‡	ˆ	%o	Š	<	Œ	□	Ž	□
9	□	'	'	"	"	•	-	—	˜	™	š	>	œ	□	ž	ÿ
A		ı	¢	£	×	¥	ı	§	"	©	ª	«	¬	-	®	¯
B	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Ex Donner la valeur du code ASCII de caractère « A » :

Quelle relation existe-t-il entre les caractères latins minuscule et majuscule ?

2.UNICODE UTF-8

Unicode est un standard informatique qui permet des échanges de textes dans différentes langues, à un niveau mondial. Unicode se contente de recenser, nommer les caractères et leur attribuer un numéro. Mais il ne dit pas comment ils doivent être codés en informatique.

Plusieurs codages des caractères Unicode existent :

- UTF-32 qui code chaque caractère sur 32 bits (soit quatre octets)
- UTF-16 qui code chaque caractère sur 16 ou 32 bits (soit deux ou quatre octets)
- UTF-8 qui code chaque caractère sur 8, 16, 24 ou 32 bits (soit un, deux, trois ou quatre octets).

Le plus couramment utilisé, notamment pour les pages Web, Python est UTF-8.

La principale caractéristique d'UTF-8 est qu'elle est rétro-compatible avec le standard ASCII, c'est-à-dire que **tout caractère ASCII se code en UTF-8 sous forme d'un unique octet, identique au code ASCII**. Par exemple « A » (A majuscule) a pour code ASCII 65 et se code en UTF-8 par l'octet 65. Chaque caractère dont le point de code est supérieur à 127 (caractère non ASCII) se code sur 2 à 4 octets. Le caractère « € » (euro) se code par exemple sur 3 octets : 226, 130, et 172.

Tableau récapitulatif de la concordance entre les codes UNICODE et UTF-8

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	11100000 10xxxxxx 10xxxxxx	E0 (le 2 ^e octet est restreint de A0 à BF)	3 octets, codant 16 bits
U+1000 à U+1FFF	11100001 10xxxxxx 10xxxxxx	E1	
U+2000 à U+3FFF	1110001x 10xxxxxx 10xxxxxx	E2 à E3	
U+4000 à U+7FFF	111001xx 10xxxxxx 10xxxxxx	E4 à E7	
U+8000 à U+BFFF	111010xx 10xxxxxx 10xxxxxx	E8 à EB	
U+C000 à U+CFFF	11101100 10xxxxxx 10xxxxxx	EC	
U+D000 à U+D7FF	11101101 10xxxxxx 10xxxxxx	ED (le 2 ^e octet est restreint de 80 à 9F)	
U+E000 à U+FFFF	1110111x 10xxxxxx 10xxxxxx	EE à EF	4 octets, codant 21 bits
U+10000 à U+1FFFF	11110000 1001xxxx 10xxxxxx 10xxxxxx	F0 (le 2 ^e octet est restreint de 90 à BF)	
U+20000 à U+3FFFF	11110000 101xxxxx 10xxxxxx 10xxxxxx	F1	
U+40000 à U+7FFFF	11110001 10xxxxxx 10xxxxxx 10xxxxxx	F2 à F3	
U+80000 à U+FFFFFF	1111001x 10xxxxxx 10xxxxxx 10xxxxxx	F4 (le 2 ^e octet est restreint de 80 à 8F)	
U+100000 à U+10FFFF	11110100 1000xxxx 10xxxxxx 10xxxxxx		

Convertir de l'UNICODE en UTF-8

Soit le code UNICODE U+ XXXX qui donne D₁₅D₁₄D₁₃D₁₂ D₁₁D₁₀D₉D₈ D₇D₆D₅D₄ D₃D₂D₁D₀ en binaire

1. Si le code UNICODE est inférieur ou égal à U+007F -*caractères ASCII standards*-
Le code UTF-8 est l'octet du code ASCII il n'y a aucun changement, il commence par un zéro.

$$0 D_6 D_5 D_4 D_3 D_2 D_1 D_0$$

2. Si le code UNICODE est compris entre U+0080 et U+07FF compris
Le code UTF-8 est reparti sur deux octets. Le 1^{er} octet commence par 110 et le 2^{ème} par 10,

$$110 D_{10} D_9 D_8 D_7 D_6 \quad 10 D_5 D_4 D_3 D_2 D_1 D_0$$

Exemple : Pound Sign £ U+00A3 = 0000 0000 1010 0011

U+0080 <U+00A3<U+07FF alors UTF 8 = 1100 0010 1010 0011 = 0xC2 A3

3. Si le code UNICODE est compris entre U+0800 et U+FFFF compris

Le code UTF-8 est reparti sur trois octets. Le 1^{er} octet commence par 1110, le 2^{ème} par 10, le 3^{ème} par 10

1110 D₁₅D₁₄D₁₃D₁₂ 10 D₁₁D₁₀D₉D₈ D₇D₆ 10 D₅D₄ D₃D₂D₁D₀

Exemple : Comment coder U+2639 qui donne 0010 0110 0011 1001 en binaire

U+2636 est supérieur à U+07FF, le codage en UTF-8 sera sur 3 octets

1110 D₁₅D₁₄D₁₃D₁₂ 10 D₁₁D₁₀ D₉D₈D₇D₆ 10 D₅D₄ D₃D₂D₁D₀

1110 0 0 1 0 10 0 1 1 0 0 0 10 1 1 1 0 0 1

E 2 9 8 B 9

0xE2 0x98 0xB9 Il s'agit de ce symbole : ☹ <http://unicode-table.com/fr/#control-character>

3. Encodage en python

Pour imposer le codage des caractères utf-8 pour le programme en python

```
# -*- coding: utf-8 -*-  
# coding: utf-8
```

En Python, les chaînes de caractères sont en réalité codées en UTF-8, un code (ou encodage) plus général que l'ASCII et qui permet de représenter plus de caractères, par exemples les caractères chinois. En UTF-8, Les caractères d'indices inférieurs à 128 sont exactement les mêmes qu'en ASCII : le tableau ci-dessus

```
fonctionne donc. """  
valeur ASCII d'un caractère  
ord() renvoie la valeur unicode d'un caractère  
l'unicode commence par les caractères ASCII qui gardent leurs valeurs  
chr() renvoie le caractère dont on donne son code  
https://unicode-table.com  
"""
```

```
c = 'p'  
print(f"La valeur ASCII du caractère {c} est", ord(c))  
print(f"Le caractère ASCII dont le code est {ord(c)} est le caractère ", chr(ord(c)))  
"""
```

En utilisant la notation u"\u2600" avec une valeur en hexadécimal on affiche le caractère grace à son code unicode

```
"""  
s=u"\u0070"  
print(s)  
print(type(s))  
print(u"\u2600 \u2601 \u2602 \u2603 \u2604 \u2605")  
print("☀ ☁ ☂ ☃ ☄ ★")  
s=u"\u2600"  
print(s)  
print(type(s))  
s=u"\u2600".encode("utf-8")  
print(s)  
print(type(s))
```

La valeur ASCII du caractère p est 112

Le caractère ASCII dont le code est 112 est le caractère p

```
p  
<class 'str'>  
☀ ☁ ☂ ☃ ☄ ★  
☀ ☁ ☂ ☃ ☄ ★  
☀  
<class 'str'>  
b'\xe2\x98\x80'  
<class 'bytes'>
```

Exercice 1 : Le code du symbole ☺ est U+263A en UNICODE. Exprimer ce symbole en UTF-8.

Conversion en binaire :

Utiliser le tableau qui convient à la conversion :

1	1	0	D10	D9	D8	D7	D6	1	0	D5	D4	D3	D2	D1	D0

1	1	1	0	D15	D14	D13	D12	1	0	D11	D10	D9	D8	D7	D6	1	0	D5	D4	D3	D2	D1	D0

Code UTF-8 :

Exercice 2 : Le code du symbole Ç est U+00C7 en UNICODE. Exprimer ce symbole en UTF-8.

Conversion en binaire : 0000 0000 1100 0111

Utiliser le tableau qui convient à la conversion :

1	1	0	D7	D6	D5	D4	1	0	D3	D2	D1	D0
1	1	0	1	1	0	0	1	0	0	1	1	1

1	1	0	D10	D9	D8	D7	D6	1	0	D5	D4	D3	D2	D1	D0

1	1	1	0	D15	D14	D13	D12	1	0	D11	D10	D9	D8	D7	D6	1	0	D5	D4	D3	D2	D1	D0

Code UTF-8 :

Exercice 3 : Le code du symbole ☸ est U+07F7 en UNICODE. Exprimer ce symbole en UTF-8.

Symbole *n'ko* gbakourounèn

Conversion en binaire :

Utiliser le tableau qui convient à la conversion :

1	1	0	D10	D9	D8	D7	D6	1	0	D5	D4	D3	D2	D1	D0

1	1	1	0	D15	D14	D13	D12	1	0	D11	D10	D9	D8	D7	D6	1	0	D5	D4	D3	D2	D1	D0

UTF-8 :

