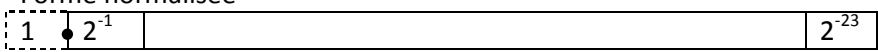




A- Représentation approximative des nombres réels :

Quel que soit le nombre n de bits utilisés pour représenter les nombres à virgule flottante, on dispose toujours d'un nombre fini de combinaisons (2^n) qui ne pourra jamais représenter tous les réels (une infinité).

En simple précision (sur 32 bits selon norme IEEE754), les nombres sont codés de la manière suivante :

1bit	8bit	23bit
Signe	Exposant	Mantisse ou partie fractionnaire.
0/1 +/-	Valeur prise en compte : $2^{(\text{Exposant} - 127)}$ Soit de 2^{-127} à 2^{+128} Donc de $\approx 10^{-38}$ à 10^{+38}	Forme normalisée  Erreur relative : $\epsilon = 2^{-23} \approx 1,2 \cdot 10^{-7}$; soit mieux que 10^{-6} Soit 6 chiffres significatif garanti en décimal
Au final la valeur se calcule : $(-1)^S \times (1 + M) \times 2^{E-127}$		

Source : <https://www.youtube.com/watch?v=mtizhxB-Zw>

Exemple :

1	Sign 0	Exponent 01111111	Mantissa 00000000000000000000000
0,5	Sign 0	Exponent 01111110	Mantissa 00000000000000000000000
0,25	Sign 0	Exponent 01111101	Mantissa 00000000000000000000000
2	Sign 0	Exponent 10000000	Mantissa 00000000000000000000000
8	Sign 0	Exponent 10000010	Mantissa 00000000000000000000000
$1,25 \times 2^{-3}$ = 0,15625	Sign 0	Exponent 01111100	Mantissa 01000000000000000000000

Erreur connue liée à ce codage :

```
>>> 0.1+0.2
0.30000000000000004
>>> 0.3 == (0.1+0.2)
False
```

Si vous devez vérifier l'égalité de deux réels, vérifiez plutôt que la distance entre les deux est faible :

```
>>> abs(0.3-(0.2+0.1)) < 0.00000001
True
```

Comparaison des types float en simple ou double précision

Type de float	Encodage	Signe	Exposant	Mantisse	Valeur min	Valeur max	Chiffres significatifs garantis
Simple précision	32bits	1bit	8bits	23bits	1.2 E-38	3.4 E38	6
Double précision	64bits	1bit	11bits	52bits	2.2 E-308	1.8 E308	15

B- Codage d'un caractère

Les systèmes numériques ne pouvant mémoriser/transmettre que des nombres binaires, il faut définir un codage (correspondance nombre ↔ caractère) reconnu par le plus grand nombre.



Le Codage ASCII

En 1963, IBM propose à l'organisme de normalisation (ANSI) le code ASCII (*American Standard Code for information Interchange*) qui serait commun à tous les ordinateurs.

Il est codé sur 7 bits, représente 128 caractères, dont les 32 premiers ne sont pas éditables (caractères de contrôle).

Dec	Hex	Name	Char	Ctrl-char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	Null	NUL	CTRL-@	32	20	Space	64	40	@	96	60	`
1	1	Start of heading	SOH	CTRL-A	33	21	!	65	41	A	97	61	a
2	2	Start of text	STX	CTRL-B	34	22	"	66	42	B	98	62	b
3	3	End of text	ETX	CTRL-C	35	23	#	67	43	C	99	63	c
4	4	End of xmit	EOT	CTRL-D	36	24	\$	68	44	D	100	64	d
5	5	Enquiry	ENQ	CTRL-E	37	25	%	69	45	E	101	65	e
6	6	Acknowledge	ACK	CTRL-F	38	26	&	70	46	F	102	66	f
7	7	Bell	BEL	CTRL-G	39	27	'	71	47	G	103	67	g
8	8	Backspace	BS	CTRL-H	40	28	(72	48	H	104	68	h
9	9	Horizontal tab	HT	CTRL-I	41	29)	73	49	I	105	69	i
10	0A	Line feed	LF	CTRL-J	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	VT	CTRL-K	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	FF	CTRL-L	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage feed	CR	CTRL-M	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	SO	CTRL-N	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	SI	CTRL-O	47	2F	/	79	4F	O	111	6F	o
16	10	Data line escape	DLE	CTRL-P	48	30	0	80	50	P	112	70	p
17	11	Device control 1	DC1	CTRL-Q	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	DC2	CTRL-R	50	32	2	82	52	R	114	72	r
19	13	Device control 3	DC3	CTRL-S	51	33	3	83	53	S	115	73	s
20	14	Device control 4	DC4	CTRL-T	52	34	4	84	54	T	116	74	t
21	15	Neg acknowledge	NAK	CTRL-U	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	SYN	CTRL-V	54	36	6	86	56	V	118	76	v
23	17	End of xmit block	ETB	CTRL-W	55	37	7	87	57	W	119	77	w
24	18	Cancel	CAN	CTRL-X	56	38	8	88	58	X	120	78	x
25	19	End of medium	EM	CTRL-Y	57	39	9	89	59	Y	121	79	y
26	1A	Substitute	SUB	CTRL-Z	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	ESC	CTRL-[59	3B	;	91	5B	[123	7B	{
28	1C	File separator	FS	CTRL-\	60	3C	<	92	5C	\	124	7C	
29	1D	Group separator	GS	CTRL-]	61	3D	=	93	5D]	125	7D	}
30	1E	Record separator	RS	CTRL-^	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	US	CTRL-`	63	3F	?	95	5F	`	127	7F	DEL

Standard Unicode UTF-8

En 1988, des industriels forment le consortium Unicode.

Le but est de répertorier tous les caractères (en 2016 : 137 374 caractères) de toutes les langues (une centaine d'écritures) et de leur associer un code.

- Unicode désigne le catalogue de code points.
- UTF (Universal Character Set Transformation Format) désigne l'encodage.

Le code UTF-8 est de longueur variable (de 1 à 4 octets). Les caractères les plus utilisés sont codés sur 1 octet et respecte le code ASCII. Actuellement le plus utilisé.

Le code UTF-16 utilise 2 ou 4 octets. Il est compatible avec la version 1.0 d'Unicode (UCS).

Le code UTF-32 est de longueur fixe de 4 octets. Il est peu utilisé.

A	Ω	語	𐄎	UTF-32
00000041	000003A9	00008A9E	00010384	
A	Ω	語	𐄎	UTF-16
0041	03A9	8A9E	D800 DF84	
A	Ω	語	𐄎	UTF-8
41	CE A9	E8 AA 9E	F0 90 8E 84	

Source : jolicode

C- Codage d'une image :

En numérique une image matricielle est constituée de points élémentaires appelés pixels (*picture element*).

La définition (pas la résolution) de l'image donne le nombre de pixels (par exemple 480x640 ou 5Mpixels).

La couleur de chaque pixel est codée à partir des 3 couleurs Rouge Vert Bleu. Le mode 16 millions de couleurs code chacune des 3 couleurs à l'aide d'un octet, soit 256 nuances pour chaque couleur primaire.

Voir https://www.w3schools.com/colors/colors_hexadecimal.asp

Un octet de transparence (ou opacité) peut être rajouté. On parle de canal alpha.

Un fichier image au format BMP enregistre les 3 octets de chaque pixel dans l'ordre, sans compression. Il est alors facile de calculer sa taille.

Quelques formats d'images

Format	Bmp (bitmap)	Jpeg/jpg	png	gif	svg
Poids	Lourd	Bon à très bon	Bon	Bon	Bon
Nombre de couleurs	De 2 (noir&blanc) à 16 millions	16 millions	De 2 à 16 millions ou plus	256 par calques	16 millions
Compression	Non	Oui, avec perte	Oui, sans perte	Oui, sans perte	Oui, possible
Transparence	Non	Non	Oui	Oui	Oui
Domaines appropriés	Format historique : presque plus utilisé	Photo couleur	Tous sauf photo et images animées	Images animées	Images vectoriels