

1. Bases de numérations

Un système de numération est un ensemble de symboles (les chiffres) qui sont assemblés en suivant des règles d'écriture précises permettant d'écrire, de lire et d'énoncer les nombres.

Dans l'Histoire, des dizaines de systèmes de numération ont été inventés pour répondre à différents besoins. Il existe des systèmes de numération :

- additifs tels que les chiffres romains. Exemples : XXXII=32, IX=9
- positionnels comme la numération décimale (base 10) qu'on utilise dans la vie de tous les jours. La place du chiffre dans le nombre définit sa valeur. Exemple « 342 » c'est 2 unités, 4 dizaines et 3 centaines.

2. Les bases 2, 10, 16

En fonction des besoins, en sciences de l'ingénieur, on utilise 3 bases :

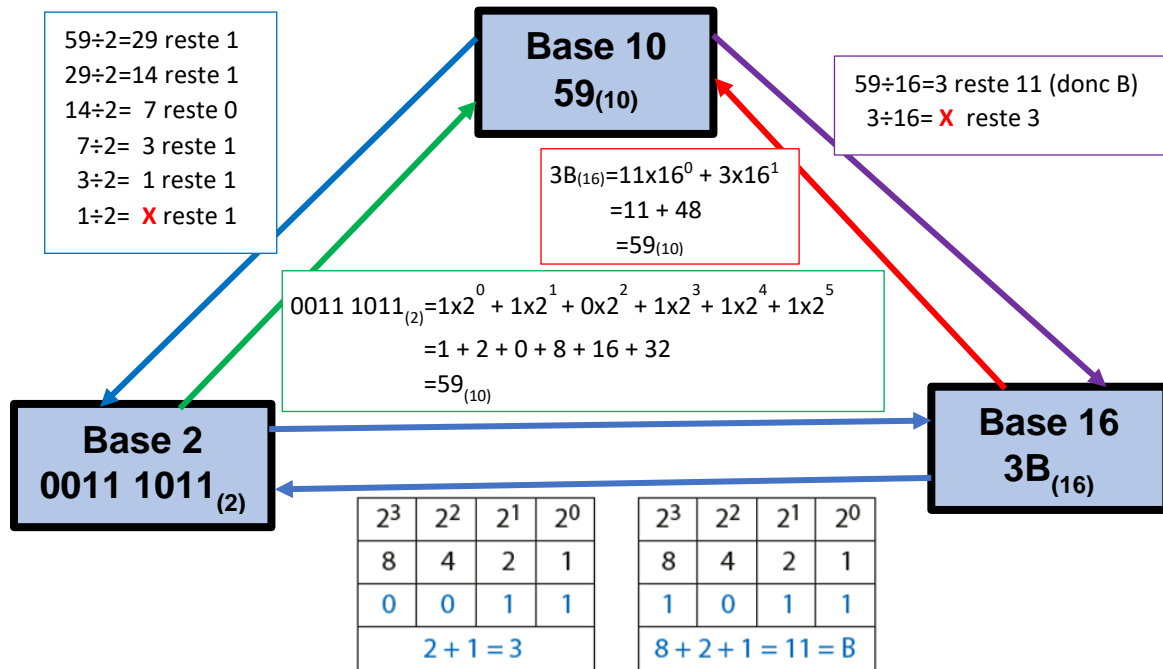
- la base 10 que nous utilisons dans la vie de tous les jours ;
- la base 2 utilisée par les ordinateurs ;
- la base 16 qui permet de compacter les mots écrits en base 2 pour les rendre plus lisibles par les humains.

Le tableau ci-dessous permet de lire directement comment sont écrits les 16 premiers nombres dans les 3 bases :

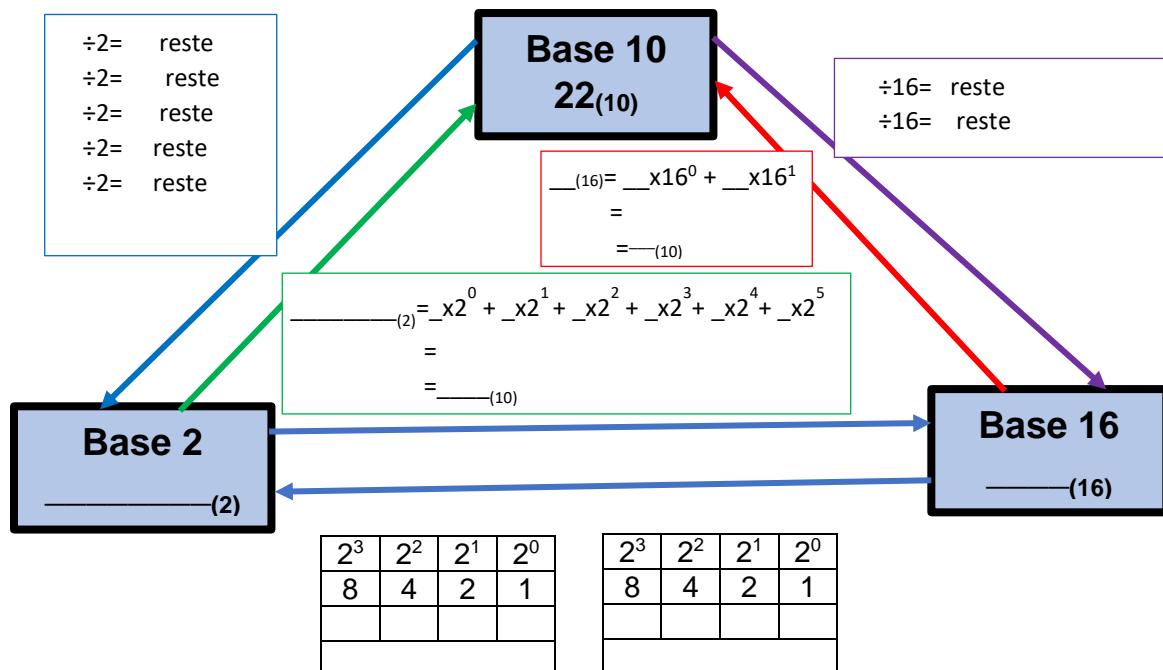
Décimal (base 10)	Binaire naturel (base 2)	Hexadécimal (base 16)
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Une **base**, dans un système de numération positionnel, est le nombre de symboles (de chiffres) qui sont utilisés pour représenter les nombres.

Voici, sur un exemple, les méthodes pour passer d'une base à l'autre.



Compléter ce 2^{ème} exemple :



Notation

Lorsqu'on écrit un nombre, il faut spécifier dans quelle base il est écrit, sinon il y a des risques de confusion. Par convention, on indique la base en indice :

- $3A9_{(16)} = (3A9)_{16}$ → base 16 : hexadécimal
- $238_{(10)} = (238)_{10}$ → base 10 : décimal
- $0100\ 1101_{(2)} = (0100\ 1101)_2$ → base 2 : binaire

En programmation, la notation permettant de voir que l'on est en hexadécimal dépend du langage :

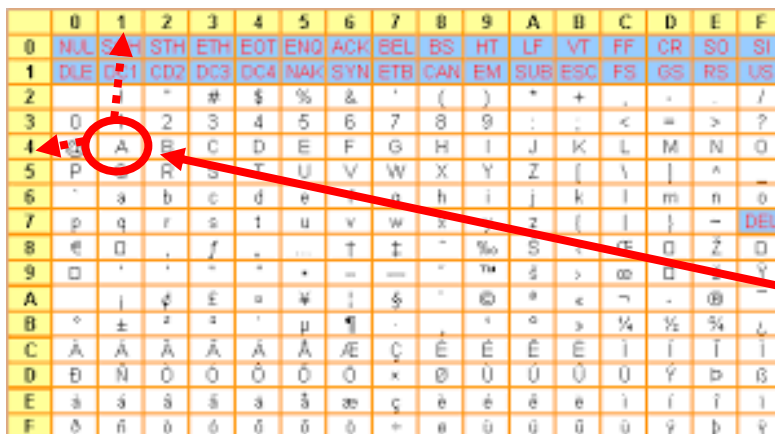
Exemple avec $(AE4F)_{16}$:

Langage	Préfix	Exemple
C, C++, Java	0x	0xAE4F
Pascal	\$	\$AE4F
Basic	&h	&hAE4F
HTML	#	#AE4F

3. Codages

A. Code ASCII

Le code ASCII (American Standard Code for Information Interchange) est principalement utilisé pour la transmission en série des caractères alphanumériques (Chiffres, lettres, caractères spéciaux) dans les systèmes informatiques.



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETH	EOT	ENO	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	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	€	0	.	/	-	...	†	‡	‰	§	¶	•	◊	◻	◼	◽
9	□	◊	◻	◼	◽	◾	◿	⋄	⋆	⋈	⋊	⋋	⋌	⋍	⋎	⋏
A		¡	¢	£	¤	¥	¦	§	¨	©	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

Chaque caractère en code ASCII est codé sur 8 bits, donc sur 2 caractères exprimés en hexadécimal.

Exemple :

Le caractère « A » est codé en hexadécimal par 41

Coder les initiales « si » (en minuscule) en base 16, puis en base 2

B. Unicode

Pour coder de manière universelle l'ensemble des symboles utilisés, quelle que soit la langue (Français, Anglais, Grec, Chinois, ...), il faut attribuer à tout caractère ou symbole un nom et un identifiant numérique, et ce de manière unifiée. C'est ce que propose la norme Unicode.

Pour stocker sur un support informatique un texte constitué de caractères Unicode, il faut utiliser un procédé appelé processus d'encodage.

Actuellement un des systèmes d'encodage couramment utilisé est UTF-8.



character	encoding	bits
A	UTF-8	01000001
A	UTF-16	00000000 01000001
A	UTF-32	00000000 00000000 00000000 01000001