

REPRÉSENTATION DES ENTIERS

1. INTRODUCTION

Dans un ordinateur, toutes les informations (textes, images, sons, nombres, programmes...) sont représentées avec seulement deux chiffres : 0 et 1.

Chaque 0 ou 1 est appelé **un bit**.

Les bits sont généralement regroupés par 8 pour former **un octet**.

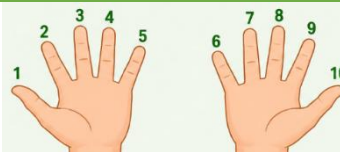
Les informations sont généralement stockées sous forme d'octets.

Pour représenter les différentes informations, l'ordinateur utilise des **encodages**.

Un **encodage** est une règle qui associe une suite de 0 et de 1 à une information (une lettre, un nombre, une couleur, etc.).

2. ENCODAGE DES ENTIERS NATURELS

1. Écrire un nombre en base 10



Un nombre entier en **base 10** est une **séquence de chiffres compris entre 0 et 9**.

Plus concrètement prenons l'exemple de la séquence 57 302 que l'on représente dans le tableau ci-dessous en indiquant les positions des chiffres en commençant à compter les indices à partir de 0 de la droite vers la gauche ainsi que les poids correspondants.

Séquence	5	7	3	0	2
Position	4	3	2	1	0
Poids	10^4	10^3	10^2	10^1	10^0

$$57\,302 = 5 \times 10^4 + 7 \times 10^3 + 3 \times 10^2 + 0 \times 10^1 + 2 \times 10^0$$

👉 Le chiffre le plus à gauche est appelé **poids fort** : 5

👉 Le chiffre le plus à droite est appelé **poids faible** : 2

À toi de jouer : Complète

Séquence					
Position	4	3	2	1	0
Poids	10^4	10^3	10^2	10^1	10^0

$$63\ 027 =$$

2. Écrire un nombre en base 2



De manière similaire à l'encodage en base 10, une séquence de chiffres binaires peut s'interpréter comme nombre écrit en base 2.

Par exemple, l'octet de bits 01101101 peut être représenté dans le tableau ci-dessous :

Séquence	0	1	1	0	1	1	0	1
Position	7	6	5	4	3	2	1	0
Poids	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

$$01101101_2 = (0 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0)_{10}$$

$$01101101_2 = (64 + 32 + 8 + 4 + 1)_{10}$$

$$01101101_2 = 109_{10}$$

À toi de jouer : Complète

Séquence								
Position	7	6	5	4	3	2	1	0
Poids	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

$$10011010_2 =$$

$$10011010_2 =$$

$$10011010_2 =$$

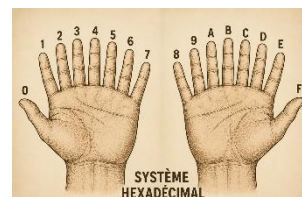
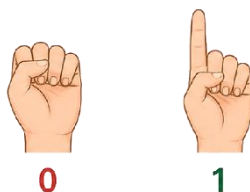
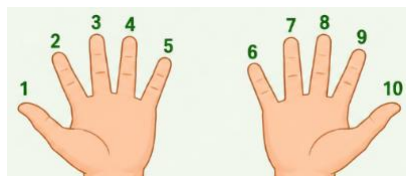
3. Écrire un nombre en base 16

Une autre base fréquemment utilisée est **la base 16**, dite **base hexadécimale**.

Puisqu'il faut pouvoir écrire 16 chiffres hexadécimaux on utilise les chiffres de 0 à 9 pour les 10 premiers puis les lettres A, B, C, D, E et F pour les 6 derniers.

La valeur de chaque lettre est alors simplement donnée dans le tableau de correspondance ci-dessous :

À toi de jouer : Complète



Décimal	Binaire	Hexadécimal
0		0
1		1
2		2
3		3
4		4
5		5
6		6
7		7
8		8
9		9
10		A
11		B
12		C
13		D
14		E
15		F

De manière similaire aux bases 2 et 10, on peut représenter les séquences de chiffres hexadécimaux en colonnes, en indiquant position et poids des chiffres

Séquence	A	4	D
Position	2	1	0
Poids	16^2	16^1	16^0

$$A4D_{16} = (10 \times 16^2 + 4 \times 16^1 + 13 \times 16^0)_{10}$$

$$A4D_{16} = (2\,560 + 64 + 13)_{10}$$

$$A4D_{16} = 2\,637_{10}$$

À toi de jouer : Complète

Séquence			
Position	2	1	0
Poids	16^2	16^1	16^0

$$2BD_{16} =$$

$$2BD_{16} =$$

$$2BD_{16} =$$

La base 16 est souvent utilisée pour simplifier l'écriture de nombres binaires. En effet, on peut facilement passer d'un nombre en base 2 à un nombre en base 16 en regroupant les chiffres binaires par 4.

À toi de jouer : Complète

Par exemple, la séquence de bits 1010010111110011 correspond au nombre hexadécimal

4. Unités de mesure

Rappels

Préfixe	Tera	giga	méga	kilo	unité	milli	micro	nano
Symbole	T	G	M	k		m	μ	n
10^n	10^{12}	10^9	10^6	10^3	$10^0 = 1$	10^{-3}	10^{-6}	10^{-9}

Il est très courant en informatique de mesurer la capacité mémoire d'un disque dur de la RAM d'un ordinateur ou d'un débit de données internet avec une unité de mesure exprimée comme un multiple d'octets.

Ses multiples sont traditionnellement des puissances de 10 et on utilise aussi les préfixes kilos méga etc ...

Préfixe	giga	méga	kilo	unité
Symbole	Go	Mo	ko	octet
10^n	10^9	10^6	10^3	$10^0 = 1$

3. Lien avec Python



0b est un préfixe qui indique que le nombre est en binaire, Python donnera sa traduction en décimal.

```
>>> 0b01111
15
```

Inversement, on peut convertir une valeur n vers la base 2 à l'aide de la fonction bin(n).

Cette fonction renvoie une chaîne de caractères.

```
>>> bin(42)
'0b101010'
```

De la même manière, on peut manipuler des nombres hexadécimaux à l'aide de la notation 0x suivi de n'importe quelle séquence de chiffres en base 16,

Python donnera sa conversion en décimal.

Inversement, on peut convertir une valeur n vers la base 16 à l'aide de la fonction `hex(n)`.

Cette fonction renvoie une chaîne de caractères.

À toi de jouer : Donne deux exemples sur Python