

Types et valeurs de base – Fiche de cours

1. Les systèmes de numération

a. Présentation

Pour organiser les données, les systèmes informatiques utilisent des représentations numériques

b. Numération décimale

La numération décimale utilise 10 chiffres (symboles) :
0, 1, 2, 3, 4, 5, 6, 7, 8, 9

c. Numération binaire

La numération binaire utilise 2 chiffres (symboles) :
0, 1

Un octet est composé de 8 bits

d. Numération hexadécimale

La numération hexadécimale utilise 16 chiffres (symboles) :
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

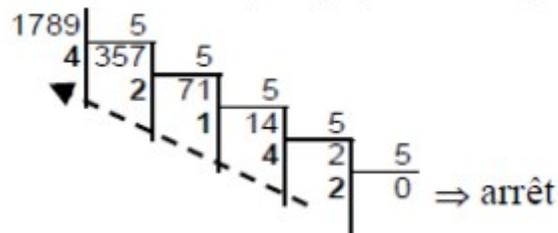
2. Changement de base

a. Conversion de la base b à la base 10

$$(a_k a_{k-1} \dots a_1 a_0)_b = (a_k \cdot b^k + a_{k-1} \cdot b^{k-1} + \dots + a_1 \cdot b + a_0)_{10}$$

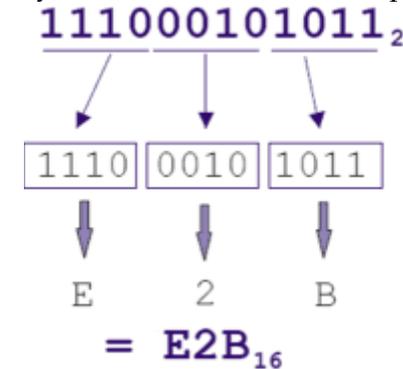
b. Conversion de la base 10 à la base b

On utilise la méthode de la division successive par b



c. Conversion de la base 2 à la base 16

On découpe le nombre binaire par paquet de 4 à partir du bit de poids faible et on associe le symbole hexadécimal correspondant



3. Représentation des booléens

a. Type de données booléennes

Les nombres booléens prennent 2 valeurs : True / False et sont stockés sur 1 octet (8 bits)

b. Opérations associées

Les opérations de logique usuelles peuvent être utilisés avec des booléens

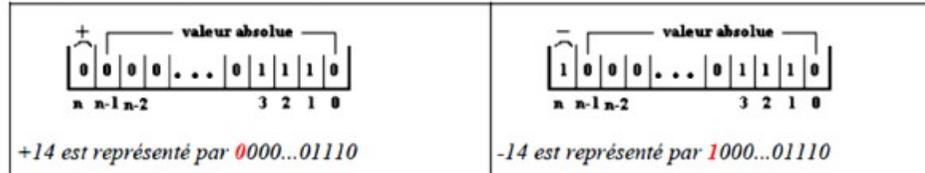
- fonction NON : **not** condition
- fonction ET : condition1 **and** condition2
- fonction OU : condition1 **or** condition2
- fonction OU EXCLUSIF : condition1 **xor** condition2

4. Représentation des entiers relatifs (entiers signés)

a. Binaire signé

- premières machines

Un mot binaire à n bit est composé de 1 bit de signe et de n-1 bits représentant la valeur absolue du nombre



- complément à 2

b. Type de données « entier »

Un nombre de type int est codé sur 4 octets (32 bits)

Un nombre de type long est codé sur 8 octets (64 bits)

Python s'adapte au type de données et sélectionne automatiquement le type à utiliser

c. Opérations associées

Il est possible de réaliser toutes les opérations arithmétique usuelles avec les nombres de types entiers

5. Représentation des nombres réels

a. Conversion d'un réel en binaire

Les parties entière et décimale en base 10 sont codées en binaire

exemple : $(5,1875)_{10} = (0101,0011)_2$

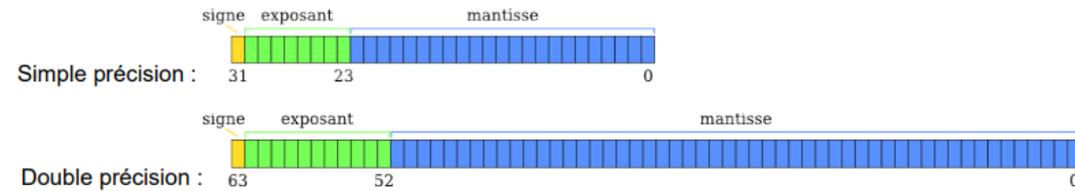
b. Conversion binaire en réel

Les parties entière et décimale binaire sont codées en base 10

exemple : $(100,0101)_2 = (4,3125)_{10}$

c. Représentation des flottants

La norme IEEE 754 est la plus employée pour la représentation des nombres à virgule flottante en informatique.



d. Type de données en virgule flottante

Un nombre de type float est codé sur 4 octets (32 bits)

Un nombre de type double est codé sur 8 octets (64 bits)

Python s'adapte au type de données et sélectionne automatiquement le type à utiliser

e. Opérations associées

Il est possible de réaliser toutes les opérations arithmétique usuelles avec les nombres de types entiers

6. Représentation des caractères

a. Code ASCII

Le code ASCII sert à coder les caractères usuels sur 1 octet (8bits)

b. Opérations associées

en langage Python `chr()` retourne le caractère d'un code ascii
`ord()` retourne le code ascii d'un caractère

c. Type de données string

Les variable de type string sont des tableaux de caractères associés à des codes ascii