

---

# Nombres flottants

---

## 1 Présentation

Dans les langages de programmation comme Python, les nombres à virgule sont appelés des **flottants** ou **float** et se notent avec un point. Par exemple, le nombre 2,5 se note 2.5.

Comme les nombres entiers, ils s'écrivent en binaire:

- La partie entière est composée de puissances de 2 positives;
- La partie après la virgule (ou le point), est composée de puissances de 2 négatives.

### Exemple

$$24,53_{10} = 2 \times 10^1 + 4 \times 10^0 + 5 \times 10^{-1} + 3 \times 10^{-2}$$
$$10,01_2 = 1 \times 2^1 + 0 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$$

## 2 Conversion

Pour convertir un nombre binaire en nombre décimal, on calcule la somme des puissances de 2 positives et négatives. On rappelle quelques valeurs de puissances de 2.

$2^1$	$2^0$	$2^{-1}$	$2^{-2}$	$2^{-3}$	$2^{-4}$	$2^{-5}$
2	1	0,5	0,25	0,125	0,0625	0,03125

### Exemple

Conversion du nombre binaire  $1,01_2$  en base 10.

$$1,01_2 = 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} = 1 + 0 + 0,25 = 1,25_{10}$$

### Méthode

La conversion d'un nombre en écriture décimale en écriture binaire (cas le plus fréquent puisque nous saisissons des nombres en base 10), nous procédons en deux temps.

1. Conversion de la partie entière avec la méthode des soustractions ou divisions successives;
2. Conversion de la partie après la virgule par multiplications par 2 successives.  
Après chaque multiplication par 2, on retire la partie entière obtenue et on recommence avec la partie décimale jusqu'à obtenir la valeur 1.

### Exemple

Conversion du nombre 5,375 en binaire:

1. La partie entière :  $5_{10} = 101_2$

2. La partie après la virgule :

- $0,375 \times 2 = 0,750$  donc le premier chiffre après la virgule est 0;
- $0,750 \times 2 = 1,5$  donc le second chiffre après la virgule est 1,
- $0,5 \times 2 = 1$  donc le dernier chiffre après la virgule est un 1

La conversion est achevée et le nombre  $5,375_{10}$  en base 10 s'écrit  $101,011_2$  en binaire.

### Remarque

La méthode de conversion ci-dessus revient à transformer une écriture décimale en écriture fractionnaire dont le numérateur est un nombre entier et le dénominateur une puissance de 2.

$$5,375 = \frac{5,375 \times 2^3}{2^3} = \frac{43}{8}$$

Comme le nombre  $43_{10} = 32 + 8 + 2 + 1 = 2^5 + 2^3 + 2^1 + 2^0$  soit  $101011_2$  et avec un décalage de trois rangs, on obtient le nombre  $101,011_2$ .

## 3 Notation scientifique

Dans le système décimal, les grands et petits nombres sont écrits en notation scientifique.

Par exemple, la valeur 7 milliards se note  $7 \times 10^9$ . On rappelle que le nombre écrit devant la puissance de 10 est un nombre compris de 1 à 9.

En binaire, les nombres peuvent s'écrire avec une puissance de 2. La virgule dans le nombre se positionne selon la puissance ou l'exposant utilisé. Voilà pourquoi on parle de **flottant** en faisant référence à cette virgule qui **flotte**.

### Exemple

- Avec un exposant  $n = 3$  :  $1101,0101_2 = 1,1010101 \times 2^3$
- Avec un exposant  $n = 2$  :  $1101,0101_2 = 11,010101 \times 2^2$
- Avec un exposant  $n = -4$  :  $1101,0101_2 = 11010101 \times 2^{-4}$

### Remarque

Sur les machines, des normes d'écriture des nombre flottants existent selon l'encodage en 32 ou 64 bits et reposent sur l'usage des puissances de 2.

- le premier bit est réservé au signe du nombre;
- les bits suivants (8 ou 11) sont réservés à l'exposant;
- les derniers bits (23 ou 52) appelé **mantisse** sont réservés à la valeur du nombre.

### Exercice

1. Donner l'écriture binaire du nombre  $5,6875_{10}$ .
2. Donner l'écriture binaire du nombre  $0,2_{10}$ . Que remarque-t-on ?

## 4 Conclusion

Les nombres à virgule ont une écriture binaire. Dans certains cas, la partie après la virgule est infinie. Comme la machine alloue un nombre de bits fini, la valeur est donc approchée et peut alors générer des erreurs dans les calculs.

## Exemple

La valeur obtenue par le calcul  $0,1 + 0,2$  en binaire est une valeur approchée de la valeur exacte  $0,3$ .

Les nombres  $0,1$  et  $0,2$  ont une partie après la virgule infinie donc approchée par la machine. Le nombre  $0,3$  a lui aussi une valeur approchée par la machine.

Ci-dessous, des calculs effectués par Python qui montre ces approximations.

```
[1]: 0.1+0.2
```

```
[1]: 0.30000000000000004
```

```
[2]: 0.1*3
```

```
[2]: 0.30000000000000004
```

```
[3]: 0.3*3
```

```
[3]: 0.8999999999999999
```