

Bases de numération

Fiche de synthèse

① ÉCRIRE UN NOMBRE DANS UNE BASE

Définition 1 (Base b) En base b , on dispose de b chiffres, de 0 à $b - 1$. En partant de la droite, le chiffre de rang k a pour poids b^k : il compte des paquets de b^k .

En base 10, $374 = 3 \times 10^2 + 7 \times 10^1 + 4 \times 10^0$. On note la base en indice : 101_2 vaut $1 \times 4 + 0 \times 2 + 1 \times 1 = 5$; sans indice, un nombre est écrit en base 10.

- **Base 2**, ou binaire : un circuit distingue deux états, notés 0 et 1. Un chiffre binaire est un **bit**; huit bits forment un **octet**.
- **Base 16**, ou hexadécimal : seize chiffres, de 0 à 9 puis A, B, C, D, E et F pour 10 à 15. Elle écrit les suites de bits quatre fois plus court.

Base 10	8	9	10	11	12	13	14	15
Base 2	1000	1001	1010	1011	1100	1101	1110	1111
Base 16	8	9	A	B	C	D	E	F

② DU BINAIRE AU DÉCIMAL

On additionne les poids des bits qui valent 1.

Bit	1	0	1	1	0	1	1	0
Poids	128	64	32	16	8	4	2	1

$10110110_2 = 128 + 32 + 16 + 4 + 2 = 182$. Un octet va de $00000000_2 = 0$ à $11111111_2 = 255$: il prend 256 valeurs.

③ DU DÉCIMAL AU BINAIRE

☞ **Divisions successives.** On divise le nombre par 2 : le reste donne le bit des unités. On recommence avec le quotient, jusqu'à obtenir un quotient nul. On lit les restes du dernier au premier.

Division	Quotient	Reste
$89 = 2 \times 44 + 1$	44	1
$44 = 2 \times 22 + 0$	22	0
$22 = 2 \times 11 + 0$	11	0
$11 = 2 \times 5 + 1$	5	1
$5 = 2 \times 2 + 1$	2	1
$2 = 2 \times 1 + 0$	1	0
$1 = 2 \times 0 + 1$	0	1

Lus de bas en haut, les restes donnent $89 = 1011001_2$. Vérification : $64 + 16 + 8 + 1 = 89$. On ne s'arrête pas quand le quotient vaut 1 : la dernière division donne encore le bit de gauche. Autre méthode : retirer la plus grande puissance de 2 possible, et recommencer, $89 = 64 + 16 + 8 + 1$.

④ BINAIRE ET HEXADÉCIMAL

Comme $16 = 2^4$, un chiffre hexadécimal correspond à quatre bits.

- Du binaire vers l'hexadécimal : on découpe en paquets de quatre bits **à partir de la droite**, puis on remplace chaque paquet. 10110110_2 donne 1011 et 0110, soit $B6_{16}$.
- De l'hexadécimal vers le binaire : on remplace chaque chiffre par ses quatre bits. $2C9_{16} = 0010\ 1100\ 1001_2$.
- Du décimal vers l'hexadécimal : divisions successives par 16. $2026 = 16 \times 126 + 10$, $126 = 16 \times 7 + 14$, $7 = 16 \times 0 + 7$: $2026 = 7EA_{16}$.

L'hexadécimal écrit les couleurs du Web, **#2E86C1** pour rouge 46, vert 134 et bleu 193, les adresses mémoire et les adresses MAC des cartes réseau.

⑤ EN PYTHON

```
print(bin(89))           # 0b1011001
print(int('1011001', 2)) # 89
print(hex(89))           # 0x59
print(int('59', 16))    # 89
```

⑥ ADDITION EN BINAIRE

On additionne colonne par colonne en partant de la droite : $1 + 1 = 10_2$, on écrit 0 et on retient 1 ; $1 + 1 + 1 = 11_2$, on écrit 1 et on retient 1. Ainsi $0110_2 + 0111_2 = 1101_2$, soit $6 + 7 = 13$. Sur un nombre fixé de bits, une retenue qui sort à gauche est perdue : c'est un **dépassement de capacité**. Sur un octet, $11111111_2 + 1$ donne 00000000_2 .

☑ À retenir.

- Le chiffre de rang k pèse b^k en base b .
- Binaire vers décimal : additionner les poids des bits à 1.
- Décimal vers binaire : divisions successives par 2 jusqu'au quotient nul, restes lus de bas en haut.
- Un chiffre hexadécimal vaut quatre bits.