NSI · PREMIÈRE / COURS

Représenter les entiers : binaire et hexadécimal

Astrolab Cours · astrolabcours.fr

01 Le problème

Un ordinateur ne manipule que des 0 et des 1 : du courant ou pas de courant. Pourtant il affiche 2026, −17 ou la couleur `#FF8A3D`. Comment écrire n'importe quel entier avec seulement deux chiffres ?

01 L'idée : le compteur kilométrique à deux chiffres

Sur un compteur de voiture, quand la roue des unités passe de 9 à 0, celle des dizaines avance d'un cran. Imagine un compteur dont chaque roue n'a que 0 et 1 : elle repasse à 0 tout de suite et fait avancer sa voisine. Il compte 0, 1, 10, 11, 100, 101… C'est le binaire.

01 La base 2

En **base 2**, chaque chiffre (un **bit**) vaut une puissance de 2, de droite à gauche : $1, 2, 4, 8, 16, 32, 64, 128\ldots$

$$1101_2 = 1\times 8 + 1\times 4 + 0\times 2 + 1\times 1 = 13$$

Pour passer de la base 10 à la base 2, on divise par 2 et on garde les restes, lus de bas en haut :

13 = 2 × 6 + **1** ; 6 = 2 × 3 + **0** ; 3 = 2 × 1 + **1** ; 1 = 2 × 0 + **1**. Donc $13 = 1101_2$.

Un **octet** est un groupe de 8 bits. Sur n bits, on écrit 2^n valeurs différentes, de 0 à $2^n - 1$. Sur un octet : de 0 à 255.

01 La base 16

Les longues suites de bits sont illisibles. On les regroupe par paquets de 4 : chaque paquet donne un chiffre **hexadécimal**, de 0 à 9 puis A (10) à F (15).

$$1111\,1010_2 = \mathrm{FA}_{16} = 15\times 16 + 10 = 250$$

Les couleurs du web s'écrivent ainsi : `#FF8A3D`, c'est rouge FF (255), vert 8A (138), bleu 3D (61).

En Python : `bin(13)` donne `'0b1101'`, `hex(250)` donne `'0xfa'`, et `int("1101", 2)` donne 13.

Les entiers négatifs : le complément à deux

Sur n bits, on code les entiers de -2^{n-1} à $2^{n-1} - 1$. Pour écrire $-x$:

1. on écrit x en binaire sur n bits ;
2. on inverse tous les bits ;

© Astrolab Cours · astrolabcours.fr/nsi/premiere/representation-des-entiers/ · Tous droits réservés : usage personnel uniquement, republication et usage commercial interdits

3. on ajoute 1.

Sur 8 bits : $5 = 0000\,0101$, inversé $1111\,1010$, plus 1 : $-5 = 1111\,1011$. Le bit de gauche vaut 1 pour les négatifs. Et $5 + (-5)$ donne bien $0000\,0000$ (la retenue finale est perdue).

Les pièges

- Lire les restes de haut en bas : on obtient le nombre à l'envers.
- 8 bits codent 256 valeurs, mais le maximum est 255.
- En complément à deux, ne pas oublier le « + 1 » après l'inversion.

© Astrolab Cours · astrolabcours.fr/nsi/premiere/representation-des-entiers/ · Tous droits réservés : usage personnel uniquement, republication et usage commercial interdits