Comment représenter les nombres entiers dans la mémoire de l'ordinateur?

## 1 Cellules mémoires



FIGURE 1 – Le bit est la plus petite unité informatique.

# 2 Encodage des entiers naturels

### 2.1 Écriture en base 10

$$6103 = 6 \times 10^3 + 1 \times 10^2 + 0 \times 10^1 + 3 \times 10^0$$

#### 2.2 Écriture en base 2

$$5 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$
$$5_{10} = 101_2$$

#### 2.3 Conversion

$$\begin{array}{c|c|c}
41 & 2 & & \\
1 & 20 & 2 & & \\
0 & 10 & 2 & & \\
0 & 5 & 2 & & \\
1 & 2 & 2 & & \\
0 & 1 & 2 & & \\
1 & 0 & & & \\
\end{array}$$

Figure  $2 - 41_{10} = 101001_2$ 

# 2.4 Écriture en base 16

La base 16 est régulièrement utilisé pour représenter les nombres binaires plus facilement. Chaque chiffre hexadécimal est représenté par 4 bits.



1 octet est représenté par 2 chiffres hexadécimaux.



décimal	hexadécimal	bits
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	В	1011
12	С	1100
13	D	1101
14	E	1110
15	F	1111

# 2.5 Python et les entiers

- Par défaut les nombres entiers sont encodés en base 10 en Python.
- Pour utiliser des nombres binaires, il suffit d'ajouter le préfixe 0b.
- Le préfixe 0x permet de manipuler des nombres en base hexadécimale.
- La fonction bin() convertit en base 2 n'importe quelle valeur.

