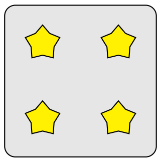
CHAPITRE 2 Des 0s et des 1s

1 Bases de numération

Il est important de distinguer en informatique, la **valeur** de sa **représentation**. Par exemple, lorsque l'on parle du nombre d'étoiles dans le carré ci-contre, on dit qu'il y a "quatre" étoiles, il s'agit d'une *valeur*. Mais on peut *représenter* cette information de différente manière : le symbole 4, ou bien le symbole *IV* (écriture romaine), ou bien 四 (écriture chinoise), etc.

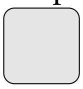


1.1 La base 10

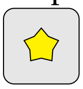
Pour représenter des valeurs numériques en base 10, on utilise deux règles :

Première règle. On utilise 10 symboles arbitraires. Actuellement une convention communément admise est d'utiliser les symboles 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 . On leur donne le sens suivant :

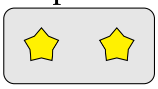
0 représente



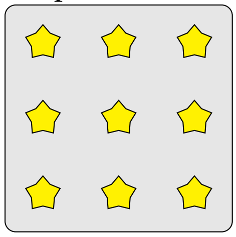
1 représente



2 représente



9 représente



Deuxième règle. Lorsque la valeur numérique est supérieure à 9, on utilise la position des symboles pour représenter le nombre de paquets de 10 créés.

Exemple 1. Si on dispose de 2317 objets identiques, et que l'on constitue des paquets de 10 objets, on en obtient 231 et il reste 7 objets. Ensuite, avec les 231 paquets on forme des ensembles de 10 paquets, on en obtient 23 ; puis on regroupe les 23 ensembles par 10, on obtient 2 groupes et il reste 3 ensembles. On a donc :

Résultat

```
[2][3][1][7]
|   |  |  |
|   |  |  7 unités
|   |  1 groupes de 10
|   3 groupes de 10 groupes de 10 (10 x 10 = 100)
2 groupe de 10 groupes de 10 groupes de 10 (10 x 10 x 10 = 1000)
```

La valeur est donc donnée par le calcul :

$$2 \times 10^3 + 3 \times 10^2 + 1 \times 10^1 + 7 \times 2^0 = \overline{2317}^{10}$$

Il y a deux mille trois cent dix sept étoiles dans un carré.

1.2 La base 2

On peut définir de la même manière la base 2.

Première règle. On utilise 2 symboles arbitraires. Par exemple 0 et 1.

Deuxième règle. Lorsque la valeur numérique est supérieure à 1, on utilise la position des symboles pour représenter le nombre de paquets de 2 créés.

Exemple 2. Si on dispose de 2317 objets et que l'on constitue des paquets de 2 on obtient 1158 paquets et il reste un objet. On groupe les paquets restants par 2 il en reste 579 et il en reste 0. On groupe les 579 paquets (de quatre) par 2 : on obtient 289 paquets de 8 et il en reste 1. (etc).

Résultat

```
[1] [0] [0] [1] [0] [0] [0] [0] [1] [1] [0] [1]
 |   |   |   |   |   |   |   |   |   |   |   |
 |   |   |   |   |   |   |   |   |   |   |   1 unité
 |   |   |   |   |   |   |   |   |   |   0 paquet de 2
 |   |   |   |   |   |   |   |   |   1 paquet de paquet de 2 (4)
 |   |   |   |   |   |   |   |   1   paquet de paquet de 4 (8)
 ...
 1 paquet de 1024
```

La valeur est donc donnée par le calcul
$1\times2^{10}+0\times2^9+0\times2^8+1\times2^7+\ldots+1\times2^3+1\times2^2+0\times2^1+1\times2^0=\overline{100100001101}^2$
Il y a deux mille trois cent dix sept étoiles dans un carré.

Propriété 1. *Pour obtenir l'écriture en base 10 d'un nombre écrit en base 2, il faut ajouter toutes les puissances de 2 qui correspondent au chiffre 1 dans l'écriture binaire du nombre.*

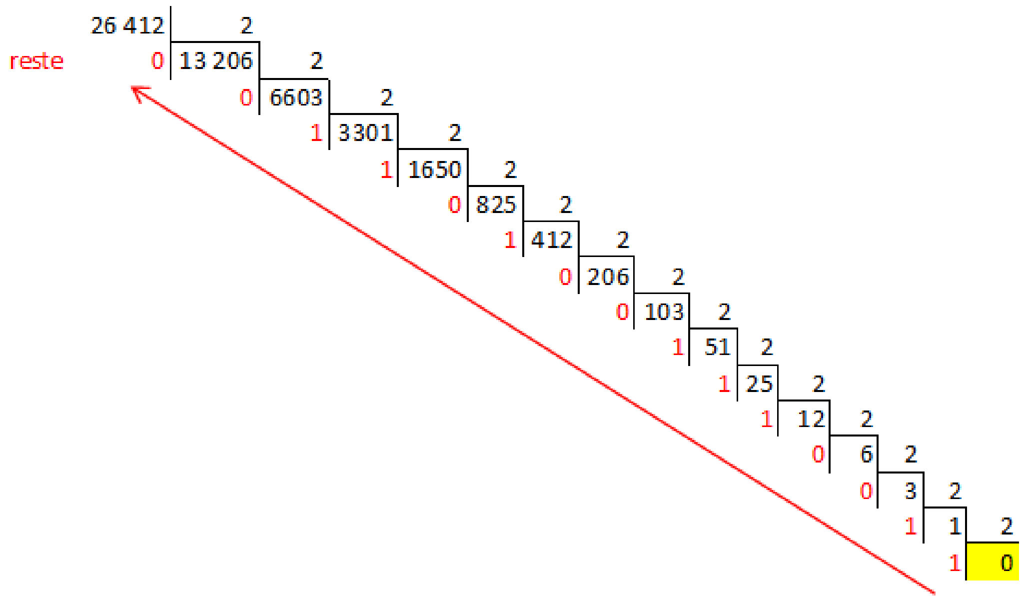
Propriété 2. *Pour obtenir l'écriture en base* 2 *d'un nombre* n *écrit en base 10 :*

- *On divise* n *par* 2 *et on note le quotient* q_0 *et le reste* b_0.
- *On divise le quotient* q_0 *par* 2 *et on note le quotient* q_1 *et le reste* b_1.
- *On continue ce procédé jusqu'à ce que le quotient obtenu soit égal à 0.*

L'écriture en base 2 *du nombre* n *est alors la suite des restes obtenus, écrits de gauche à droite, du dernier au premier :*

$$n=\overline{b_k\ldots b_1b_0}^2$$

Exemple 3. Par exemple, pour obtenir l'écriture en base 2 de $n=\overline{26412}^{10}$:



n	26412	13206	6603	...	3	1
q = n//2	13206	6603	3301	...	1	0
r = n%2	0	0	1	...	1	1

On a alors $n=\overline{110011100101100}^2$.

Propriété 3. *Les 16 premiers entiers écrits s'écrivent en base 2 sur quatre bits de la manière suivante.*

Base 10	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*
Base 2	*0000*	*0001*	*0010*	*0011*	*0100*	*0101*	*0110*	*0111*

Base 10	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*
Base 2	*1000*	*1001*	*1010*	*1011*	*1100*	*1101*	*1110*	*1111*

1.3 Base 16 : hexadecimal

La base 16 est couramment utilisée en informatique.

Première règle. On utilise 16 symboles arbitraires. Par exemple 0 pour représenter zéro unité, ..., 9 pour neuf unités, A (ou a) pour dix unités, B ou (b) pour onze unités, ..., F (ou f) pour quinze unités.

Deuxième règle. Lorsque la valeur numérique est supérieure à 16, on utilise la position des symboles pour représenter le nombre de paquets de 16 créés.

Exemple 4. $\overline{AF}^{16}$ représente le nombre $10 \times 16^1 + 15 \times 16^0 = \overline{175}^{10}$.
Attention, $\overline{123}^{16}$ représente le nombre $1 \times 16^2 + 2 \times 16^1 + 3 \times 16^0 = \overline{291}^{10}$.

Propriété 4. *Les 16 premiers entiers écrits s'écrivent en base 2 sur quatre bits et en base 16 de la manière suivante.*

Base 2	*0000*	*0001*	*0010*	*0011*	*0100*	*0101*	*0110*	*0111*
Base 16	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*

Base 2	*1000*	*1001*	*1010*	*1011*	*1100*	*1101*	*1110*	*1111*
Base 16	*8*	*9*	*a*	*b*	*c*	*d*	*e*	*f*

Propriété 5. *Pour écrire en base 2 un nombre écrit en base 16, on peut convertir chaque chiffre hexadécimal en un mot de quatre bit.*

Exemple 5. Pour convertir $n = \overline{4ea5}^{16}$ en base 2 on écrit :

4	e	a	5
0100	1110	1010	0101

Donc en base 2 n s'écrit : $\overline{100111010100101}^{2}$.

Propriété 6. *Pour écrire en base 16 un nombre écrit en base 2, on peut regrouper les bits quatre par quatre et donner l'écriture hexadécimale de chacun des paquets de quatre.*

Exemple 6. Pour convertir $n = \overline{1111101000101101}^{2}$ on écrit :

1111	1010	0010	1101
f	a	2	d

Donc en base 16 le nombre n s'écrit : $\overline{fa2d}^{16}$.

2 Logique et algèbre de Boole

Dans la logique « classique » une proposition est vraie ou fausse, il n'y a que deux possibilités. En python on utilise les valeurs `True` (Vrai) et `False` (Faux). On représente parfois la valeur `True` par le symbole 1 et la valeur `False` par le symbole 0.

Définition 1. On appelle algébre de Boole l'ensemble B constitué de deux éléments appelés **valeurs de vérité** $\{\text{Vrai}; \text{Faux}\}$ ou encore $\{0; 1\}$ (ou encore $\{\top, \bot\}$ avec $\top$ pour Vrai et $\bot$ pour Faux).
On définit sur cette ensemble trois opérations : ou, et, et non. On donne en donne les **tables de vérité** ci-dessous.

a	b	a ou b
0	0	0
0	1	1
1	0	1
1	1	1

a	b	a et b
0	0	0
0	1	0
1	0	0
1	1	1

a	non a
1	0
0	1

Remarque. Ainsi, si a et b sont deux valeurs booléenes dans B, a et b est un booléen qui n'est vrai que si a et b sont tous les deux vrais. a ou b est vrai si au moins l'une des deux valeurs a et b est vrai (elles peuvent l'être toutes les deux, on parle de ou "inclusif").

Propriété 7. *Pour tout a, b, c dans B :*

Commutativité

$$a \text{ ou } b = b \text{ ou } a \qquad a \text{ et } b = b \text{ et } a$$

Associativité

$$(a \text{ ou } b) \text{ ou } c = a \text{ ou } (b \text{ ou } c) \qquad (a \text{ et } b) \text{ et } c = a \text{ et } (b \text{ et } c)$$

Distributivité

$$a \text{ et } (b \text{ ou } c) = (a \text{ et } b) \text{ ou } (a \text{ et } c) \qquad a \text{ ou } (b \text{ et } c) = (a \text{ ou } b) \text{ et } (a \text{ ou } c)$$

Exemple 7. Démontrer que a et $(b$ ou $c) = (a$ et $b)$ ou $(a$ et $c)$ en construisant les tables de vérité des deux expressions.

a	b	c	b ou c	a et $(b$ ou $c)$
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

a	b	c	a et b	a et c	$(a$ et $b)$ ou $(a$ et $c)$
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			