

# COMPTAGE BINAIRE



# Comment comptons nous en décimal ?

- Pour bien comprendre comment on compte dans les autres bases, il est indispensable de revoir comment est fait notre système en base dix.
- En effet, tout le monde sait compter en base 10. C'est pratique dans la vie de tous les jours. Mais comment fonctionne notre mode comptage réellement ? Comment est construit notre système de nombres ?
- Pour répondre à cela, oublions tout et reprenons depuis le début : comment avez vous appris à compter à l'école ?

- Ça peut paraître simple comme question, mais notre système de comptage suit une logique simpliste. Sa compréhension est la clé qui vous ouvrira ensuite la porte pour apprendre à compter dans n'importe quelle autre base.
- Dans la pratique, nous comptons en base 10. Certains diront que cette pratique est venue du fait que nous avons 10 doigts. Il en découle principalement deux choses :
- Il existe **10 chiffres** : 0, 1, 2, 3, 4, 5, 6, 7, 8 et 9.
- Avec ces chiffres on peut compter jusqu'à 9. (La plus haute valeur des chiffres.)

- Pour aller au delà de 9 il faut **changer de rang**. Ça veut dire que si le rang des unités est plein, on commence le rang des dizaines et on remet les unités à zéro. Ensuite, on re-complète le rang des unités jusqu'à ce qu'il soit de nouveau plein. Puis on ajoutera une dizaine et les unités seront de nouveau remis à 0, et ainsi de suite.
- Par exemple, arrivé à 19, le rang des unités est plein. On ajoute donc une dizaine et on remet à zéro le rang des unités : on arrive donc à 20. Vous me suivez ?

- J'ai parlé de rangs des centaines, de dizaines et d'unités. On voit que une centaine vaut 10 dizaines et que une dizaines vaut 10 unités. Plus mathématiquement, **un rang est égale au précédent multiplié 10.**  
On peut dire que chaque rang est à une puissance de 10 supérieur au précédent.
- De cette manière, le **nombre  $56 = 50 + 6$**  mais que l'on peut aussi écrire  **$56 = 5 \times 10^1 + 6 \times 10^0$ .**
- **Ce que je viens de faire, c'est décomposer 56 en puissances de 10 (unités, dizaines, centaines...).**

- On peut décomposer chaque nombre en puissances de 10 successives. Par exemple,

$$3506 = 3 \times 10^3 + 5 \times 10^2 + 6 \times 10^0.$$

- Avec cette explication, vous devez avoir compris qu'en base 10 :
- On change de rang dès que la précédente est à 9.
- On peut décomposer tous les nombres en puissance de 10.
- Si on décompose un nombre en puissances de 10, c'est parce que 10 est notre base. Ceci est important, car en base 2, il faudra décomposer en puissances de... Deux !

# Le BINAIRE

- **Présentation**
- Le binaire est le mode de comptage non plus en base 10 mais **en base 2**. Il est utilisé par les ordinateurs, car les machines ne peuvent comparer que deux valeurs : des 1 et des 0.
- Je vous avais parlé des rangs (unités, dizaines, centaines...), et bien sachez qu'en binaire on emploie le mot « bit » (contraction de « binary-digit », signifiant simplement « rang binaire »). Par exemple, le nombre en base 2 « 10011 » s'étale sur 5 bit.

- Là où cela se complique, c'est qu'en binaire **chaque rang ne peut prendre que deux valeurs** (il pouvait en prendre dix en décimal). Donc, dès que le rang atteint sa deuxième – la plus haute – valeur on change de rang. **En binaire, un rang commence à 0 et se termine à 1.**
- Vous pouvez en comprendre que chaque bit représente une puissance de 2, tout comme chaque rang en base 10 est une puissance de 10.



Bon, pour commencer et tenter d'y voir un peu plus clair, on va compter en binaire jusqu'à dix :

valeur en décimal :	équivalent en binaire :	explications :
0	0	logique !
1	1	simple !
2	10	Le premier rang a atteint le maximum autorisé ! Qu'à cela ne tienne, on passe au rang suivant. On met le second à 1 et on remet le premier à 0.

3	11	On re-remplit le rang 1.
4	100	Le rang 1 est plein, mais le 2 aussi ! On passe donc au troisième et on remet les précédents à 0 (comme on le fait lorsque l'on passe de 0999 à 1000, par exemple).
5	101	On procède de même.
6	110	
7	111	
8	1000	On entame le quatrième rang.
9	1001	On recommence au premier...
10	1010	On remplit les rangs.

- Il suffit d'appliquer une règle : entamer le rang suivant quand celui en cours est plein!
- Bon, pour compter jusqu'à 10 ou même 20, cela va encore de remplir ce tableau, mais si je vous demande de convertir 450 en binaire ? Vous n'allez pas monter un par un, si ? Dans ce qui suit, on va voir une technique générale.

# Conversion du décimal en binaire

- Pour le moment, on n'a compté jusqu'à dix. Mais on ne sait pas encore **convertir**. Sans plus attendre donc, voici la conversion !
- **Méthode 1 : les puissances de 2**
- Pour y arriver, on doit décomposer notre nombre en puissances de 2. C'est le même principe que la décomposition en puissances de dix, sauf que l'on ne décompose pas en milliers, centaines et dizaines, mais en puissances de deux ; qui sont : 1, 2, 4, 8, 16, 32, 64 ..., 512, 1024, etc (une valeur est égale à la précédente multipliée par 2).

- Ainsi, si l'on prend l'exemple du nombre 26, on obtient la décomposition suivante :  $26 = 16 + 8 + 2$ . Il suffit ensuite de remplacer ces nombres par les puissances :
- $26 = 16 + 8 + 2$
- $26 = 1 \times 16 + 1 \times 8 + 1 \times 2$
- $26 = 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^1$  (on écrit les coef sous forme de puissances de 2)
- $26 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$  (on ajoute les puissances de 2 qui manquent)
- $26 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$  (voyez les puissances de 2 qui sont toutes là)
- $26 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$  (en orange : notre nombre en binaire !)

- **Il est important de ne pas oublier les puissances dont les coefficients sont zéro.**

Enfin, pour obtenir le nombre 26 en binaire, il suffit de mettre les coefficients qui sont devant les puissances de 2 à la suite. On obtient : 11010.

- **On écrit :  $(26)_{\text{dec}} = (11\ 010)_{\text{bin}}$**

- **Je récapitule la méthode :**
- On a notre nombre en décimal.
- On le décompose en valeurs de puissances de 2
- Si certaines puissances manquent, on les rajoutent en mettant 0 devant.
- On lit les coefficients devant les puissances de 2, ce sera notre nombre en binaire !
- Par commodité, d'écriture, on regroupe les chiffres par 4.  
(par ex : 101010101 se notera 1 0101 0101). On verra pourquoi plus loin.

- **Méthode 2 : les divisions euclidiennes par 2**
- Tout aussi simple à comprendre. Cette méthode est mieux pour des grands nombres et est plus facile à utiliser en programmation (il est facile d'en faire un *algorithme*). Voilà comment on fait :
- On a notre nombre en décimal.
- On le divise par 2 et on note le reste de la division (c'est soit un 1 soit un 0).
- On refait la même chose avec le quotient précédent, et on met de nouveau le reste de côté.
- On re-itére la division, et ce jusqu'à ce que le quotient est 0.
- Le nombre en binaire apparaît : le premier à placer est le dernier reste non nul. Ensuite, on remonte en plaçant les restes que l'on avait. On les place à droite du premier 1.



# Comme rien ne vaut un exemple

- Notre nombre est 164
- $164 \div 2 = 82 + 0$
- $82 \div 2 = 41 + 0$
- $41 \div 2 = 20 + 1$
- $20 \div 2 = 10 + 0$
- $10 \div 2 = 5 + 0$
- $5 \div 2 = 2 + 1$
- $2 \div 2 = 1 + 0$
- $1 \div 2 = 0 + 1$
- On voit apparaître notre nombre binaire en rouge : il faut le lire de bas en haut.

# Conversion du binaire en décimal

- Dans l'autre sens maintenant : convertir un nombre en base 2 en un nombre en base 10 ! je vous rassure tout de suite, c'est plus simple!
- Prenons le nombre (au hasard) : 101 0110. On voit qu'il s'étale sur 7 rangs, et sait que chaque rang correspond à une puissance de 2 : le premier (en partant de la **droite**) est le rang 0, le second est le rang 1, etc.
- Pour le convertir en décimal, on procède de la manière suivante : on multiplie par  $2^0$  la valeur du rang 0, par  $2^1$  la valeur du rang 1, par  $2^2$  la valeur du rang 2, [...], par  $2^{10}$  la valeur du rang 10, etc.

- Pour notre nombre 101 0110, on a donc  $0 \times 2^0 + 1 \times 2^1 + 1 \times 2^2 + 0 \times 2^3 + 1 \times 2^4 + 0 \times 2^5 + 1 \times 2^6$ .

Ensuite, il suffit simplement de remplacer les puissances de 2 par leurs valeurs et de faire la somme :  $0 \times 1 + 1 \times 2 + 1 \times 4 + 0 \times 8 + 1 \times 16 + 0 \times 32 + 1 \times 64 = 86$ .

- **donc :  $(101\ 0110)_{\text{bin}} = (86)_{\text{dec}}$**

# L'Hexadécimal

- **Présentation**
- Après le binaire, voici venu une autre base : **le système hexadécimal** qui travaille en base 16.

Si vous avez suivi jusqu'ici, vous devinerez qu'il faudra 16 caractères différents pour représenter chacune des 16 valeurs.

C'est alors qu'avec une originalité déroutante, en hexadécimal, les caractères sont 0, 1, 2 etc. jusqu'à 9 ainsi que A, B, C, D, E et F.

Vous l'aurez compris : A en hexadécimal vaut 10 en décimal, B vaut 11, ... et F vaut 15.

- En hexadécimal, le changement de rang se fait donc à F. Ainsi  $E+1 = F$  et  $F+1 = 10$  (dire “un-zéro”).

Plus compliqué :  $F+B = 1A$ .

**Passons à la conversion !**

- La conversion d'un nombre de la base 10 en base 16 est aussi “facile” qu'avec le binaire. Pour le binaire il fallait décomposer en puissances de 2, ici on décompose en puissances de 16.

# Conversion du décimal en hexadécimal

- Ces puissances de 16 sont :
- $16^0 = 1$
- $16^1 = 16$
- $16^2 = 256$
- $16^3 = 4096$
- $16^4 = 65536$
- ...

- Pour l'exemple, je prendrais le nombre 1680. Il faut donc commencer par le décomposer en puissances de 16 :

$$1680 = 6 \times 256 + 9 \times 16 + 0 \times 1$$

$$1680 = 6 \times 16^2 + 9 \times 16^1 + 0 \times 16^0.$$

La conversion en hexadécimal de 1680 est donc **690** (lire “six-neuf-zéro”).

- Exemple 2 : convertissons 2009 en hexadécimal :
- $2009 = 7 \times 16^2 + 13 \times 16^1 + 9 \times 16^0$ .
- Le nombre en base 16 correspondant à 2009 est donc 7D9 (rappelez vous, chaque rang peut monter jusqu'à 15 en base 16, et le D vaut 13).
- C'est le même principe qu'avec le binaire, le changement de base se fait juste à 16 au lieu de 2.



# Conversion de l'hexadécimal en décimal

- Dans ce sens, c'est plus simple : prenons un nombre : **4F2C**. Il a **4 rangs** : chaque rang est une puissance de 16 : pour convertir, on multiplie le premier rang (en partant de la droite) par  $16^0$ , le second par  $16^1$ , etc.
- Ainsi on obtient :  
$$4F2C = 4 \times 16^3 + F \times 16^2 + 2 \times 16^1 + C \times 16^0$$
$$4F2C = 4 \times 16^3 + 15 \times 16^2 + 2 \times 16^1 + 12 \times 16^0$$
$$4F2C = 4 \times 4096 + 15 \times 256 + 2 \times 16 + 12 \times 1$$
$$4F2C \text{ hex} = 20\ 268 \text{ dec.}$$

# Conversion du binaire en hexadécimal

- La conversion entre l'hexadécimal et le binaire est super facile si vous savez manipuler ces bases entre les nombres **0 et 15**.
- Prenons un nombre en binaire :  
**101 0011 1011**.
- Notez que je l'ai séparé en blocs de 4 chiffres (comme on sépare les nombres en bloc de 3. Par exemple, 30000 s'écrit 30 000).

# Conversion du binaire en hexadécimal

- Ceci nous simplifie la tâche : en effet, on sait que 4 rangs binaires permettent de monter jusqu'à 15. Et bien, 1 rang en hexadécimal aussi ! (Cela vient du fait que  $2^4$  (4 rangs en base 2) =  $16^1$  (1 rang en base 16)).
- De cette façon, **4 bits en binaire** seront représentés par **un rang en hexadécimal** !
- Ainsi, le premier quadruplet : 1011 deviendra un seul rang en hexadécimal :  
 $1011 = 11$  en décimal = B en hexadécimal. Le second quadruplet 0011 devient 3 en hexadécimal ; et finalement le dernier : 101 (ou 0101) devient : 5.
- Ainsi,  $(101\ 0011\ 1011)_{\text{bin}} = (53B)_{\text{hex}}$ .

# Conversion de l'hexadécimal en binaire

- On va utiliser le même principe que ci-dessus, à savoir qu'un rang en base 16 correspond à 4 rangs en base 2.

On convertira le nombre hexadécimal **BE57**. On prend chaque rang que l'on convertit individuellement en binaire :

- $(B)_{\text{hex}} \Leftrightarrow (11)_{\text{dec}} \Leftrightarrow (1011)_{\text{bin}}$
- $(E)_{\text{hex}} \Leftrightarrow (14)_{\text{dec}} \Leftrightarrow (1110)_{\text{bin}}$
- $(5)_{\text{hex}} \Leftrightarrow (5)_{\text{dec}} \Leftrightarrow (0101)_{\text{bin}}$
- $(7)_{\text{hex}} \Leftrightarrow (7)_{\text{dec}} \Leftrightarrow (0111)_{\text{bin}}$

- Prenez bien soin de mettre **0101** au lieu de **101**, car il ne faut pas se tromper quand on va mettre les quadruplets bout à bout :  
**BE57 <=> 1011 1110 0101 0111 .**

# Généralisation à toutes les bases

- C'est tout aussi simple, mais on va utiliser des mathématiques :-).
- Notons simplement que si l'hexadécimal utilise les chiffres de 0 à 9 et les lettres de A à F, les bases plus grandes utilisent la même chose : ainsi la base 18 utilise les caractères 0123456789ABCDEFGHIH.
- Au delà de 36 (on serait à Z), il faudrait utiliser autre chose, par exemple des lettres grecques ( $\alpha, \beta, \gamma, \delta, \epsilon \dots$ ), russes, etc. Mais on s'en passera.
- J'ajoute aussi qu'il est parfaitement inutile d'apprendre à convertir dans toutes les bases.
- Seules les bases 10 (décimal), 16 (hexadécimal), 2 (binaire) et 8 (octal) sont utilisées (en informatique surtout).