

# Entiers Positifs en Binaire et Hédadécimal

## Rappel : Les bases dans la vie courante

Notre système de notation repose sur une disposition en colonnes.

Nous comptons avec 10 symboles allant de 0 à 9 que l'on nomme **chiffre**.

Une fois que nous avons atteint le chiffre 9, si l'on souhaite rajouter 1, on se rend compte que l'on n'a pas de chiffres supérieurs.

On crée ainsi une colonne à gauche qui contiendra un chiffre allant de 1 à 9 en remettant le chiffre de la colonne originelle (ou les suivantes) à 0.

Par exemple 426 est composé des chiffres 4, 2 et 6 représentés dans les colonnes des centaines, dizaines et unités.

## Les bases dans la vie courante

Notre quotidien est entouré de nombres, que ce soit pour les heures qui passent, les notes, le nombre d'oeufs dans une boîte.

Ils sont partout et font partie de nos vies sans même que l'on s'en rende compte.

Mais il existe une quasi infinité de base !

Par exemple, pour les oeufs nous avons souvent des douzaines d'oeufs, issue du fait qu'historiquement nous suivions le compte des oeufs avec le nombre de phalanges que l'on a sur une main (sans compter le pouce car celui ci pointe les phalanges pour compter).

### Exemples:

- Les boîtes d'oeufs sont en base 12.
- Les heures sont divisées en base 60.

En effet il était plus simple de diviser les heures en 60 minutes pour calculer les durées plus facilement. Cela provient des babyloniens qui comptaient dans un système sexagésimal.

Certaines bases ont une représentation des chiffres particulière, qui sont alphanumériques.

- Les Shadok comptent en base 4. Les personnages de ce dessin animé comptent avec 4 chiffres pour écrire leurs nombres : GA, BU, ZO, MEU.
- Le système bibi-binaire est un système en base 16 conçu par le compositeur Boby Lafont en 1968. Cet auteur aimait utiliser des calembours et a voulu créer son système bibi-binaire (comportant système à bibi). Il est proposé d'utiliser des consonnes et des voyelles pour

représenter les nombres : H, B, K, D et A, E, O, I.

L'avantage, c'est que l'on peut *chanter les nombres*.

- L'octal est une base contenant 8 symboles allant de 0 à 7.

Cette base servait aux informaticiens qui développaient en langage machine pour réduire le nombre de bits à écrire.

On l'écrit  $B_8 = \{0, 1, 2, 3, 4, 5, 6, 7\}$

## Définitions

Une base correspond au nombre de symboles qui permettent de représenter les chiffres ou les nombres.

Pour notre base décimale habituelle on écrit :  $B_{10} = \{0, 1, 2, \dots, 9\}$

En informatique, il a été décidé d'utiliser le binaire pour plusieurs raisons.

La principale est **le fait qu'une machine peut facilement détecter la différence entre deux états**. En effet, un processeur est composé de transistors qui traitent deux états : ouverts et fermés à la manière d'un interrupteur.

Le binaire, ou représentation en base 2 est un moyen de représenter les nombres avec 2 symboles ici 0 ou 1.

On peut écrire  $B_2 = \{0, 1\}$ .

On nomme **bit** les chiffres de la représentation en base 2 et un ensemble de 8 bits est appelé un *byte* ou *octet*.

*Question :*

*Combien de nombres peut-on représenter avec  $n$  bits ?*

## Compter en base 2

Pour compter en base 2, on opère de la même manière. On a une colonne qui peut valoir 0 ou 1. Une fois que la colonne atteint 1, on rajoute une colonne à sa gauche à 1 et l'on passe la colonne de droite à 0.

## Comment passer de base 2 à base 10?

*Rappel : Compter en base 10*

$$154_{(10)} = 1 \times 10^2 + 5 \times 10^1 + 4 \times 10^0$$

Pour passer de la base 2 à la base 10, on réalise le même calcul en remplaçant la base 10 par la base 2 :

$$1101_{(2)} = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 8 + 4 + 1 = 13_{(10)}$$

L'écriture ci-dessus n'est pas anodine. A partir de maintenant, pour écrire un nombre si la base n'est pas explicite, il faut la préciser. On peut l'écrire de plusieurs manières :

- $154_{(10)}$  avec la base écrite sous le nombre en parenthèses.
- $\overline{154}^{(10)}$  avec le nombre surligné et la base indiquée entre parenthèses.

**Exercice** : Convertir les nombres de binaire en base décimale

- $1101_{(2)}$
- $1001_{(2)}$
- $1010_{(2)}$
- $1111_{(2)}$

## Comment passer de base 10 à base 2

### Méthode des divisions successives

Pour passer de la base 10 à la base 2, on peut utiliser la méthode que l'on appelle *des divisions successives*.

De la même manière que l'on écrit en addition de puissances de bases (comme vu pour le rappel sur comment compter en base 10), on peut réaliser l'opération inverse pour trouver la représentation binaire.

**Exemple:**

```

29 | 2
   | ---
1  | 14 | 2
    | ---
    0 | 7 | 2
      | ---
      1 | 3 | 2
        | ---
        1 | 1 | 2
          | ---
          1 | 0

```

On divise successivement le nombre à convertir par 2. Chaque **reste** correspond au nombre dans la représentation et chaque **quotient** est à diviser à la suite par 2.

On répète ces opération jusqu'à ce que le quotient soit 0 et le reste 1.

Pour obtenir la représentation on part du reste le plus en bas qui représente le bit le plus à gauche (bit de poids fort) et on remonte pour obtenir le bit le plus à droite (bit de poids faible).

Pour 29 on obtient ainsi :  $11101_2 = 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^0 = 16 + 8 + 4 + 1 = 29_{10}$

**Exercice:** Convertir les nombres en écriture décimale en binaire:

- $27_{10}$
- $14_{10}$
- $42_{10}$
- $33_{10}$

## Méthode des soustractions successives

Une autre méthode utilisable est la méthode des soustractions successives.

Pour ce faire, il suffit de se munir d'un tableau de puissances de 2 et de savoir calculer toutes les puissances de 2 (une calculatrice peut aussi aider pour les puissances élevées).

$2^n$	$2^{n-1}$	$2^{n-2}$	...	$2^2$	$2^1$	$2^0$

On choisit la puissance de 2 supérieure au nombre à convertir et on regarde si on peut retirer chaque puissance de 2.

**Exemple:** Convertir 42 en binaire

On choisit  $64 > 42$  donc on aura un tableau de 7 cases.

Est-ce que  $42 > 64$  ? Non, on met 0 dans la case du tableau.

Est-ce que  $42 > 32$  ? Oui, on met 1 dans la case du tableau. On opère maintenant sur le reste de la soustraction de 42 et 32.

Est-ce que  $10 > 16$  ? Non, on met 0 dans le tableau.

Et ainsi de suite jusqu'à avoir soit 0 au reste soit 1 qui correspond à la dernière case du tableau.

$2^6 = 64$	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
0	1	0	1	0	1	0

On peut aussi passer de binaire à décimal grâce à cette méthode, prenons en exemple  $101110_2$ .

$2^6 = 64$	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
0	1	0	1	0	1	0

Il suffit d'ajouter toutes les puissances de 2 où la case correspondante est à 1.

Ici on aurait :  $2^5 + 2^2 + 2^1 = 42$ , aussi facile que cela.

## Hexadécimal

En informatique, une autre base très importante est la base hexadécimale (ou *hex*). Cette base représente les nombres avec 16 symboles.

Cette base est très utile pour divers usages notamment :

- La représentation d'adresse mémoire
- La représentation des couleurs

On peut noter cette base  $B_{16} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$

On représente chaque symbole de la représentation en base hexadécimale par un ensemble de 4 bits. On peut donc dresser le tableau suivant :

## Convertir en base hexadécimale

### Passage de la base décimale à l'héxadécimale

#### Méthode des divisions successives

A l'instar de la base 2 (et de toutes les bases) pour convertir on peut utiliser la méthode des divisions successives comme développé pour la partie binaire mais en divisant successivement par 16.

#### Passer par la base 2

Une autre méthode serait de convertir le nombre que nous avons en base 10 en binaire.

Ainsi, on regroupe par paquets de 4 bits en partant du bit de poids faible et en rajoutant des 0 si jamais le paquet le plus à droite contient moins de 4 bits.

Enfin, on associe chaque paquet de 4 bits à son équivalent en base 16.

#### Exemple :

On a  $430_{10}$ . On cherche sa représentation en base 2.

On obtient  $1\ 1010\ 1110_2$ .

On rajoute les 0 à gauche pour créer un paquet de 4 et on obtient :  $000110101110_2$ .

Enfin, on associe les bits correspondants à leur représentation hexadécimale:

$$0001_2 = 1_{16}; 1010_2 = A_{16}; 1110_2 = E_{16}$$

On en déduit alors que  $0001\ 1010\ 1110_2 \Leftrightarrow 1AE_{16}$ .

## Passage de la base hexadécimale à la base décimale

### Puissances de 16

On peut en connaissant les puissances de 16, réaliser les additions de puissances de 16 correspondant à chaque symbole du nombre écrit en hexadécimal.

Pour rendre ça compréhensible, on peut remplacer chaque symbole hexadécimal par sa représentation en base 10.

Ainsi ,en reprenant l'exemple précédent:

$$1_{16} = 1_{10}; A_{16} = 10_{10}; E_{16} = 14_{10}$$

$$1AE_{16} = 1 \times 16^2 + 10 \times 16^1 + 14 \times 16^0 = 256_{10} + 160_{10} + 14_{10} = 430_{10}$$

### Passage par la base 2

La première étape est de convertir chaque symbole du nombre en base hexadécimale en base 2.

**Exemple:**

$$1AE_{16} = 0001\ 1010\ 1110_2$$

Ensuite, simplement, on convertit de la base 2 à la base décimale comme vu précédemment.

$$000110101110_2 = 1 \times 2^8 + 1 \times 2^7 + 1 \times 2^5 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 = 430_{10}.$$

**Exercice 1 :** Conversion de l'hexadécimal au décimal :

Convertir le nombre hexadécimal  $1F3_{16}$  en décimal.

**Exercice 2 :** Conversion du décimal à l'hexadécimal :

Convertir le nombre décimal  $393_{16}$  en hexadécimal.

[Fiche d'exercice entiers en binaire et hexadecimal](#)