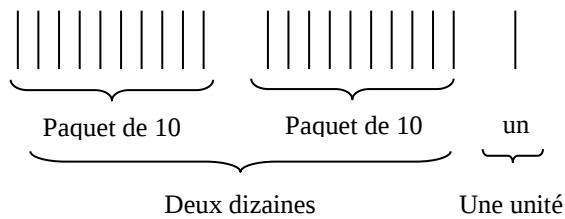


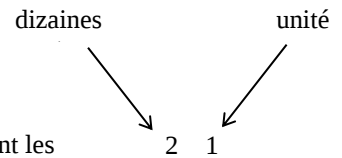
I. Expérience d'introduction

Expérience : On vide une trousse de ses stylos et on les compte. (par ex : 21)

- On peut les regrouper en paquets de 10 :

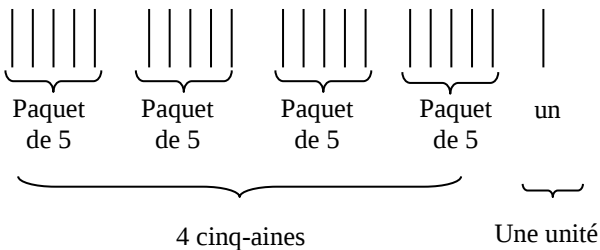


En base 10 en utilisant les chiffres arabes :

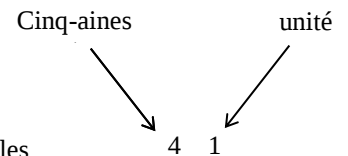


On regroupe ensuite les paquets de 10 en 10 « paquets de dix », soit 10 dizaines : on obtient les centaines. Puis en 10 centaines, on obtient les milliers etc...

- On peut les regrouper en paquets de 5 :

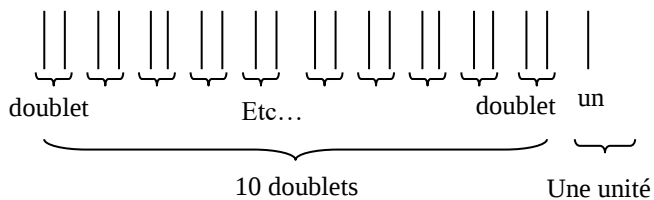


En base 5 en utilisant les chiffres arabes :

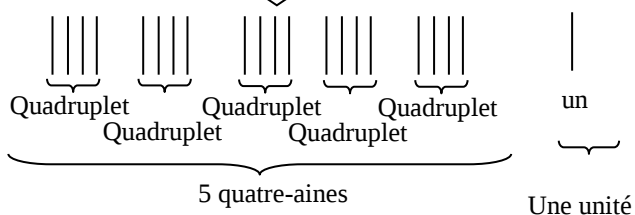


On regroupe ensuite les paquets de 5 en 5 « paquets de 5 », soit 5 cinq-aines : on obtient les vingt-cinq-aines. Puis en 5 vingt-cinq-aines, on obtient les cent-vingt-cinq-aines etc...

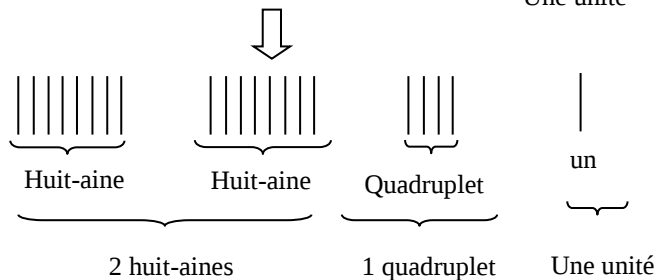
- On peut les regrouper en paquets de 2 :



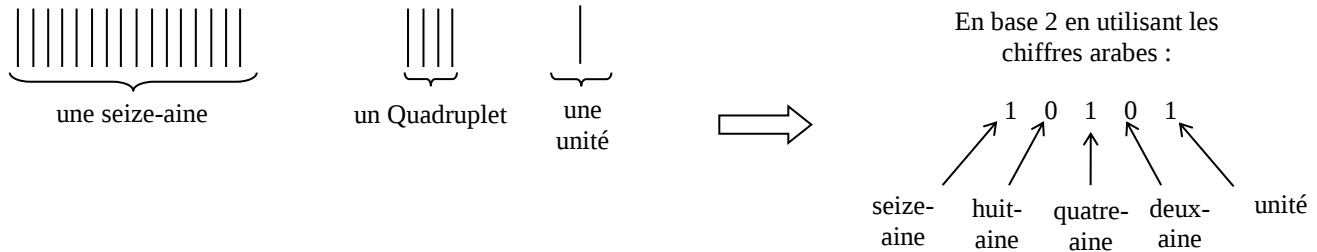
On regroupe ensuite les paquets de 2 en 2 « paquets de 2 », soit des quadruplets : on obtient les quatre-aines



Puis en 2 quatre-aines, on obtient les huit-aines ...



Puis en 2 huit-aines, on obtient les seize-aines etc...



Conclusion :

- Le concept de nombre est indépendant de sa représentation graphique. Ici : $21^{10} = 41^5 = 10101^2$. La représentation graphique d'un nombre utilise des symboles conventionnels (généralement les chiffres arabes et les lettres) dont la signification dépend de la base choisie.
- Dans la représentation de position l'ordre des symboles est fondamental dans la lecture du nombre considéré.

II. La base décimale

Exemple : (demander l'année de naissance)

$$\begin{array}{ccccccc}
 & & & \text{millier} & \text{centaines} & \text{dizaines} & \text{unité} \\
 & & & \downarrow & \downarrow & \downarrow & \downarrow \\
 a = 1 & 9 & 9 & 7 & = & 1 \times 10^3 + 9 \times 10^2 + 9 \times 10^1 + 7 \times 10^0 \\
 \nearrow & \nearrow & \nearrow & \nearrow & & & \\
 10^0 & 10^0 & 10^0 & 10^0 & & &
 \end{array}$$

Conclusion : Tous les nombres en base 10, en représentation de position, s'écrivent comme des mots composés des dix caractères suivant : $\{0,1,2,\dots,9\}$.

III. La base binaire

3.1 Définition

Tous les nombres en base binaire, en représentation de position, s'écrivent comme des mots composés des 2 caractères suivant : $\{0,1\}$.

3.2 Analyse de la représentation

Exemple : Convertissons le nombre binaire précédemment utilisé dans la base décimale.

$$\begin{aligned}
 b &= \overline{1\ 0\ 1\ 0\ 1}^2 = 1 \times 16 + 0 \times 8 + 1 \times 4 + 0 \times 2 + 1 \\
 &= 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\
 &= \overline{21}^{10}
 \end{aligned}$$

Un tableau récapitulatif (**tab. 1**) des puissances de deux est très utile pour effectuer la conversion :

2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8	2^9	2^{10}
1	2	4	8	16	32	64	128	256	512	1024

Exercice : convertir le nombre $c = \overline{1\ 0\ 0\ 1\ 1\ 0\ 1}^2$ en représentation décimale.

$$c = \overline{1\ 0\ 0\ 1\ 1\ 0\ 1}^2 = 1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 64 + 0 + 0 + 8 + 4 + 0 + 1 = \overline{77}^{10}$$

3.3 Bits et octets

Définitions :

- Un caractère binaire est un « bit ».
- Un mot de 8 bits est un octet.

Exercice : Jusqu'à combien peut-on compter (en base 10) avec un octet ?

Base 2	Base 10
0 0 0 0 0 0 0 0	0
0 0 0 0 0 0 0 1	1
0 0 0 0 0 0 1 0	2
0 0 0 0 0 0 1 1	3
...	...
1 1 1 1 1 1 1 1	255

Démo :

$$\begin{aligned} 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1^2 &= 1 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\ &= 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255^{10} \end{aligned}$$

3.4 Conversion de la base 2 à la base 10

Exemple : On considère un nombre $x = 61^{10}$. Quelle est la représentation de ce nombre en base 2 ?

1^{ère} méthode : On cherche la puissance de 2 qui se rapproche le plus de $x = 61$ par valeur inférieure. D'après le tableau **tab. 1**, on trouve :

$$x = 32 + R_0 = 32 + 29$$

On réitère la même opération avec le reste R_0 : on cherche ensuite la puissance de 2 qui se rapproche le plus de $R_0 = 29$ par valeur inférieure. On trouve :

$$x = 32 + 16 + R_1 = 32 + 16 + 13$$

On réitère la même opération avec le reste R_1 , etc... On trouve :

$$x = 32 + 16 + 8 + 5 = 32 + 16 + 8 + 4 + 1.$$

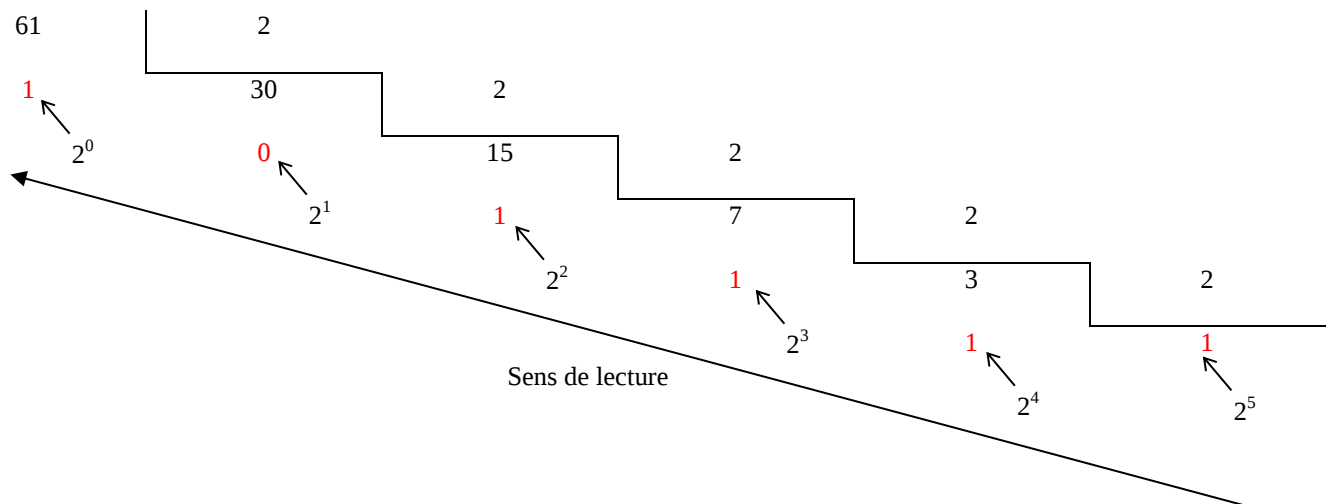
On écrit les puissances de deux correspondantes :

$$x = 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = \underline{1 \ 1 \ 1 \ 1 \ 0 \ 1^2}.$$

2^{ème} méthode (la meilleure) : On écrit les divisions euclidiennes successives par 2. Cette méthode est la plus générale et peut s'appliquer une base numérique quelconque.

On rappelle que la division euclidienne de $x = 61$ par 2 s'écrit de la manière suivante : $x = 2 \times Q + R$ dans lequel Q est le quotient de la division euclidienne et R est le reste de la division euclidienne. Ici $x = 2 \times 30 + 1$, donc $Q = 30$ et $R = 1$.

La manière d'écrire est la suivante :



On a donc :

$$\begin{aligned}
 x=61 &= 2 \times 30 + 1 &= 2 \times (15 \times 2 + 0) + 1 &= 15 \times 2^2 + 1 \times 2^0 = 15 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\
 &= (7 \times 2 + 1) \times 2^2 + 1 &= 7 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\
 &= (2 \times 3 + 1) \times 2^3 + 1 \times 2^2 + 1 \times 2^0 = 3 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\
 &= (2 \times 1 + 1) \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\
 &= \overline{111101}^2
 \end{aligned}$$

On peut lire directement le résultat en partant du dernier reste des différentes divisions euclidiennes successives, puis en lisant les différents quotients des divisions en montant vers 61.

IV. La base hexadécimale

4.1 Exemples

➤ Considérons le nombre $a = \overline{83}^{10}$. Quelle est sa représentation en base 16 ?

On utilise la méthode 2 :

$$\begin{array}{r|l}
 83 & 16 \\
 \hline
 3 & 5 \\
 \uparrow & \uparrow \\
 16^0 & 16^1
 \end{array}$$

Soit $a = \overline{83}^{10} = 5 \times 16 + 3 = \overline{53}^{16}$

➤ Considérons le nombre $b = \overline{2547}^{10}$. Quelle est sa représentation en base 16 ?

On utilise la méthode 2 :

$$\begin{array}{r|l}
 2547 & 16 \\
 \hline
 3 & 159 & 16 \\
 \uparrow & \uparrow & \uparrow \\
 16^0 & 16^1 & 16^2 \\
 & 15 & 9
 \end{array}$$

Soit $b = \overline{2547}^{10} = 159 \times 16 + 3 = (9 \times 16 + 15) \times 16 + 3 = 9 \times 16^2 + 15 \times 16 + 3 = \overline{9153}^{16}$

Cette représentation n'est pas correcte car l'écriture « 15 » est trompeuse. On pourrait croire que 15 représente $1 \times 16^2 + 5 \times 16$ alors qu'il représente 15×16^1 . Par conséquent la base décimale $\{0, \dots, 9\}$ n'est pas adaptée pour représenter la base hexadécimale. Celle-ci doit être définie par **16 caractères différents** :

<u>Base 10</u>	<u>Base 16</u>
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	A
11	B
12	C
13	D
14	E
15	F

Le nombre b peut donc s'écrire de la manière suivante :

$$b = \overline{2547}^{10} = \overline{9F3}^{16}$$

Conclusion :

- La base hexadécimale est composée de 16 caractères différents : $\{0, \dots, 9, A, \dots, F\}$.
- Les nombre dans cette base sont des mots composés de ces 16 caractères, en représentation de position.

4.2 Avantages et utilités de la représentation hexadécimale

Problème 1 : Combien de bits un caractère hexadécimal nécessite-t-il pour être représenté dans la base binaire ?

Avec un caractère hexadécimal, il est possible de compter au maximum jusqu'à 15 (en base 10) : $\overline{15}^{10} = \overline{F}^{16}$.

Dans la base binaire le nombre $\overline{15}^{10}$ est représenté par :

$$\begin{array}{rcl}
 15 & | & 2 \\
 1 & | & 7 \quad 2 \\
 & | & 1 \quad 3 \quad 2 \\
 & | & 1 \quad 1
 \end{array}$$

Par conséquent : $\overline{15}^{10} = \overline{F}^{16} = \overline{1111}^{16}$.

Conclusion : Il faut 4 bits pour coder un caractère hexadécimal. La représentation hexadécimale est donc plus compacte que la représentation binaire, et que la représentation décimale.

Problème 2 : Quel est le nombre le plus élevé qu'il est possible d'atteindre avec deux caractères hexadécimal ? Combien de bits deux caractères hexadécimaux nécessitent-ils pour être représenté dans la base binaire ?

Le nombre le plus élevé représenté par deux caractères hexadécimaux est :

$$\overline{FF}^{16} = 15 \times 16^1 + 15 \times 16^0 = 15 \times 16 + 15 = \overline{255}^{10} = \overline{11111111}^2$$

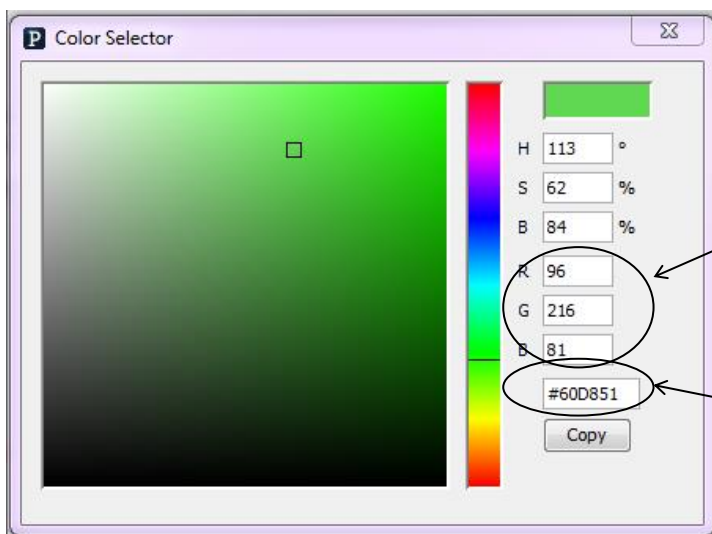
8 bits

Nombre maximal qu'il est possible d'atteindre avec 1 octet !!

Conclusion : 2 caractères hexadécimaux correspondent à 1 octet !!

Application à l'image numérique :

Sous Processing un outil intéressant pour gérer la couleur est le « color selector ». Le menu permet de remarquer que chaque couleur est associée à 3 nombre en base 10 compris entre 0 et 255 (rouge - R, vert - G, bleu - B), et aussi à code html, commençant par « # », composé de chiffres entre 1 et 9 et de lettres entre A et F.



Coordonnées dans l'espace des couleurs :
(rouge, vert, bleu) = (R, V, B) = (96, 216, 81)

3 nombres compris entre 0 et 255,
soit 3 octets

Code HTML de la couleur : # 60 D8 51

Le code HTML des couleurs est composé de 6 caractères hexadécimaux, ce qui représente 3 octets. Pour comprendre cette représentation, cherchons la valeur des trois nombres hexadécimaux $\overline{60}^{16}$, $\overline{D8}^{16}$ et $\overline{51}^{16}$:

- $\overline{60}^{16} = 6 \times 16 + 0 \times 1 = \overline{96}^{10}$ → valeur du rouge
- $\overline{D8}^{16} = 13 \times 16 + 8 \times 1 = \overline{216}^{10}$ → valeur du vert
- $\overline{51}^{16} = 5 \times 16 + 1 \times 1 = \overline{81}^{10}$ → valeur du bleu

Conclusion : La base hexadécimale est utilisée pour coder de façon compacte les couleurs des pixels d'une image, mais aussi les caractères ascii des claviers numériques...