**Activité**

**IMAGES NUMERIQUES**

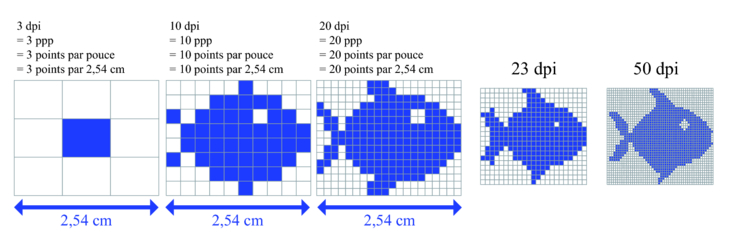
Une image numérique est constituée d'un ensemble de points appelés pixels (abréviation de ***pic***ture***el***ement) pour former une image. Le pixel est le plus petit élément constitutif d'une image numérique. L'ensemble de ces pixels est contenu dans un tableau à deux dimensions constituant l'image.

**Définition d’une image**

La définition de l'image est le nombre de points (pixels) constituant une image : c'est le nombre de colonnes de l'image que multiplie son nombre de lignes.  
Par exemple, une image possédant 10 colonnes et 11 lignes aura une définition de 10×11.

**Résolution d’une image**

La résolution est le nombre de points contenu dans une surface précise (en pouce). Elle est exprimée en points par pouce (PPP), correspondant à la notation anglaise DPI (*dots per inch*), plus répandue. Un pouce mesure 2,54 cm.  
La résolution correspond au rapport définition / dimension.



Quelques définitions d'appareils courants :

* écrans d'ordinateur : 72 dpi.
* imprimantes : entre 360 dpi et 1 400 dpi.
* matériel d'impression professionnel : au minimum 4 800 dpi.

**Codage d'une image en 2 couleurs**

On réalise un codage sur 1 bit par pixel (bpp). Il y aura donc deux possibilités : soit 0, soit 1, donc deux couleurs possibles, blanc ou noir.

**Codage d'une image en couleur avec des pixels en couleur**

Chaque pixel comporte trois sous-pixels : rouge, vert, bleu (RVB). Chaque sous-pixel est codé sur 1 octet (8 bits).

Chaque sous-pixel aura donc 256 possibilités (28).

Comme on a 8 bits pour le rouge, 8 bits pour le vert et 8 bits pour le bleu, on utilisera 3×8 bits, soit 24 bits utilisés au total. On parle alors de codage RVB 24 bits.  
Or, chaque pixel peut prendre 256 × 256 × 256 = 224, soit 16,7 millions de couleurs.  
Une image numérique est codée par un tableau de nombres : chaque case du tableau correspondra au pixel qui aura la même place sur le tableau, et chaque case aura un nombre binaire de 24 bits (3 octets) qui donnera la couleur précise en rouge, vert et bleu.

**Conversion décimal-binaire**

#### **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**×16 + **1**×8 + **1**×2

26= **1**×24 + **1**×23 + **1**×21 (on écrit les coefficients sous forme de puissances de 2)

26= **1**×24 + **1**×23 + **0**×22 + **1**×21 + **0**×20 (on ajoute les puissances de 2 qui manquent)

26= **1**×2**4** + **1**×2**3** + **0**×2**2** + **1**×2**1** + **0**×2**0** (voyez les puissances de 2 qui sont toutes là)

26= **1**×24 + **1**×23 + **0**×22 + **1**×21 + **0**×20 (en orange : notre nombre en binaire !)

Remarque : il est important de ne pas oublier les puissances dont les coefficients sont zéro.  
  
Finalement, 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. Ainsi (26)dec = (11010)bin

#### **Méthode 2 : les divisions euclidiennes par 2**

Cette méthode est plus adaptée aux grands nombres et est plus facile à utiliser en programmation (il est facile d'en faire un algorithme). Méthode :

* On le divise le nombre décimal 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 réitère 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.

Exemple : 164

* 164 ÷ 2 = 82 + **0**
* 82   ÷ 2 = 41 + **0**
* 41   ÷ 2 = 20 + **1**
* 20   ÷ 2 = 10 + **0**
* 10   ÷ 2 = 5  + **0**
* 5   ÷ 2 = 2  + **1**
* 2   ÷ 2 = 1  + **0**
* 1   ÷ 2 = **0**  + **1**

On voit apparaître le nombre binaire à droite. **Il faut le lire de bas en haut**.

**EXERCICE**

Etablir les tableaux de nombres de codage des deux images ci-dessous.

|  |  |
| --- | --- |
|  |  |
| Codage sur 1 bit | Codage sur 24 bits |

**Calculs**

………………………………………………………………………………………………………………………………………………………………………….……………………………………………………………………………………………………………………………………………………………….………………

…………………………………………………………………………………………………………………………………………………………….………………

…………………………………………………………………………………………………………………………………………………………….………………

…………………………………………………………………………………………………………………………………………………………….…………………………………………………………………………………………………………………………………………………………………………….…………………………………………………………………………………………………………………………………………………………………………….………………

…………………………………………………………………………………………………………………………………………………………….………………

…………………………………………………………………………………………………………………………………………………………….………………

…………………………………………………………………………………………………………………………………………………………….………………

…………………………………………………………………………………………………………………………………………………………….………………

…………………………………………………………………………………………………………………………………………………………….…………………………………………………………………………………………………………………………………………………………………………….………………

…………………………………………………………………………………………………………………………………………………………….………………

…………………………………………………………………………………………………………………………………………………………….………………

…………………………………………………………………………………………………………………………………………………………….………………

…………………………………………………………………………………………………………………………………………………………….………………

…………………………………………………………………………………………………………………………………………………………….…………………………………………………………………………………………………………………………………………………………………………….………………

…………………………………………………………………………………………………………………………………………………………….………………

…………………………………………………………………………………………………………………………………………………………….………………

…………………………………………………………………………………………………………………………………………………………….………………

…………………………………………………………………………………………………………………………………………………………….………………

…………………………………………………………………………………………………………………………………………………………….…………………………………………………………………………………………………………………………………………………………………………….………………

…………………………………………………………………………………………………………………………………………………………….………………

…………………………………………………………………………………………………………………………………………………………….………………

…………………………………………………………………………………………………………………………………………………………….………………



|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |



|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

