



Objectifs.

Comprendre les images numériques élémentaires.

Mots clés.

Pixel, RGB (RVB), décimal, binaire, bit, octet (o, ko, Mo, Go, etc.), poids d'une image.

Consigne 1 individuel (10 minutes)

Convertir en décimal le nombre binaire : **1100101**

Aide : fichier [décimal binaire.pdf] (copie ci-dessous).

<http://www.ma-calculatrice.fr/convertir-binaire-hexadecimal.php>

Système décimal Base 10
10 chiffres : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Système binaire	Base 2
2 chiffres (2 bits) :	0 et 1
On compte : 0, 1, 10 , 11, 100 , 101, 110, 111, 1000 , 1001, 1010, 1011, 1100, 1101, 1110, 1111, 10000 , etc.	
1011 = $1_x 2^3 + 0_x 2^2 + 1_x 2^1 + 1_x 2^0$	
<i>En décimal : $8+0+2+1=11$</i>	
Avec 4 bits on peut écrire $2^4 = 16$ nombres : de 0 à 1111	
<u>Avec n bits, on peut écrire 2^n nombres</u>	
donc en 8 bits (octet) : $2^8 = 256$ (de 0 à 255)	

Consigne 2 groupe de 3 (**30 minutes**)

Fabrication « à la main » d'un fichier bitmap élémentaire.

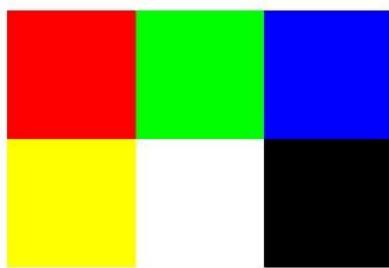
Utiliser le document [[bitmap.pdf](#)] et ouvrir le fichier [[bitmap.ppm](#)] avec XnView (ou autre logiciel de traitement d'image) ; il faut zoomer...

Fichier image bitmap élémentaire

```
P3
3 2
255
255 0 0    0 255 0      0 0 255
255 255 0   255 255 255  0 0 0
```

Le P3 signifie que les couleurs sont en ASCII (L'American Standard Code for Information Interchange : norme de codage de caractères),
par 3 colonnes et 2 lignes,
ayant 255 pour valeur maximum, et qu'elles sont en RGB (RVB).

Résultat :



Ecrire dans Wordpad (bloc-notes ou autre) un fichier qui représente le dessin de votre choix... Enregistrer avec l'extension **.ppm** et visualiser avec XnView (ou autre logiciel de traitement d'image).

Aide : document [[codage RGB.pdf](#)]

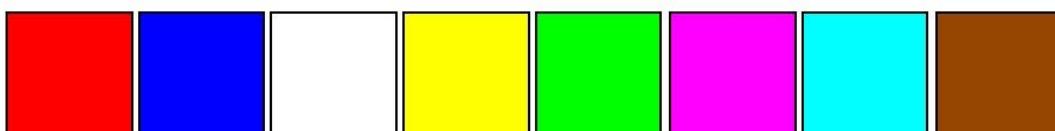
Pixel.

Une image est constituée d'un ensemble de points appelés **pixels** (pixel est une abréviation de *PICTure EElment*). Le pixel représente ainsi le plus petit élément constitutif d'une image numérique.

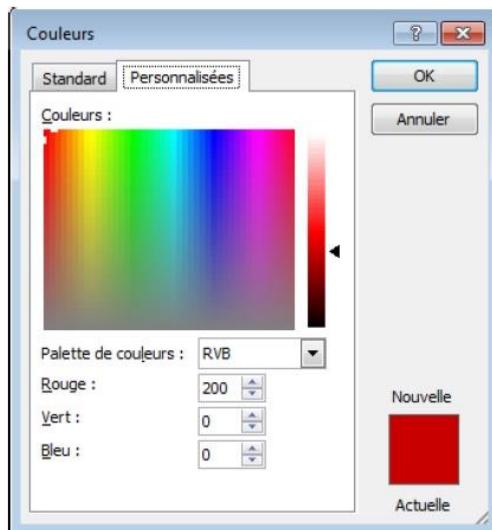


Codage de l'image bitmap 24 bits (3 octets).

Les couleurs d'écran sont obtenues par combinaison de Rouge, Vert et Bleu (RVB). Pour chaque pixel le codage de la couleur est réalisé par trois nombres compris entre 0 et 255 (trois « octets »). Le nombre de couleurs différentes pouvant être ainsi représenté est :
 $256 \times 256 \times 256 \approx 16,8 \times 10^6$ donc plus de 16 millions de couleurs



R	255	0	255	255	0	255	0	150
V	0	0	255	255	255	0	255	70
B	0	255	255	0	0	255	255	0



Poids d'une image non compressée :

Exemple : Img1.ppm 1536 x 1024 x 24 bits = 4,5 Mo (1 Mo = 1024 x 1024 bits)

Consigne 3 individuel (20 minutes)

1. Visualiser le fichier [img2.ppm] et son code avec Wordpad (bloc-notes) : « colorier » en rouge le coin en haut à gauche de cette image en modifiant le code (enregistrer puis visualiser).

2. **Interpréter les données** des images contenues dans le fichier **[img1.pdf]** ; visualiser les fichiers image **[img1.ppm]** et **[img1r.ppm]** avec XnView (ou autre) et retrouver ces données dans - **édition – propriétés**
3. **Comparer avec les données** de **[img1.ppm]** et **[img1.jpg]** et **interpréter**.



img1.ppm 4,5 Mo 1536 x 1024 x 24 RGB



img1 recadrée.ppm 37,82 ko 389 x 384 x 24 RGB

Compléments :

<http://info.univ-angers.fr/~gh/Farcompr/rapport-images.pdf>

<http://www.convertir-une-image.com/optimiser/compresser-une-photo-en-ligne.asp>