

Qu'est-ce qu'une image 24 bits ?

Une image numérique est composée de **pixels** (petits carrés de couleur).

Chaque pixel est défini par trois valeurs :

- **R** : Rouge (Red)
- **V** : Vert (Green)
- **B** : Bleu (Blue)

Par exemple : `image.Pixel(255,0,0)` correspond à un pixel rouge.

Pourquoi 24 bits ?

Chaque couleur est codée sur **8 bits**.

Avec 8 bits, on peut représenter : $2^8 = 256$ valeurs

soit les nombres : 0 à 255

- 0 = absence de couleur
- 255 = intensité maximale

Comme un pixel possède trois couleurs :

Rouge : 8 bits

Vert : 8 bits

Bleu : 8 bits

on obtient : $8 + 8 + 8 = 24$ bits. On parle donc d'**image 24 bits**.

Combien de couleurs ?

Chaque couleur possède 256 valeurs possibles : $256 \times 256 \times 256 = 16\,777\,216$ couleurs, soit plus de **16 millions de couleurs**.

Dans un programme Python

Pour chaque pixel, on peut récupérer les trois composantes : `r, g, b = oldimage.getPixel(col,row)`

Puis on peut les modifier pour créer des effets, par exemple :

- supprimer le rouge ;
- mettre l'image en noir et blanc ;
- créer un négatif ;
- appliquer un filtre personnalisé.

Le principe est toujours le même : lire les valeurs Rouge, Vert et Bleu d'un pixel, les modifier, puis créer un nouveau pixel.