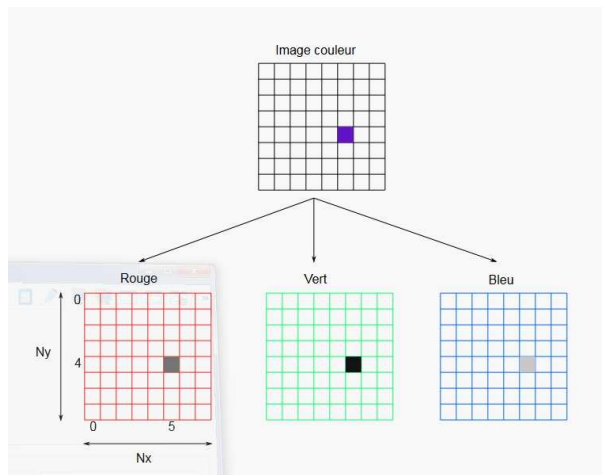


# Les images

## Lecture d'une image.

Les images fournis par les appareils photo sont généralement en couleur. Une image en couleur est constituée de trois couche : une couche rouge (R), une couche verte (G) et une couche bleue (B), R, G et B sont des entiers entre 0 et 255 ou des flottants entre 0 et 1, mesurant l'intensité du rouge.



Si  $l$  est le nombre de lignes de l'image et  $c$  le nombre de colonne, une image est donc constituée de  $l \times c$  pixels, chaque pixel étant lui-même un vecteur de trois valeurs (R, G, B).

On peut donc stocker une image sous la forme d'une matrice à trois dimension :  $l \times c \times 3$ .

Si on a une simple image en nuance de gris (il n'y a qu'une couleur, le tableau est alors de dimension deux :  $l \times c$ ).

Pour les images en RGBA, on rajoute au pixel une quatrième valeur qui est le degré de transparence. (dimensions  $l \times c \times 4$ ).

Outre ce codage de l'image, le fichier est aussi compressé afin de tenir moins de place. Le format sans compression est le format .bmp, le format de compression usuel est le format .jpg, qui n'est pas un format libre de droit, même s'il est gratuit, le format des logiciels libres est le format .png

## Convertir une image au format .png

Avec matplotlib.pyplot, nous ne pouvons travailler qu'avec des fichiers au format .png

Pour nous faciliter les manipulations, nous allons utiliser la bibliothèque PIL (Python Imaging Library) `from PIL import Image`

Quelque soit l'image que l'on souhaite manipuler, on va le transformer au format .png grâce à PIL.

```
from PIL import Image
nom = 'image'
img = Image.open(nom+'.jpg')
img.save(nom+'.png') # enregistrement de l'image au format .png
```

On peut le faire également avec le module imageio de pyzo

```
import imageio as iio
Nom = 'image'
img = iio.imread(nom+'.jpg')
iio.imsave(nom+'.png',img)# enregistrement de l'image au format .png
```

## Travailler les images .png

On utilise matplotlib.pyplot pour la gestion des images (`import matplotlib.pyplot as plt`) et numpy pour modifier la matrice en mémoire (`import numpy as np`)

Dans la plupart des formats, les pixels correspondent à des triplets d'entiers entre 0 et 255 (1 octet = 256 valeurs), mais dans le format png, il s'agit de triplet de réels entre 0 et 1.

### Travailler une image consiste à :

**ouvrir le fichier** et le charger en mémoire (`plt.imread()`) syntaxe `plt.imread() : img=plt.imread(nom+'.png')`

**le modifier** : c'est une matrice numpy de type array

**afficher l'image** `plt.imshow()` et `plt.show()`

**syntaxe** `plt.imshow(img) ; plt.show()`

    Attention : syntaxe particulière pour les niveaux de gris

**syntaxe** `plt.imshow(img, cmap=cm.gray) ; plt.show()`

**enregistrer l'image** `plt.imsave()`

**syntaxe** `plt.imsave(nom+'.png',img)`