

Les images qu'on va utiliser sont des images dites matricielles. Elles sont composées d'une matrice (tableau) de points colorés appelés **pixels**. Chaque pixel est un triplet de nombres entre 0 et 255 : un nombre pour chaque couleur primaire rouge, vert, bleu. Un tel nombre est représentable sur 8 bits (1octet).

Il y a donc  $256^3 = 2^{24} = 16777216$  couleurs possibles. On utilise ici la synthèse additive des couleurs : le triplet (0, 0, 0) correspond à un pixel noir alors qu'un pixel blanc est donné par (255, 255, 255). Un pixel « pur rouge » est codé par (255, 0, 0).

Le script Python suivant indique comment récupérer un tableau numpy donnant un tableau tridimensionnel, de taille  $h \times l \times 3$  où  $h$  est la hauteur de l'image et  $l$  est sa largeur.

```
from PIL import Image #importation du sous-module Image du module PIL

im=Image.open("\chemin\vers\image.png") #ouverture d'une image au format png dans Python.

tab=np.array(im)
```

Dans la suite, on va jouer avec les images en les modifiant, Pour cela, il suffit de modifier le tableau numpy associé, puis le convertir en image et l'afficher ou l'enregistrer :

```
nouvelle_image=Image.fromarray(tab)

nouvelle_image.show() # pour afficher l'image

nouvelle_image.save("nom_de_la_nouvelle_image.png") # pour l'enregistrer au format voulu
```

**NB :** Le tableau numpy tab doit être composé d'éléments **uint8** (c'est-à-dire d'entiers non signés codés sur 8 bits, soit entre 0 et 255).

### Effet Miroir :

**Question1.** Ecrire une fonction **miroir(A)** qui renvoie la transformée de la matrice image A par la symétrie d'axe vertical ?

### Effet Rotation :

**Question 2.** Écrire une fonction **rot90(A)** qui renvoie la transformée de la matrice image A par rotation de 90 degré.

### Effet Niveau de gris

Dans une image en niveaux de gris, les trois composantes R, V, B de chaque pixel ont la même valeur. Celle-ci vaut 0 pour un pixel noir, 255 pour un pixel blanc.

L'œil est plus sensible à certaines couleurs qu'à d'autres. Le vert (pur), par exemple, paraît plus clair que le bleu (pur). Pour tenir compte de cette sensibilité dans la transformation d'une image couleur en une image en niveaux de gris, on ne prend généralement pas la moyenne arithmétique des intensités de couleurs

fondamentales, mais une moyenne pondérée. La formule standard donnant le niveau de gris en fonction des trois composantes est :

$\text{gris} = \lfloor 0.299 \cdot \text{rouge} + 0.587 \cdot \text{vert} + 0.114 \cdot \text{bleu} \rfloor$ , où  $\lfloor . \rfloor$  désigne la partie entière.

**Question 3.** Ecrire une fonction **niveau\_gris(A)** qui renvoie la transformée de la matrice Image couleur A en niveau de gris. Les trois niveaux R, V, B d'un pixel sont égaux au niveau de gris donné par la formule ci-dessus ?

Dans une image en niveau de gris, les niveaux de rouge, vert et bleu étant identiques, les informations sont redondantes. Il existe un format (« B ») pour stocker des images en noir et blanc (inutile de répéter trois fois la même valeur !). Il est possible de sauvegarder la matrice des pixels non pas en associant à chaque pixel un triplet de niveau (R,V,B) mais une valeur unique égale au niveau de gris.

**Question 4.** Ecrire une fonction **formatB(A)** qui renvoie la transformée de la matrice Image couleur en format B ?

### Effet Fusion

Il s'agit ici de fusionner deux images de même dimension. Pour chaque pixel de l'image fusionnée, chacune de ses composantes est égale à la composante de valeur la plus élevée des deux pixels des images à fusionner.

**Question 5.** Ecrire une fonction **fusion(A,B)** qui renvoie la fusion des deux matrices Image A et B ?

### Filtre Moyenneur

Le filtre moyenneur est une opération de traitement d'images utilisée pour réduire le bruit dans une image et/ou flouter une image. Concrètement, la valeur filtrée d'un pixel p est égale à la moyenne des valeurs des pixels voisins.

**Question 6.** Ecrire une fonction **moyenneur (A)** qui renvoie la transformée de la matrice Image A par le filtre moyenneur ?

### Convolution

La *convolution*, ou *produit de convolution*, est une généralisation du filtre moyenneur où l'on considère cette fois une moyenne pondérée selon une image qui contient les coefficients de pondération. On l'appelle généralement *noyau de convolution* ou *masque de convolution*.

**Question 7.** Ecrire une fonction **convolution (A , F )** qui renvoie la convolution de la matrice Image A par le masque F ?

### Histogramme d'une image (Format B)

L'histogramme d'une image permet de compter le nombre de pixel d'un niveau de gris donné.

**Question 8.** Écrire une fonction **histogramme(A)** qui trace la courbe histogramme de la matrice image A où les abscisses correspondent aux niveaux de gris et les ordonnées les nombres de pixel ?