

**TD INFO08 – CORRECTION  
MANIPULATION D'IMAGES**

```
import matplotlib.pyplot as plt
import numpy as np
import math as m

image = plt.imread("ara.png")
plt.figure("couleur")
plt.imshow(image)

(hauteur, largeur, ncouleur) = image.shape

# Exercice n°1 : Filtres rouge, vert et bleu

image_rouge = np.zeros((hauteur, largeur, 3))
image_vert = np.zeros((hauteur, largeur, 3))
image_bleu = np.zeros((hauteur, largeur, 3))

image_rouge[:, :, 0] = image[:, :, 0]
image_vert[:, :, 1] = image[:, :, 1]
image_bleu[:, :, 2] = image[:, :, 2]

plt.figure("rouge")
plt.imshow(image_rouge)
plt.figure("vert")
plt.imshow(image_vert)
plt.figure("bleu")
plt.imshow(image_bleu)
plt.show()

# Exercice n°2 : Niveaux de gris

# gris moyen
R = image[:, :, 0]
G = image[:, :, 1]
B = image[:, :, 2]
G_moyen = (R+G+B)/3
plt.figure("gris moyen")
plt.imshow(G_moyen, cmap="gray")

# option : clarté
G_clarté = np.zeros((hauteur, largeur))
for i in range(hauteur):
    for j in range(largeur):
        gris = (max(image[i, j]) + min(image[i, j])) / 2
        G_clarté[i, j] = gris
plt.figure("gris clarté")
plt.imshow(G_clarté, cmap="gray")
```

```
# option : luminosité
G_luminosité = 0.21*R+0.71*G+0.07*B
plt.figure("gris luminosité")
plt.imshow(G_luminosité,cmap="gray")
plt.show()
```

### # Exercice n°3 : noir et blanc

```
image_noir_blanc = np.zeros((hauteur,largeur))

seuil = 0.4
for i in range(hauteur):
    for j in range(largeur):
        if G_luminosité[i,j] > seuil:
            image_noir_blanc[i,j] = 1

plt.figure("noir et blanc")
plt.imshow(image_noir_blanc,cmap="gray")
plt.show()
```

### # Exercice n°4 : miroir

```
image_miroir = np.zeros((hauteur,largeur,3))

for i in range(hauteur):
    for j in range(largeur):
        image_miroir[i,j] = image[i,largeur-j-1]

plt.figure("miroir")
plt.imshow(image_miroir)
plt.show()
```

### # Exercice n°5 : amélioration du contraste

```
image_contraste = 0.5+0.5*np.sin((m.pi*(G_luminosité-0.5)))
plt.figure("image constratée")
plt.imshow(image_contraste,cmap="gray")
plt.show()
```

### # Exercice n°6 : détection bord

```
image_bord = np.zeros((hauteur,largeur))
seuil = 0.1
for i in range(1,hauteur-1):
    for j in range(1,largeur-1):
        valeur = m.sqrt((G_luminosité[i,j-1]-G_luminosité[i,j+1])**2
+ (G_luminosité[i-1,j]-G_luminosité[i+1,j])**2)
        if valeur < seuil:
            image_bord[i,j] = 1
plt.figure("image bord")
plt.imshow(image_bord,cmap="gray")
plt.show()
```