

En vous inspirant de ce qui a été fait précédemment, écrivez un programme qui inverse les valeurs des canaux bleu et rouge sans changer la valeur du canal vert.

En vous inspirant de ce qui a été fait précédemment et après avoir fait quelques recherches sur le "négatif d'une image", écrivez un programme qui donne le négatif d'une image.

En vous inspirant de ce qui a été fait précédemment et après avoir fait quelques recherches sur les "images en niveau de gris", écrivez un programme qui transforme une "image couleur" en une "image en niveau de gris".

Petite astuce qui pourrait vous aider : en Python pour avoir une division entière (le résultat est un entier), il faut utiliser l'opérateur // à la place de l'opérateur /

En vous inspirant de ce qui a été fait précédemment testez les programmes écrits dans avec une image de votre choix (attention aux variables "largeur\_image" et "hauteur\_image").

Pour l'instant nous avons modifié tous les pixels de l'image. Avec l'instruction "if", il est possible de modifier seulement certains pixels.

Travail :

Saisissez et testez le programme suivant :

```
from PIL import Image
img = Image.open("pomme.jpg")
largeur_image=500
hauteur_image=500
for y in range(hauteur_image):
    for x in range(largeur_image):
        r,v,b=img.getpixel((x,y))
        if b<200:
            n_b=255-b
        else :
            n_b=b
        img.putpixel((x,y),(r,v,n_b))
img.show()
```

Expliquez en quelques mots ce que fait ce programme.

Il est même possible de combiner plusieurs conditions :

Saisissez et testez le programme suivant :

```
from PIL import Image
img = Image.open("pomme.jpg")
largeur_image=500
hauteur_image=500
for y in range(hauteur_image):
    for x in range(largeur_image):
        r,v,b=img.getpixel((x,y))
        if v>100 and y>250:
            n_v=0
        else :
            n_v=255
        img.putpixel((x,y),(r,n_v,b))
img.show()
```

Expliquez en quelques mots ce que fait ce programme.