

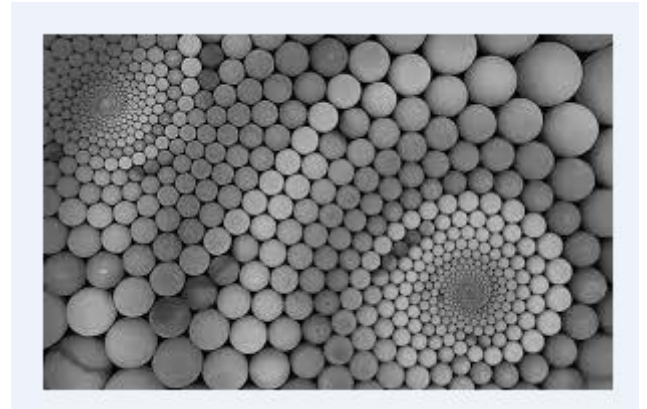
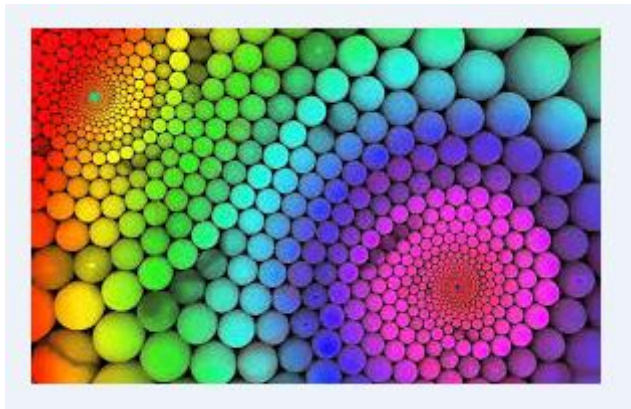
## Image couleur /niveaux de gris

### Compétences visées

Un des objectifs de l'enseignement de SNT est de développer et de coder des scripts PYTHON afin d'apporter une réponse à une problématique précise. A travers le thème " la photographie numérique", nous pouvons notamment travailler les compétences suivantes dans l'activité proposée :

- Traiter algorithmiquement une image pour la transformer en agissant sur les trois composantes de ses pixels.
- Écrire et développer des algorithmes pour résoudre une problématique.
- Connaître le système RGB et la structure d'une image numérique.

### Situation déclenchante



Des algorithmes permettent aujourd'hui de traiter toutes les lumières, d'effectuer une retouche facile, avec une excellente qualité. Avec l'arrivée du téléphone mobile, des algorithmes de retouche d'images se multiplient notamment à travers les filtres. Comment ces algorithmes fonctionnent-ils ?

### Problématique

Comment programmer un algorithme qui permette de transformer une image couleur en une image avec différents niveaux de gris ?

## Fiche méthode

### Déroulement possible du projet

Travail de groupe possible. En amont de ce projet, il serait intéressant que les élèves présentent plusieurs exposés :

- Un exposé sur la structure d'une image numérique.
- Un exposé sur le système RGB.
- Un exposé sur la méthode de conversion d'une image couleur en une image en niveaux de gris par moyenne des pixels RGB.

Ces exposés permettront d'acquérir les bases nécessaires pour la réalisation de ce projet.

Il faut ensuite choisir une image avec un nombre maximal de pixels de  $320 \times 190$ , parmi les fichiers exemples proposés (IMAGE1:  $9 \times 5$  pixels ; IMAGE2 :  $284 \times 177$  pixels ; GO :  $221 \times 190$  pixels).

### Proposition de résolution

Bibliothèque nécessaire (voir paragraphe importation de bibliothèque ci-après)

Les 2 paramètres de la fonction correspondent aux nombres de pixels (largeur et hauteur) de la photo utilisée.

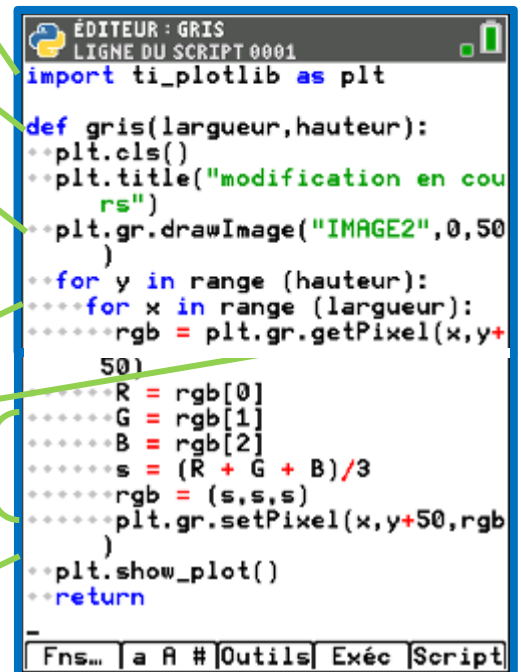
L'instruction `plt.gr.drawImage("IMAGE2",0,50)` permet l'affichage de l'image nommée IMAGE2 avec le coin gauche de l'image au niveau du pixel (0,50) sur l'écran de la calculatrice. Pour rappel, l'écran de la calculatrice possède  $320 \times 240$  pixels. Instruction à taper lettre par lettre (instruction absente des menus pour l'instant).

La double boucle `for` permet de passer en revue tous les pixels de l'image.

L'instruction `plt.gr.getPixel(x,y+50)` permet de récupérer le code RGB du pixel en position (x,y+50) sur l'écran de la calculatrice.

Instructions qui permettent la conversion en niveaux de gris.

L'instruction `plt.gr.setPixel(x,y+50,rbg)` permet l'affichage du pixel en position (x,y+50) sur l'écran de la calculatrice avec la couleur `rbg` obtenue.



```
ÉDITEUR : GRIS
LIGNE DU SCRIPT 0001
import ti_plotlib as plt

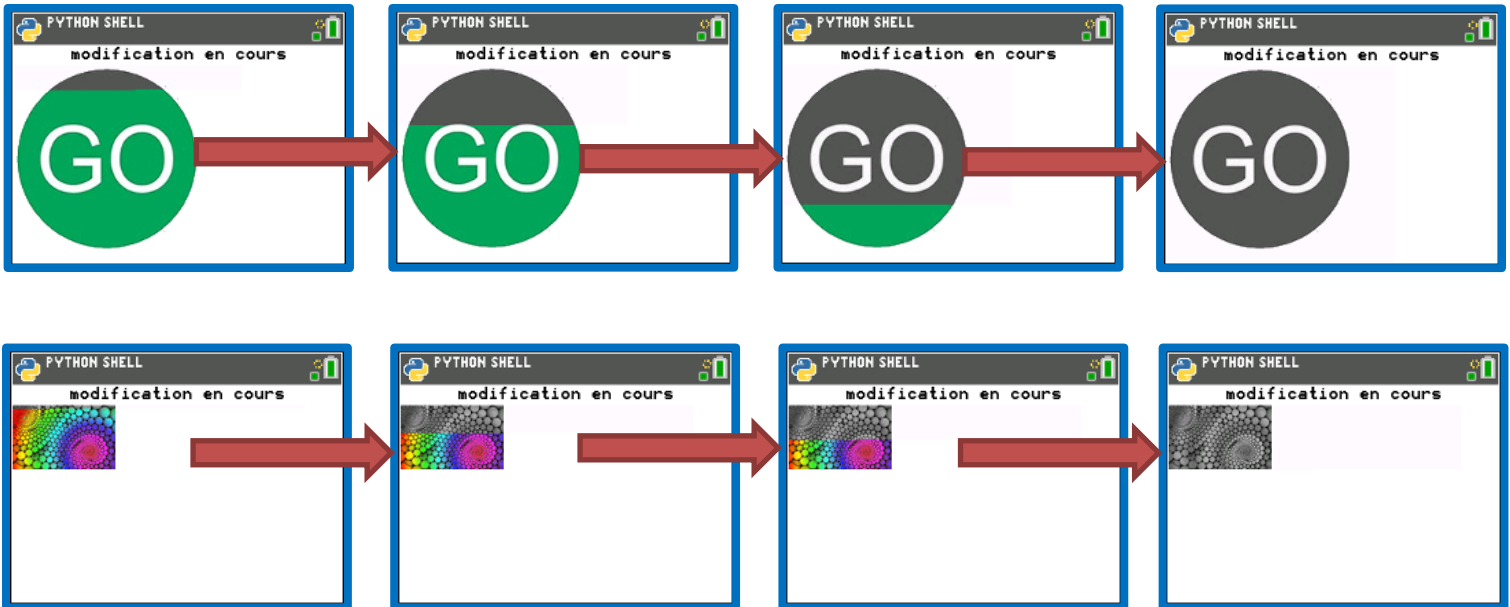
def gris(largeur,hauteur):
    plt.cls()
    plt.title("modification en cou
rs")
    plt.gr.drawImage("IMAGE2",0,50
)
    for y in range (hauteur):
        for x in range (largeur):
            rbg = plt.gr.getPixel(x,y
+
50)
            R = rbg[0]
            G = rbg[1]
            B = rbg[2]
            s = (R + G + B)/3
            rbg = (s,s,s)
            plt.gr.setPixel(x,y+50,rbg
)
    plt.show_plot()
    return
```

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



## Fiche méthode

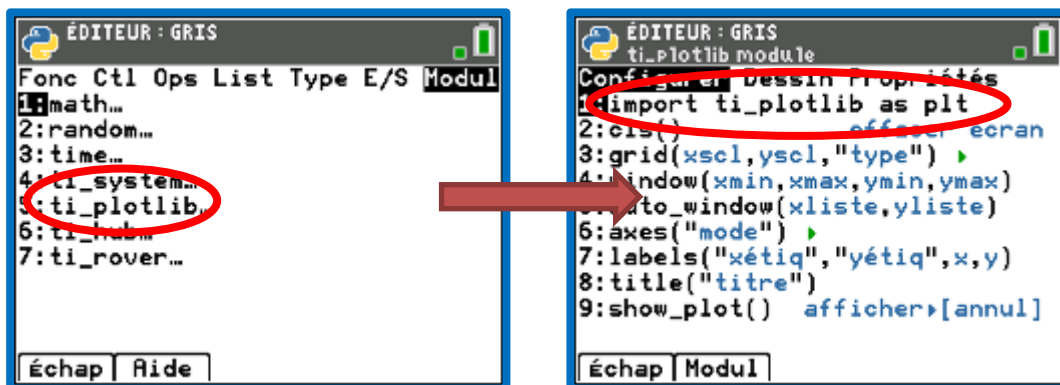
Voici des exemples de résultat du script : on observe la modification de l'image ligne par ligne de pixels (on visualise ainsi le fonctionnement de la double boucle *for*).



Le temps de traitement du programme peut parfois être assez long en fonction du nombre de pixels de l'image. Pour une exécution rapide on pourra utiliser le fichier IMAGE1 (9 x 5 pixels).

## Utilisation des bibliothèques

Une fois le nouveau script créé, pour importer la bibliothèque *ti\_plotlib*, appuyer sur la touche  $f(x)$ , sélectionner la rubrique Modul, puis suivre les copies d'écran suivantes :



Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus

