

# Exercices : Photographie numérique

---

## Exercice 1

Pour utiliser une image dans un programme python on peut utiliser la bibliothèque PIL (Python Imaging Library). Cette bibliothèque permet de lire et d'écrire des images dans de nombreux formats.

Pour installer la bibliothèque PIL, il faut taper la commande suivante dans un terminal :

```
pip install pillow
```

Dans Thonny, il faut aller dans le menu **Outils** puis **Gérer les paquets** et taper **pillow** dans la zone de recherche.

Voici un exemple d'utilisation de la bibliothèque PIL pour lire une image :

```
from PIL import Image

# Charger l'image à partir du fichier
image = Image.open('Exercice1.jpg')

# Afficher les dimensions de l'image
print(image.size)

# Afficher le premier pixel de l'image
print(image.getpixel((0, 0)))

# Modifier le premier pixel de l'image
image.putpixel((0, 0), (255, 0, 0))

# Sauvegarder l'image modifiée dans un nouveau fichier
image.save('Exercice1_resultat.jpg')
```

Ce programme est disponible dans le fichier [ExempleUtilisationPil.py](#)

Ecrire un programme pour récupérer la couleur du pixel en position (2, 2) de l'image **Exercice1.jpg**

---

## Exercice 2

Pour modifier l'image on va y appliquer différents filtres. Chaque filtre va modifier la couleur d'un pixel et on appliquera le filtre à tous les pixels de l'image.

Le code suivant est une base pour appliquer un filtre qui retire la composante rouge de l'image :

```
## Importation des modules

from PIL import Image

## Déclaration des constantes

baseImage = Image.open('Exercice1.jpg')

## Déclaration des fonctions

def filtreSansRouge(pixel):
    """
    Retourne un pixel avec la composante rouge à 0
    @param pixel: le pixel à modifier
    @return: la valeur du pixel modifié
    """
    return (0, pixel[1], pixel[2])

def sauvegarderImage(image, nomFichier):
    """Sauvegarde l'image dans un fichier"""
    image.save(nomFichier)

def appliquerFiltre(image, filtre):
    """Applique un filtre à une image"""
    for x in range(image.width):
        for y in range(image.height):
            image.putpixel((x,y), filtre(image.getpixel((x,y))))

## Fonction principale

def main():
    """Fonction principale"""
    appliquerFiltre(baseImage, filtreSansRouge)
    baseImage.show()
    sauvegarderImage(baseImage, 'Exercice1_rouge.jpg')

## Programme principal

if __name__ == '__main__':
    main()
```

Ce programme est disponible dans le fichier [BaseFiltre.py](#)

Testez ce programme. Vous devez obtenir une image avec la composante rouge retirée.

---

## Exercice 3

Créez un nouveau filtre qui retire la composante verte de l'image et appliquez-le à l'image [Exercice1.jpg](#).

---

## Exercice 4

Créez un nouveau filtre qui ne conserve que la composante rouge de l'image et appliquez-le à l'image

`Exercice1.jpg`.

---

## Exercice 5

Pour convertir une image en noir et blanc, il faut remplacer la valeur de chaque pixel par la moyenne des valeurs RVB du pixel.

Par exemple, le pixel (255, 0, 0) devient (85, 85, 85) car  $(255 + 0 + 0) / 3 = 85$

Créez un nouveau filtre qui convertit l'image en noir et blanc et appliquez-le à l'image `Exercice1.jpg`.

---

## Exercice 6

```
# Pour créer une nouvelle image :  
image2 = Image.new('RGB', (100, 100), (255, 255, 255))  
  
image2.save('Exercice3.jpg')
```

Ecrire un programme pour créer une image de 100x100 pixels que vous remplirez avec des pixels de couleur aléatoire.

Vous pourrez ensuite utiliser vos filtres précédents pour modifier cette image.

---