

Premiers pas dans le traitement d'images

- Objectifs**
- Utiliser des boucles imbriquées pour parcourir les pixels d'une image.
 - Modifier une image, gérer les couleurs de pixels.

I Première étude

Dans cette partie, nous travaillerons sur l'image ci-contre :



tux_yoda.jpg

```
1 import matplotlib.pyplot as plt
2 from PIL import Image
3
4 MonImage = Image.open("tux_yoda.jpg")
5
6 print(MonImage.size)
7
8 print(MonImage.getpixel((45, 40)))
9
10 MonImage.putpixel((10, 10), (0, 255, 255))
11
12 plt.imshow(MonImage)
13 plt.show()
```

1. Quelle est la taille de l'image chargée dans le code précédent?
2. Quelle est la couleur du pixel de coordonnées (45; 40)?
3. Trouver les coordonnées d'un pixel de couleur jaune :
4. Modifier l'image pour que le pixel de coordonnées (20;20) soit bleu.
5. Mettre un pixel de couleur rouge au bout de du sabre laser.
6. Que fait le code ci-dessous?

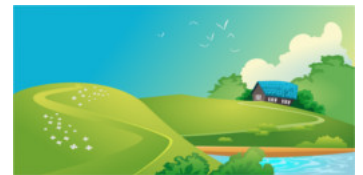
```
1 import matplotlib.pyplot as plt
2 from PIL import Image
3
4 MonImage = Image.open("tux_yoda.jpg")
5
6 Taille = MonImage.size
7
8 for x in range(Taille[0]):
9     for y in range(Taille[1]):
10         rgb = MonImage.getpixel((x, y))
11         R = rgb[0]
12         G = rgb[1]
13         B = rgb[2]
14         MonImage.putpixel((x, y), (R, R, R))
15
16 plt.imshow(MonImage)
17 plt.show()
```

7. Retirer une autre couleur ou inverser des couleurs à l'aide du code précédent.

II Niveaux de gris

1. Comment définir un *gris* en RGB?
2. Que deviendrait un pixel de composante RGB : (14, 212, 177) lors de cette transformation?
.....
.....
3. Compléter le programme suivant :

```
1 import matplotlib.pyplot as plt
2 from PIL import Image
3
4 MonImage = Image.open("paysage.jpg")
5
6 Taille = MonImage.size
7
8 for x in range(Taille[0]):
9     for y in range(Taille[1]):
10        rgb = MonImage.getpixel((x, y))
11        R = rgb[0]
12        G = rgb[1]
13        B = rgb[2]
14        gris = .....
15        MonImage.putpixel((x, y), (gris, gris, gris))
16
17 plt.imshow(MonImage)
18 plt.show()
```



paysage.jpg

III Négatif d'une image



photo_NB.jpg



photo_NB_negative.jpg

1. À votre avis, que s'est-il passé sur chaque pixel de l'image lors de cette transformation?
.....
.....
2. Que deviendrait un pixel de composante RGB (10, 10, 10) lors de cette transformation?
.....

3. Charger l'image ci-contre et créer son image négative :



photo2_NB.jpg

IV Message caché

L'image ci-dessous contient un message caché. Quel est ce message?



V Fond Vert

message.png

Placer les deux personnages sur le fond :



cinema.png

Code départ :

```
1 import matplotlib.pyplot as plt
2 from PIL import Image
3
4 MonImage = Image.open("cinema.png")
5 MonFond = Image.open("fond.png")
6
7 Taille = MonFond.size
8
9 for x in range(Taille[0]):
10     for y in range(Taille[1]):
11         rgb = MonFond.getpixel((x, y))
12         R = rgb[0]
13         G = rgb[1]
14         B = rgb[2]
15         A = 255
16         MonFond.putpixel((x, y), (R, G, B, A))
17
18 Taille2 = MonImage.size
19
20 for x in range(Taille2[0]):
21     for y in range(Taille2[1]):
22         rgb = MonImage.getpixel((x, y))
23         R = rgb[0]
24         G = rgb[1]
25         B = rgb[2]
26         A = 255
27         MonFond.putpixel((x, y), (R, G, B, A))
28
29 plt.imshow(MonFond)
30 plt.show()
```