



2nde SNT

La photo numérique

Activité 3

Traitement d'une image en python

Dans cette séance, nous allons insérer une image dans un programme agir sur ses pixels.

Traitement d'une image : changer la couleur des pixels

Dans cette séance, nous allons insérer une image dans un programme agir sur ses pixels.

- Télécharger les ressources de l'activité dans votre dossier de travail. Dans ce dossier, vous trouverez une image « *bateau.jpg* ». Copiez ce fichier dans votre dossier de travail (celui où vous enregistrerez vos programmes Python).
- Ouvrez le logiciel éditeur pour Python. (thony ou edupython...)

```
1 # --on importe le module Image de la bibliothèque PIL
2 from PIL import Image
3
4 # --on ouvre le fichier image
5 img = Image.open('bateau.jpg')
```

- On commence par charger le module Image de la bibliothèque PIL. (ou pillow) Ce module permet de lire, de manipuler et d'enregistrer des images de différents format, et de les rendre accessible au programme.
- On ouvre ensuite une image à partir d'un fichier. Ici, le fichier est « *bateau.jpg* ». L'image est lue par la méthode `open()` du module Image, et est stockée dans la variable `img`. C'est cette variable que nous utiliserons dans la suite du programme.

```
7 # --on récupère et on affiche la largeur et hauteur de l'image
8 colonne, ligne = img.size
9 print(colonne)
10 print(ligne)
```

- On récupère ensuite la largeur et la hauteur de cette image, et on affiche ces valeurs (en nombre de pixels).
- Pour afficher l'image, insérer à la fin du programme l'instruction `:img.show()`

1. Récupérer la couleur d'un pixel

Nous allons maintenant récupérer la couleur d'un pixel en utilisant la méthode `getpixel()`. Par exemple, pour récupérer la couleur du pixel de coordonnées (200, 150) et l'afficher, il suffit de saisir :

```
12 # --on récupère la couleur d'un pixel
13 pixel = img.getpixel((200, 150))
14 print(pixel)
```

Attention aux doubles parenthèses !

La dernière commande affiche : (212, 218, 234). Ces 3 nombres correspondent aux valeurs des 3 composantes R-V-B.

Testez la méthode `getpixel()` sur plusieurs pixels différents et observer les valeurs des couleurs affichées.

Vous pouvez utiliser un afficheur de couleur RGB (Paint, ou Gimp, ou en ligne <http://www.proftnj.com/RGB3.htm>) pour comparez les résultats et retrouver la couleur d'origine.

A compléter : pixel (200,150) couleur : pixel (250,100) couleur : pixel (200,200) couleur :

2. Modifier la couleur d'un pixel

La fonction `putpixel()` permet de modifier la couleur d'un pixel. Par exemple, pour mettre le pixel de coordonnées (200, 150) en rouge, il suffit de saisir :

```
16 # --on change la couleur d'un pixel
17 img.putpixel((200, 150), (255, 0, 0))
```

Remarque : L'image « *bateau.jpg* » n'est pas directement modifiée, uniquement la variable image `img` le temps du programme.

3. Modifier la couleur d'une ligne

En utilisant une boucle *for*, nous allons maintenant changer la couleur d'une ligne entière.
Complétez le code suivant pour passer la ligne centrale en vert.
Attention à l'indentation dans la boucle *for*.

```
# --on change la couleur des pixels d'une ligne
for i in range(...):
    img.putpixel(..., (... , ... , ...))
```

Vous pouvez tester le programme avec d'autres lignes et d'autres couleurs.

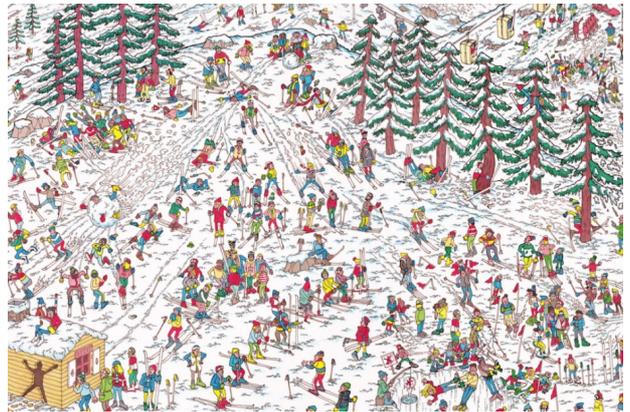
4. Modifier la couleur d'une colonne

En utilisant des boucles *for*, changez la couleur des pixels de la colonne centrale, puis la couleur d'autres colonnes.

Application

Ouvrir l'image du fichier « charlie.jpg ».
Dans l'image, repérez Charlie et encadrez-le en noir :
On pourra repérer les coordonnées de charlie en ouvrant l'image dans Gimp

- En utilisant plusieurs boucles *for* (4 boucles)
- En utilisant une seule boucle *for*.



5. Modifier la couleur d'une zone rectangulaire :

En utilisant deux boucles *for* imbriquées, nous allons maintenant modifier la couleur des pixels d'une zone rectangulaire pour obtenir le résultat ci-dessous :



Le rectangle, doit être jaune, avoir une largeur de 200 pixels et une hauteur de 50 pixels. Il doit également être centré en largeur et en hauteur sur l'image. Complétez le code ci-dessous pour obtenir le résultat voulu :

```
# --on change la couleur des pixels d'un rectangle
for i in range(..., ...):
    for j in range(..., ...):
        img.putpixel(..., (... , ... , ...))
```

Vous pouvez essayer avec d'autres rectangles, avec d'autres couleurs, à différents endroits, pour bien comprendre le fonctionnement des deux boucles *for* imbriquées.

Application

Ouvrir l'image du fichier « voiture1.jpg ».
Dans l'image, masquez la plaque d'immatriculation de la voiture avec un rectangle noir, en utilisant deux boucles *for* imbriquées.

