Projet 4 : traitement de l'image avec PIL

1) Modalités

- Ce miniprojet est à faire par groupe de 2 ;
- Ce projet est à faire sur Jupyter ;
- Ce projet est à rendre sur l'ENT.

2) Définitions et exercices

Exercice 1 :

Rechercher sur internet la définition de :

1. Définition d'une image ;
2. Résolution d'une image ;
3. La profondeur de couleur.

Exercice 2 :

Soit une image de définition 5000 × 6000 pixels que l'on imprime sur du papier photo de taille 19,8 cm × 29,7 cm. Calculer la résolution de cette image en pixels par cm puis en pixels par pouce (ppp).

Exercice 3 :

Sachant que l'on estime que pour avoir une impression de qualité il faut atteindre une résolution de 300 ppp, calculer la définition minimale d'une image dans le cas d'une impression sur du papier photo de dimensions 10 cm × 15 cm.

3) Traitement de l'image avec Python

3.1) Ouvrir une image

La bibliothèque PIL (Python Imaging Librairy) permet de traiter des images avec Python. Elle est conçue de manière à offrir un accès rapide aux données contenues dans une image.

Pour charger et afficher une image avec PIL, il suffit de taper les lignes suivantes :

```
1 from PIL import Image,ImageDraw
2
3 img1=Image.open("iguane.jpg")
4 img1.show()
```

L'image nommée `iguane.jpg` doit être dans le même dossier que le script python. L'instruction de la ligne 3 permet de charger l'image est de l'affecter dans une variable de type Image nommée `img1`. La ligne 4 se charge d'afficher l'image `img1`.

A faire : tester mes lignes de code précédentes avec une image que vous aurez trouvé sur internet.

Pour affecter les dimensions de l'image `img1` à deux variables `longueur` et `largeur`, on utilise les lignes de code suivantes :

```
1 largeur=img1.width
2 longueur = img1.height
```

Exercice 4 :

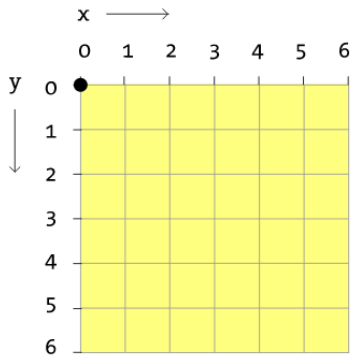
Déterminer un programme qui calcule la définition d'une image

Remarque : sur Jupyter, l'affichage d'une image nécessite d'adapter le code précédent. Il devient donc :

```
1 from PIL import Image,ImageDraw
2 from IPython.display import display
3
4 img1=Image.open("iguane.jpg")
5 display(img1)
```

3.2) Repérage et obtention de la couleur d'un pixel

Dans un premier temps, il est nécessaire de connaître les coordonnées du pixel dans l'image. Afin de repérer un pixel dans une image, Python utilise un système d'axe dont l'origine est le coin supérieur gauche de l'image. L'axe des abscisses est orienté vers la droite mais attention, l'axe des ordonnées est orienté vers le bas.



Afin de déterminer les composantes RVB d'un pixel situé en un point de coordonnées (x,y), on utilise la méthode `getpixel((x,y))`. Exemple :

```
1 x=5
2 y=10
3 couleur=img1.getpixel((x,y)) # couleur est un tuple
4 R,V,B=couleur
5 print("Les composantes R,V,B du pixel (x={},y={}) sont R={},V={} et B={}".format(x,y,R,V,B))
```

A faire : tester le code précédent avec plusieurs couples de valeurs (x,y) et observer les composantes RVB.

3.3) Création d'une variable Image et affectation de la couleur d'un pixels

Pour créer une variable de type Image vide de dimensions 400×400 et en mode RGB, il faut taper les lignes suivantes :

```
from PIL import Image

img2=Image.new("RGB",(400,400))
```

Remarque : il existe différents modes pour une image (voir lien).

Pour placer sur l'image vide `img2` un pixel aux coordonnées (x,y) que l'on veut avec un couleur de composante R,V et B, on utilise la méthode `img2.putpixel((x,y),(R,V,B))` :

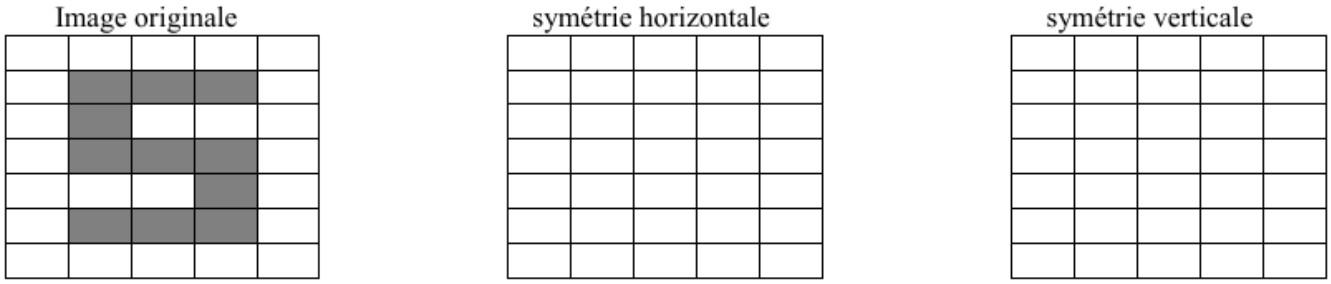
Exemple :

```
img2.putpixel((200,200),(255,255,255))
display(img2)
```

A faire : placer plusieurs points colorés sur une image vide

3.4) Exercices

Exercice 5 : Symétries



Imaginer des algorithmes que vous mettrez en application sur Python avec l'image de votre choix qui permettront de réaliser les deux transformations ci-dessus. Il est conseillé de créer une deuxième variable de type `Image` qui contiendra l'image transformée.

Exercice 6 : Transformation en niveaux de gris

Une image en niveau de gris est une image dont les 3 composantes R,V,B sont identiques.

Exemple : Si les composantes RVB, d'un pixel sont (R=125, V=100 et B=130). Le niveau de gris sera alors G=125+100+130/3 Nous pouvons attribuer à ce pixel la couleur suivante : (G,G,G)

Écrire le script et lancer le programme.

Exercice 7 : Négatif d'une image

Afin de réaliser le négatif d'une image, on affecte à chaque pixel de l'image de couleur (R,G,B) un pixel de couleur (255-R, 255-V, 255-B). Écrire le script et lancer le programme.