

## TP 1

Il existe plusieurs modules permettant de manipuler une image sous Python. Dans le cadre de ce premier TP, on va utiliser les méthodes de traitement d'images définies dans la classe **Image** du module **PIL**. Le module PIL (Python Image Library) permet d'ouvrir/créer des fichiers images et de les manipuler (accès à la valeur d'un pixel, modification d'un pixel, etc.). Le tableau présenté ci-dessous présente des instructions utiles pour réaliser le présent TP.

Instruction	Rôle
<code>img=Image.open('l1ll.jpg')</code>	Ouvrir un fichier image et créer un objet Image
<code>img.show()</code>	Afficher une image dans une fenêtre
<code>Data=img.getdata()</code> <code>L=list(Data)</code>	Retourner une séquence des valeurs des pixels dans une variable Convertir la séquence des pixels en liste
<code>Largeur, hauteur = img.size</code> <code>print(Largeur, hauteur)</code>	Retourner la taille de l'image, en pixels. La taille est donnée sous la forme d'un tuple à 2 entiers (largeur, hauteur).
<code>img.width</code> <code>img.height</code>	Retourner largeur de l'image en pixels. Retourner la hauteur de l'image, en pixels.
<code>img_gris = img.convert('L')</code>	Retourner une copie d'une image convertie dans un mode différent. Les modes d'images en Python : ✓ « 1 »: Mode 1 bit ou mode Noir et Blanc. ✓ « L »: Mode 8 bits ou mode Niveaux de gris. ✓ « RGB »: Mode 24 bits ou mode RVB.
<code>Mode=img.mode</code>	Retourner le mode d'image. C'est une chaîne spécifiant le format de pixel utilisé par l'image. Les valeurs typiques sont : "1", "L", "RGB"
<code>p = img.getpixel((0,1))</code> <code>(r, g, b) = img.getpixel((20, 35))</code>	Récupérer la valeur d'un pixel à une position donnée.
<code>img.putpixel((0,1),255)</code>	Modifier la valeur d'un pixel à une position donnée.
<code>img1= img.copy ()</code>	Créer une copie conforme d'une image.
<code>img2=img1.resize((120,120))</code> <code>img1.resize((120,120)).show()</code>	Créer une copie redimensionnée d'une image. Afficher la copie redimensionnée d'une image.
<code>img.transpose(FLIP_LEFT_RIGHT).show()</code>	Créer une copie transposée d'une image.
<code>Img_gris.rotate(45).show()</code>	Créer une copie de l'image originale déviée d'un angle de 45 degrés à partir de son centre dans le sens contraire des aiguilles d'une montre.

Ecrire un script Python qui permet de :

1. Ecrire une fonction `AfficherImg(NI)` qui permet d'afficher l'image dont le nom+extension sont passés comme paramètres. La fonction doit retourner le nom de la variable utilisée pour sauvegarder l'image.
2. Écrire une fonction `convertir_NB(img)` qui convertit l'image passée en paramètre en une image en niveau de gris.
3. Ecrire une fonction `caracter_img(imgG)` qui retourne: le nombre de pixels de l'image en niveau de gris passée comme paramètre, le nombre de pixels parfaitement noirs et le nombre de pixels parfaitement blancs.
4. Ecrire une fonction qui retourne les deux images `img2` et `img3`. L'image `img2` est une copie redimensionnée de l'image `img`. L'image `img3` est une copie de `img` après une rotation de 45°.
5. Ecrire une fonction qui permet de transformer les bordures (dix pixels de largeur) de votre image en noir.

Votre script doit afficher : le nombre de pixels de l'image, le nombre de pixels parfaitement noirs et le nombre de pixels parfaitement blancs/ l'image en niveau de gris, l'image `img2` et l'image `img3`/ le résultat de la question 5.

**Bon courage**