

TP N° 07 - Module R107

Le but de ce TP est d'étudier la fabrication et l'analyse d'images en python.

1. Introduction sur les images :

Le module PIL (Python Image Library) permet de manipuler des images.

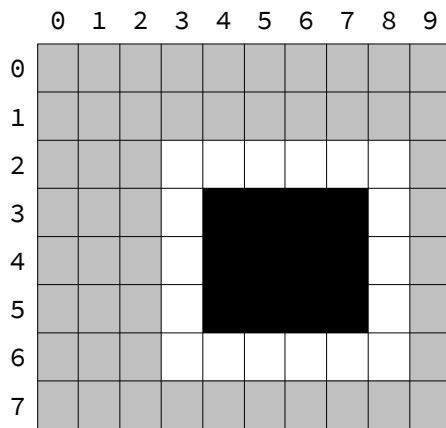
1. Pour créer une image grise de 10x8 pixels dans le fichier `img.png`, on peut écrire :

```
from PIL import Image
img=Image.new('RGB',[10,8],color='silver')
img.save('img.png')
```

2. On peut ensuite dessiner un rectangle noir dont le point en haut à gauche est en position 3,2 et le point en bas à droite en 8,6 avec la méthode `ImageDraw.rectangle(xy,fill=None,outline=None,width=1)` :

```
from PIL import Image,ImageDraw
img=Image.new('RGB',[10,8],color='silver')
imgDraw=ImageDraw.Draw(img) #Objet permettant de dessiner dans l'image
imgDraw.rectangle([3,2,8,6],fill='black',outline='white')
img.save('img.png')
```

On doit obtenir l'image suivante :



3. On peut récupérer la taille de l'image et la liste des pixels de la façon suivante :

```
w,h=img.size
print(f'Taille: {w}x{h} pixels')
pixels=img.load()
for y in range(h):
    print('*** Ligne',y,'***')
    for x in range(w):
        print(pixels[x,y])
```

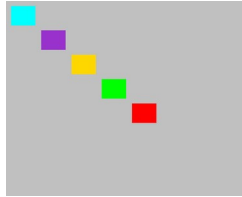
NB : Pour ouvrir une image déjà existante on peut faire : `img=Image.open('xxx.png')`

4. Vérifier qu'un pixel est bien composé de 3 valeurs (de 0 à 255) représentant respectivement les couleurs Rouge, Vert et Bleu. Comment sont codés le gris, le noir et le blanc ? Vérifier le schéma du §1.2.
5. Quelles sont, en pixels, la largeur et la hauteur du rectangle (blanc) ? Faire le lien avec les valeurs données pour la création du rectangle.
6. Créer alors la fonction **drawRectangle**(`imgDraw,x0,y0,w,h,fill=None,outline=None`) qui permette de faciliter le dessin d'un rectangle dans une image à partir des coordonnées du point en haut à gauche, de la largeur et de la hauteur.
NB : La fonction ne fait que dessiner un rectangle dans l'image fournie et ne comporte qu'une seule ligne !
7. Ajouter un second rectangle rouge sans contours (couleur 'red') de largeur 3 et de hauteur 2 pixels en position 4,3.
8. On veut maintenant détecter uniquement les « niveaux de gris » de l'image en réalisant le calcul de luminance de la norme IUT-R-601 : $Y=0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$. Créer une fonction qui réalise ce calcul et qui affiche la valeur de Y pour chaque pixel.
9. Comment trouver les pixels blancs ou noirs uniquement (et pas ceux qui sont gris ou rouge) ?

2. Fabrication d'images :

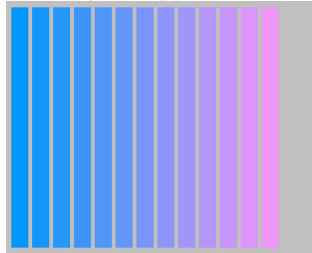
Utiliser la fonction `drawRectangle()`, que l'on aura défini dans le module `imgUtils`, pour fabriquer les images suivantes. Voir l'outil « Sélecteur de couleurs » de Geany et <https://pillow.readthedocs.io/en/stable/reference/ImageColor.html#color-names>.

1. Avec une boucle, à partir des couleurs suivantes : 'Aqua', 'DarkOrchid', 'Gold', 'Lime', 'Red'



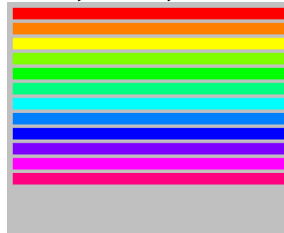
2. En faisant varier la partie « rouge » de 0 à 255 par pas de 20 (remplacer `red` par un nombre dans l'exemple ci-dessous) des couleurs avec la fonction `rgb()` du module `ImageColor` :

```
color=ImageColor.getrgb('rgb(red,150,250)')
```



3. En faisant varier la teinte des couleurs de 0 à 360 par pas de 30 (remplacer `hue` par un nombre dans l'exemple ci-dessous) avec la fonction `hsv()` du module `ImageColor` :

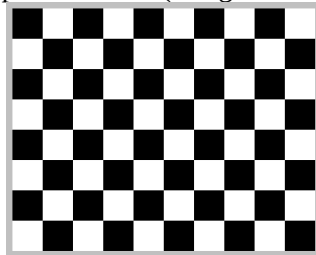
```
color=ImageColor.getrgb('hsv(hue,100%,100%)')
```



NB : Il se peut que le format `hsv()` ne fonctionne pas suivant les versions de la librairie installée, utiliser alors le format `hsl()` en indiquant 50% sur la dernière valeur :

```
color=ImageColor.getrgb('hsl(hue,100%,50%)')
```

4. Faire un damier avec le nombre de cases paramétrable (m lignes et n colonnes).



5. Tester enfin la fonction `ImageDraw.text(xy, text, fill=None, font=None, anchor=None, spacing=4, align='left', direction=None, features=None, language=None, stroke_width=0, stroke_fill=None, embedded_color=False)` avec la police `DejaVuSans.ttf` par exemple et une couleur au format `#RRGGBB...` Voir <https://pillow.readthedocs.io/en/stable/reference/ImageDraw.html#PIL.ImageDraw.ImageDraw.text>.

