TD 06

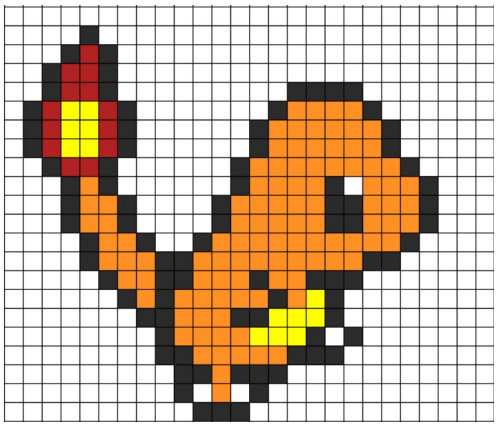
Manipulation d'images

Lycée Louis Thuillier - Physique-Chimie - PCSIB

Une image en nuance de gris

Matrice de pixel

En informatique une image "en noir et blanc" est composée de pixels (∼ des petits carrés) de couleur unie : mis les uns à côté des autres ils forment alors un image. Plus une image comporte de pixel, plus elle possède de détail et paraîtra "naturel" à l'oeil.



Pour une photo en noir et blanc, chaque pixel est donc défini par :
- ▷ une position
- ▷ un niveau de gris

En informatique, on représenter une image en niveaux de gris par un tableau bidimensionnel (∼ une liste de listes) dont l'élément d'indice (i, j) est un entier compris entre 0 (noir) et 255 (blanc) qui correspond au niveau de gris du pixel de la $i$ème ligne, $j$ème colonne.
C'est **la matrice de pixels** de l'image.

> ***Exemple 1 :*** *Représenter sur votre feuille l'image dont le tableau est le suivant :*
>
> $$image = [[0, 250, 125], [125, 250, 0]]$$

Manipulation d'image en Python

Pour manipuler des matrices de pixel en Python, on utilise la bibliothèque *matplotlib.pyplot* qu'on nommera *plt*.

```
import matplotlib.pyplot as plt
```

Pour afficher l'image associée à la matrice de pixel *image* on utilise la commande *plt.imshow* :

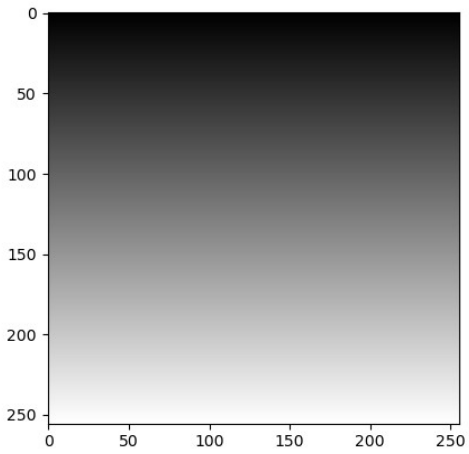
$$plt.imshow(image, cmap = "gray", clim = (0, 255))$$

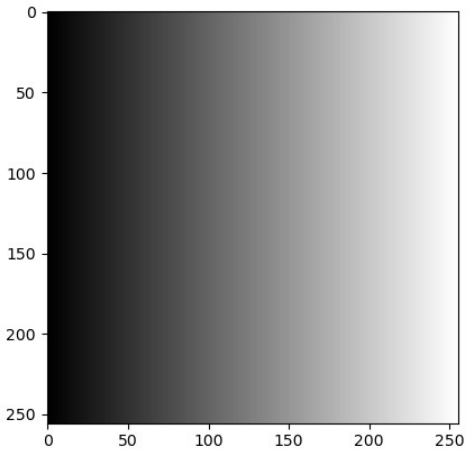
$$plt.show()$$

Application 1 : Afficher avec python l'image de l'exemple 1.

Application 2 :
- ▷ *Créer avec Python un plateau de 50×50 où les colonnes blanches et noirs n'alternent.*
- ▷ *(plus dur) Créer avec Python un plateau de dames : un plateau de 12 sur 12 avec des cases noires et blanches alternées.*

Application 3 : Créer deux dégradés verticaux et horizontaux de 256 pixels sur 256 pixels.





Application 4 : ***Petit Projet***
On veut représenter un nombre $n < 10^9$ par un code barre carré. On va utiliser la représentation suivante :
- ▷ *chaque chiffre de l'écriture décimale de n sera représenté par 10 colonnes*
- ▷ *le nombre de colonne noires, comptée de gauche à droite, sera égal au chiffre en question.*

Écrire une fonction qui prend en entrée un nombre et renvoie le code barre associée.

Image en couleur

N'importe quelle couleur peut être généré avec un mélange de rouge, de vert et de bleu. Pour représenter un image en couleur, on utilise la même représentation qu'une image en nuance de gris sauf que chaque pixel n'est plus défini par un nombre mais par une liste de 3 nombres : le niveau de rouge, de vert et de bleu, chacun allant de 0 à 255.
- ▷ [255, 0, 0] : rouge
- ▷ [0, 255, 0] : vert
- ▷ [0, 0, 255] : bleu
- ▷ [255, 255, 0] : jaune
- ▷ ...
- ▷ [255, 255, 255] : blanc
- ▷ [0, 0, 0] : noir

Attention ! Les couleurs, par convention, sont précisées dans cet ordre!!

Une image est alors une liste de liste de liste. Un élément d'une image *image_couleur* possède 3 indice :

$$image\_couleur[i][j][k]$$

- ▷ i indique le numéro de la ligne
- ▷ j indique le numéro de la colonne
- ▷ $k = 0, 1, 2$ indique le niveau de la couleur rouge, vert,bleu

Application 5 : Que va afficher Python avec ces lignes de codes ?

```
image=[ [[255,0,0] , [0,255,0]] , [[0,0,255] , [0,0,0]] , [[255,255,255]
[0,255,255]] ]
plt.imshow(image)
plt.show()
```

Application 6 : Générer le drapeau de la Pologne et celui de la France.

L'intérêt de Python est qu'il est possible d'importer une image au format "png".

▷ la photo "nom_photo.png" est dans le dossier "C :\Users \Pierre \Bureau"
Astuce : pour trouver le chemin (l'endroit où se trouve un fichier), aller dans le dossier où se trouve le fichier, clique droit sur le fichier, "Propriété". Le chemin est indiqué à "Emplacement".

▷ importer le fichier avec la commande *plt.imread* :

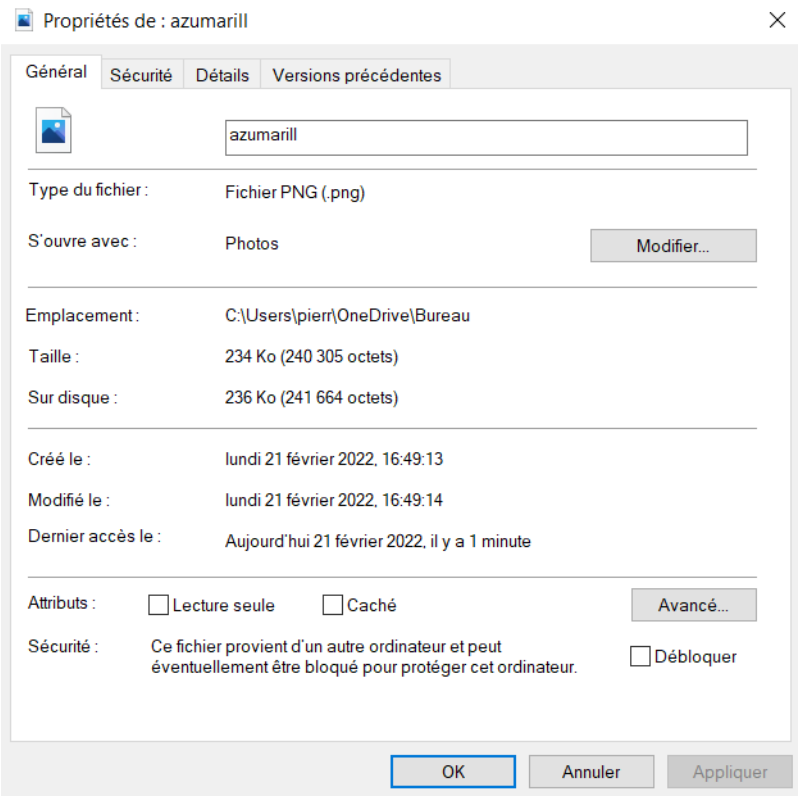
image_importee=plt.imread(C :\\Users\\Pierre\\Bureau\\nom_photo.png)

Attention ! en Python on double les \ !!

▷ Les nuances de couleur sont entre 0 et 1 avec la commande *plt.imread*. Pour les convertir en valeur comprise entre 0 et 255 :

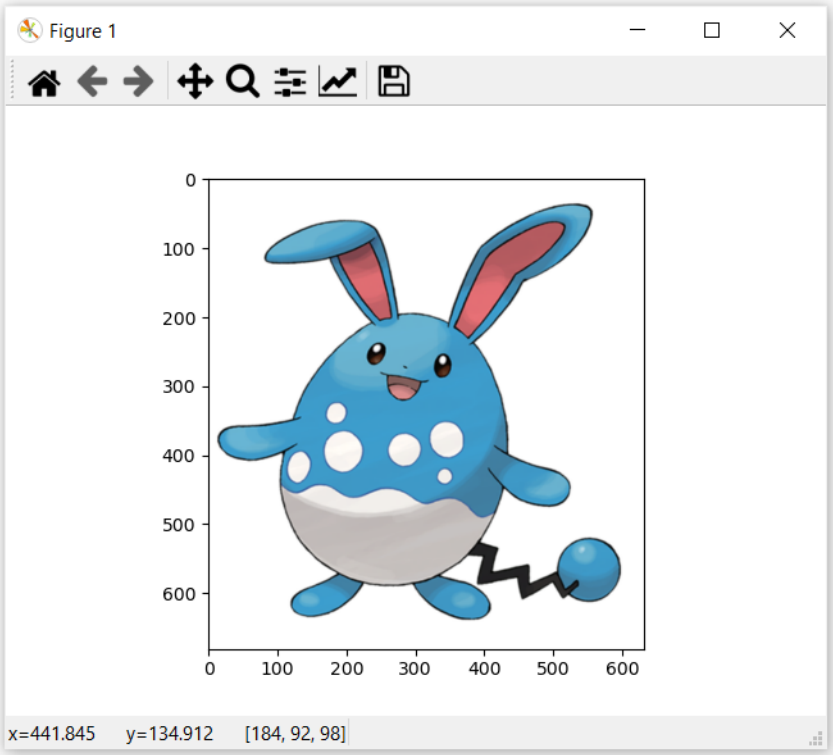
image=[[[int(255*image_importee[i][j][k]) for k in range(3)] for j in range(len(image_importee[0])] for i in range(len(image_importee))]

Exemple 2 :



```
image_importee=plt.imread('c:\\users\\pierr\\onedrive\\bureau\\azumarill.png')
image = [[[int(255*image_importee[i][j][k]) for k in range(3)]for j in
range(len(image_importee[1]))]for i in range(len(image_importee))]

plt.imshow(image)
plt.show()
```



> ***Application 7 :*** *Importer l'image de votre choix sur Python, en nuance de rouge-vert-bleu de 0 à 255.*

Modification d'une image

Si la partie précédente n'a pas été faite, pour toute la suite générer une image à l'aide de l'application 6

> ***Application 8 :*** *Extraire les niveaux (r, g, b) de rouge, de vert et de bleu du pixel :*
> ▷ *en haut à gauche*
> ▷ *en haut à droite*
> ▷ *en bas à gauche*
> ▷ *en bas à droite*

> ***Application 9 :***
> ▷ *Mettre un filtre vert sur la photo : pour cela, générer une image où chaque niveau de rouge et de bleu est divisé par deux.*
> ▷ *Mettre un filtre noir et blanc sur la photo : pour cela générer une image en nuance de gris où la nuance de gris est la moyenne des niveaux de bleu, de rouge et de vert.*

> ***Application 10 :*** *Ecrire une fonction renversement qui prend en entrée une matrice de pixel associée à une image et renvoie une matrice de pixel qui est celle de l'image symétrique de la photo par rapport à son axe vertical.*

> ***Application 11 :*** *Ecrire une fonction negatif qui prend en entrée une matrice de pixel associée à une image et renvoie une matrice de pixel qui est celle du négatif de la photo.*
> *Pour cela, chaque niveau (r, g, b) de rouge-vert-bleu doit être remplacé par $(255 - r, 255 - g, 255 - b)$.*

Application 12 : *Que fait ce programme ? (on ne l'essaiera évidemment pas avant d'avoir la réponse !)*

```
image_mystere=[]
for k in range(3,len(image)-3):
    ligne=[]
    for j in range(3,len(image[0])-3):
        triplet=[]
        for l in range(3):
            couleur=0
            for x in [-3,-2,-1,0,+1,+2,+3]:
                for y in [-3,-2,-1,0,+1,+2,+3]:
                    couleur+=image[k+x][j+y][l]/49
            triplet.append(int(couleur))
        ligne.append(triplet)
    image_mystere.append(ligne)
```