

# T.P. - Python et les images numériques

## S.N.T. - Thème Image Numérique

E. HENRY - 2021 - *Inspiré des T.P. d'I.S.N. de M. Richaud*

### 1 Le codage des images numériques

Dans cette première partie, nous allons examiner le code d'une image numérique.

1. Ouvrir l'image **cerise.pbm** contenue dans le dossier de ressources.

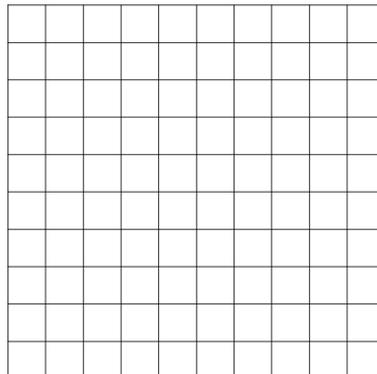
(a) Avec le logiciel GIMP

(b) Avec l'éditeur de texte (Bloc Notes).

Que constatez-vous?

2. A vous de jouer :

(a) En coloriant certaines cases du tableau suivant, créez une image en noir et blanc de  $10 \times 10$  pixels.



(b) Codez-là dans un éditeur de texte (Bloc-notes) en représentant chaque pixel par un 0 ou un 1 (blanc ou noir) à l'aide de l'en-tête suivante :

```
1 P1
2 10 10
3 1
```

Sauvegardez le fichier en le nommant **monimage.pbm** . Ouvrez-le avec GIMP et admirez votre travail.

### 2 Python et les images

Dans cette partie, nous allons voir comment travailler avec une image en couleurs.

#### 2.1 La bibliothèque Pillow

1. Ouvrir le fichier **Monprogramme.py** et remplacer le mot **ADRESSE** par l'adresse du dossier où vous allez travailler.

2. Exécuter le script pour vérifier que l'opération a réussi.

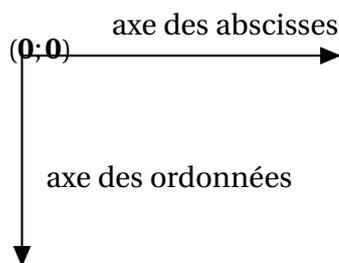
## 2.2 Créer des images

Les instructions suivantes, ajoutées à la suite du script, vous permettent de créer une nouvelle image, que vous pouvez modifier pixel par pixel.

```
1 img=Image.new("L", (10,10)) # La variable img contient une nouvelle image, de taille
  10x10
2 img.show() #Affiche l'image
3 img.save("monimage.png") #Enregistre l'image
4 px=img.getpixel((1,1)) # px contient le pixel de coordonnées (1,1)
5 img.putpixel((1,1),0) # Met le pixel (1,1) à 0
```

### Remarque importante :

Tout point (pixel) peut être repéré par ses coordonnées. Attention le repère utilisé en python n'est pas le repère mathématique usuel.



1. Créer une nouvelle image, l'afficher
2. Modifiez un pixel, affichez-là à nouveau
3. Enregistrez l'image

## 2.3 Importer une image

Pour importer l'image, utilisez à la suite de votre script la commande

```
1 img=Image.open("loup.jpg")
```

A présent, la variable **img** contient l'image à modifier. En saisissant les instructions suivantes, que remarquez-vous?

```
1 print(img.getpixel((1,1)))
```

Niveaux de couleurs : R : .....; V : .....; B : .....

## 2.4 Modifier une image

1. Examinez le programme suivant et répondez aux questions.

```
1 (largeur, longueur)=img.size
2 print(largeur)
3 print(longueur)
4 for i in range(largeur):
5     for j in range(longueur):
6         (R,V,B)=img.getpixel((i,j))
7         img.putpixel((i,j),(0,V,0))
8 img.show()
```

- (a) Que contiennent les variables **longueur** et **largeur** selon vous?
  - (b) A quoi servent les deux boucles **for**?
  - (c) Que modifie l'algorithme sur chaque pixel de l'image? (lignes 6 et 7)
2. *A vous de jouer* : Recopiez le programme qui précède à la suite du script et essayez-le. Puis, modifiez-le pour, au choix : convertir l'image en noir et blanc, passer les couleur en négatif, faire un effet de symétrie...