

Activité 11b : Traitement d'images en couleur

L'objectif de cette activité est de comprendre le codage d'une image couleur et de mettre en œuvre différents traitements. À nouveau, étudions la composition d'une image couleur.

→ **Ouvrir** le fichier `aliceC1.png` et **regarder** l'image obtenue en zoomant sur une zone. **Observer** qu'elle est constituée d'une multitude de *pixels* organisés en lignes (y) et colonnes (x), tout comme le sont les images en niveaux de gris.

Un pixel en niveaux de gris est codé par une valeur comprise entre 0 et 255 (0: noir, 255: blanc).

Dans une image couleur, le pixel est un composé de 3 couleurs primaires : **rouge, vert, bleu**.

Chaque couleur est codée par un nombre entre 0 et 255. Ce codage est nommé **R G B** pour **Red Green Blue**. Pour chacune de ces couleurs, 0 indique la présence de 0% de la couleur et la valeur 255 correspond à 100% de la couleur.

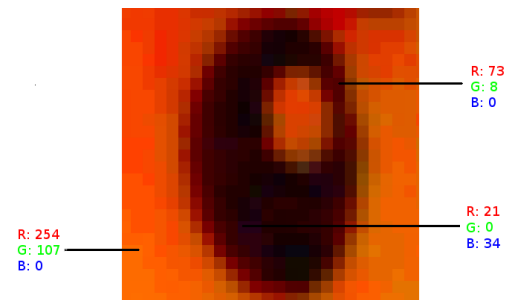


Figure 1: : Zoom sur une image couleur

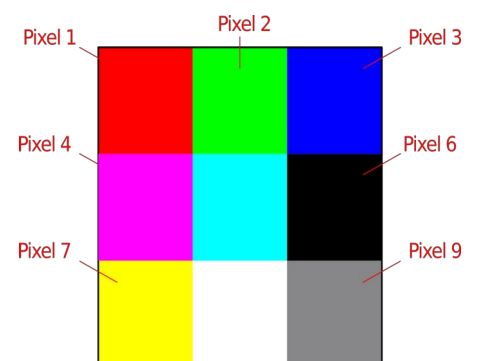
Exemples de codage de couleurs :

Couleur	R	G	B
Noir	0	0	0
Blanc	255	255	255
Rouge	255	0	0
Vert	0	255	0
Bleu	0	0	255
Violet	132	122	191

1. Traitement d'une image couleur sous Python

Code de base pour afficher une image

Le code de base suivant permet de charger dans la variable `img`, la nuance des trois couleurs primaires de chaque pixel de l'image `imageSimple.png`. Cette image ne comporte volontairement que très peu de pixels afin de faciliter la compréhension du traitement automatique d'images.



```
from manipulationImage import *#Importer la bibliothèque traitement
img=ouvrirImage("imageSimpleCouleur.png")#Charger l'image dans img.
    #img->liste contenant les couleurs primaires de chaque
        # pixel
afficherImage(img)                                     #Afficher l'objet
```

→ **Saisir** le programme précédent dans l'IDE Python. **Exécuter** le programme puis **manipuler** dans la console la variable img afin de **compléter** le tableau suivant.

Commande exécutée						
Résultat obtenu						
Action réalisée	Largeur de l'image en nombre de pixels	Teintes du pixel 4	Nuance de la couleur bleu du pixel 6	Teintes du pixel 9	Nombre de composantes d'un pixel	Nuance de couleur rouge du pixel 3

2.Traitements automatiques d'une images réelle

Cette partie a pour objectif de programmer différents traitements d'images numériques.

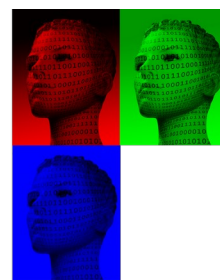
- x **Proposer puis tester** un programme qui affiche le négatif de l'image couleur imageCouleur.png (Traitement 1)
- x **Proposer puis tester** un programme qui affiche au choix de l'utilisateur, la composante rouge, verte ou bleu de l' image imageCouleur.png (Traitement 1).
- x **Proposer puis tester** un programme Python qui effectue une symétrie verticale de l'image imageCouleur.png (Traitement 3).
- x **Proposer puis tester** un programme qui assombri ou éclairci l'image imageCouleur.png en fonction du masque masque.png (traitement 4)



Image d'origine



Traitement 1



Traitement 2



Traitement 3



Traitement 4



3. Pour aller plus loin

- x **Proposer** puis **tester** un programme qui modifie le fond de l'image `aliceC.png` en vert
- x **Proposer** puis **tester** un programme qui fusionne deux images de manière à n'en former plus qu'une, contenant le fondu des deux précédentes. Le principe est assez simple, pour chaque pixel de l'image fusionnée, chaque composante est égale à la composante de valeur la plus élevée des deux pixels des images à fusionner.

