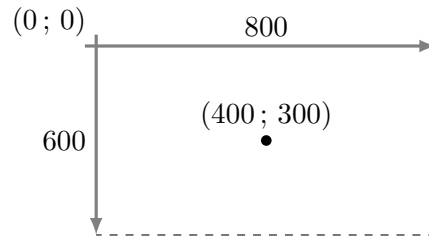




Nous allons utiliser, une nouvelle fois, le langage de programmation Python afin de directement travailler sur les pixels d'une image. Par travailler sur les pixels, on entend déterminer la valeur du canal rouge, la valeur du canal et la valeur du canal bleu pour un pixel donné ou bien encore modifier carrément la couleur d'un pixel. Avant de commencer à écrire un programme qui nous permettra de travailler sur les pixels d'une image, il est nécessaire de préciser que chaque pixel a des coordonnées  $x$  et  $y$  :



Comme vous pouvez le constater sur le schéma ci-dessus, le pixel de coordonnées  $(0; 0)$  se trouve en haut à gauche de l'image. Si l'image fait 800 pixels de large et 600 pixels de haut, le pixel ayant pour coordonnées  $(400; 300)$  sera au milieu de l'image.

Nous allons utiliser une simple photo de pomme pour faire nos essais. L'image de la pomme est téléchargeable sur le site à la date d'aujourd'hui.

Cette image devra se trouver dans le même dossier que vos programmes Python.

**Exercice 1.** Après avoir ouvert l'éditeur Spyder, saisissez et testez le programme suivant :

```
1 from PIL import Image
2 img = Image.open("pomme.jpg")
3 r,v,b=img.getpixel((100,250))
4 print("canal rouge : ",r,"canal vert : ",v,"canal bleu : ",b)
```

Ce programme nous donne le canal rouge, le canal vert et le canal bleu du pixel de coordonnées  $(100, 250)$  de l'image « pomme.jpg ». Voici une analyse ligne par ligne du programme ci-dessus :

- " from PIL import Image " : utilisation de la bibliothèque PIL qui permet de travailler sur les images.
- "img = Image.open(« pomme.jpg »)" c'est grâce à cette ligne que nous précisons que nous allons travailler avec l'image " pomme.jpg ". Pour travailler avec une autre image, il suffit de remplacer "pomme.jpg" par un autre nom. Néanmoins, votre fichier image devra se trouver dans le même dossier que le fichier de votre programme Python.
- "r,v,b=img.getpixel((100,250))" : cette ligne récupère les valeurs du canal rouge (r), du canal vert (v) et du canal bleu (b) du pixel de coordonnées  $(100, 250)$ . Dans la suite du programme, r correspondra à la valeur du canal rouge, v correspondra à la valeur du canal vert et b correspondra à la valeur du canal bleu.
- " print("canal rouge : ",r,"canal vert : ",v,"canal bleu : ",b) " permet d'imprimer le résultat.

**Exercice 2.** Modifiez le programme précédent pour qu'il affiche les valeurs du canal rouge, du canal vert et du canal bleu du pixel de coordonnées  $(250, 300)$  :

.....

Il est possible de modifier les canaux RVB d'un pixel. Pour cela, considérons l'exercice 3 :

**Exercice 3.** Saisissez et testez le programme suivant :

```
1 from PIL import Image
2 img = Image.open("pomme.jpg")
3 img.putpixel((250,250),(255,0,0))
4 img.show()
```

Regardez attentivement le centre de l'image, vous devriez voir un pixel rouge à la place d'un pixel vert. Voici une analyse ligne par ligne du programme ci-dessus :

- "img.putpixel((250,250),(255,0,0))" permet de colorier le pixel de coordonnées (250,250) en rouge (255,0,0).
- "img.show()" permet d'afficher l'image modifiée.

**Exercice 4.** Modifiez le programme précédent afin de colorier le pixel de coordonnées (100, 250) en bleu.

Modifiez un pixel c'est déjà bien mais il est possible de modifier plusieurs pixels. Pour cela, nous allons utiliser des boucles « for ». Le but ici n'est pas de détailler le fonctionnement des boucles « for » en Python ; vous devrez juste, dans un premier temps, comprendre que grâce à ces boucles nous allons pouvoir balayer toute l'image et ne plus nous contenter de modifier les pixels un par un.

**Exercice 5.** Saisissez et testez le programme suivant (ATTENTION : l'exécution de ce programme n'est pas très intéressante en soi, vous pouvez l'arrêter à tout moment en appuyant simultanément sur la touche Ctrl et sur la touche C) :

```
1 from PIL import Image
2 img = Image.open("pomme.jpg")
3 largeur_image=500
4 hauteur_image=500
5 for y in range(hauteur_image):
6     for x in range(largeur_image):
7         r,v,b=img.getpixel((x,y))
8         print("rouge : ",r,"vert : ",v,"bleu : ",b)
9 print("fin")
```

Quelques commentaires sur ce programme :

- Nous commençons par définir les variables "largeur\_image" et "hauteur\_image" ("largeur\_image=500" et "hauteur\_image=500") : l'image "pomme.jpg" fait 500 pixels de large et 500 pixels de haut. Si vous désirez travailler avec une autre image, il faudra veiller à bien modifier la valeur de ces deux variables.
- Les 2 boucles "for" nous permettent de parcourir l'ensemble des pixels de l'image.  
Le plus important ici est de bien comprendre que dans la suite du programme, les variables  $x$  et  $y$  vont nous permettre de parcourir l'ensemble des pixels de l'image : nous allons commencer avec le pixel de coordonnées (0,0), puis le pixel de coordonnées (1,0), puis le pixel de coordonnées (2,0)...jusqu'au pixel de coordonnées (499,0). Ensuite, nous allons changer de ligne avec le pixel de coordonnées (0,1), puis le pixel de coordonnées (1,1)...bref, le dernier pixel sera le pixel de coordonnées (499,499), tout cela grâce à la double boucle "for" !
- "r,v,b=img.getpixel((x,y))" : les coordonnées des pixels par  $(x,y)$  afin de considérer l'ensemble des pixels de l'image.