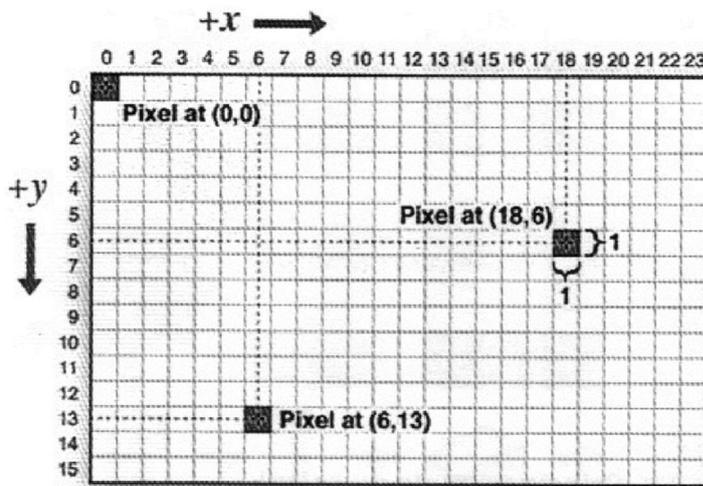


Pour un ordinateur, une image représente un fichier de données numériques. Chaque pixel est défini par ses coordonnées dans un espace à deux dimensions (tableau ou matrice). Il est possible de « manipuler » chaque pixel en modifiant ses coordonnées et/ou sa couleur.

Sur l'espace pédagogique, ouvrir la page « **Programmer le traitement d'une image** » contenant 10 situations qui vont vous apprendre à programmer de manière progressive le traitement d'une image.

Pour chacun des niveaux proposés, vous devrez aussi compléter ci-dessous la **synthèse** ci-dessous.



Niveau 1 :

L'instruction qui permet de placer un pixel de couleur (R, G, B) au point pixel de coordonnées (x, y) est

Exemple :

Niveau 2 :

Pour tracer un segment on utilise les deux instructions suivantes :

.....

Niveau 3 :

if $x < 50$ or $100 < x < 200$: signifie :

Si x est compris entre 20 et 75 (bornes comprises) ou x strictement supérieur à 100 se traduit en python par

.....

Niveau 4 :

Pour scanner (parcourir) tous les pixels d'une image de dimensions n*p on utilise la double boucle for :

.....

Niveau 5 :

L'instruction qui permet de stocker dans une variable couleur la couleur RGB d'un pixel de coordonnées (x, y) est

.....

Exemple : pour obtenir la couleur du pixel de coordonnées (23 , 157).

Niveau 6 :

Les deux instructions qui permettent de copier un pixel d'une photo pour le placer sur une image vierge est :

.....

TRAITEMENT DE L'IMAGE PAR UN ALGORITHME

Exemple :

```
(R,G,B) = photo.getpixel((50,75)) #.....  
creation.putpixel((150,25),(255,G,0)) #.....
```

Niveau 7 : Très important

Pour compter rapidement le nombre de pixels d'une image qui satisfont une condition, on utilise un compteur :

```
compteur=0 #Initialisation du compteur à 0  
for x in range(0,300): #La double boucle for pour « scanner » l'image  
    for y in range(0,300):  
        (R,G,B) = image.getpixel((x,y))  
        if ..... == ..... : #On regarde si le pixel satisfait la condition  
            compteur=compteur+1 #Le compteur augmente de 1  
print(compteur) #On affiche le compteur
```

Exemple : Dans une image de dimensions 400*300, je veux compter le nombre de pixels qui ont une composante Rouge strictement supérieure à 200.

```
compteur=0 #Initialisation du compteur à 0  
for x in range(..... , .....):  
    for y in range(..... , .....):  
        (R,G,B) = image.getpixel((x,y))  
        if ..... :  
            compteur=compteur+1  
print(compteur)
```

Niveau 8 :

Exemple : Je récupère les composantes RGB du pixel de coordonnées (30, 50) d'une photo pour le coller sur le pixel de coordonnées (70, 60) de ma nouvelle image :

```
(R,G,B) = photo. ....  
image. ....
```

Niveau 9 :

Exemple : Je récupère les composantes RGB du pixel de coordonnées (x , y) d'une photo pour le coller sur le pixel de coordonnées (300-x , 200-y) de ma nouvelle image en inversant les couleurs (négatif).

```
(R,G,B) = photo. ....  
image. ....
```

Niveau 10 :

Je pense à copier/coller des blocs d'instructions et à les modifier.

Je suis vigilant(e) sur les indentations.

J'essaie de comprendre les éventuels messages d'erreurs au moment d'exécution le code.

