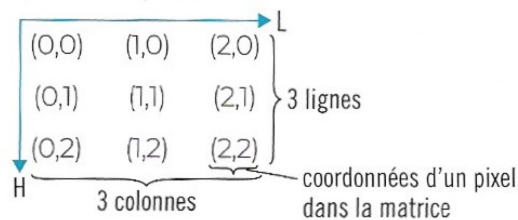


# CRÉER UNE IMAGE ET MODIFIER SES PIXELS

✚ Traiter par programme une image pour la transformer en agissant sur les trois composantes de ses pixels

## DOCUMENT 1 Créer une image pixel par pixel

L'image ci-contre est créée par le programme donné dans le document 2. C'est une image matricielle composée de 9 pixels. Mathématiquement parlant, il s'agit d'une matrice de 3 lignes et 3 colonnes. Chaque couple constitue un pixel.



On peut également imaginer un repère orthogonal (pas forcément orthonormé) dont l'origine est la pointe de la diagonale en haut à gauche de l'image. Les couleurs sont spécifiées RVB (RGB dans Python).

## DOCUMENT 2 Image matricielle de 9 pixels

```

1  from PIL import Image
2  im = Image.new("RGB", (3, 3), (255,255,255))
3  # Créer une image de 9 pixels en blanc
4  x = 0
5  y = 0
6  # Prendre le point en haut à gauche (0,0)
7  (rouge,vert,bleu) = im.getpixel((x,y))
8  # Obtenir la valeur RGB du pixel et la ranger dans (rouge,vert,bleu)
9  print (rouge, vert, bleu)
10 # Afficher la valeur RGB du pixel contenue dans (rouge,vert,bleu)
11 (rouge,vert,bleu) = (255,0,0)
12 # Mettre le rouge à 255 (maximum), le vert et le bleu à 0
13 im.putpixel((x,y),(rouge,vert,bleu))
14 # Attribuer la nouvelle valeur RGB (couleur rouge) au pixel (0,0)
15
16 x = 0
17 y = 2
18 (rouge,vert,bleu) = (0,0,255)
19 im.putpixel((x,y),(rouge,vert,bleu))      # Attribuer la couleur bleu au pixel (0,2)
20 im.putpixel((2,0),(0,0,0))                # Attribuer la couleur noir au pixel (2,0)
21 im.putpixel((2,1),(255,0,0))               # Attribuer la couleur rouge au pixel (2,1)
22
23 # Partie du programme à compléter : pixel (2,2) en jaune
24
25 im.save("im2_9px.png")
26 im.show()
27 liste_pixels = list(im.getdata())          # Obtenir les composantes RGB de tous les pixels
28
29 print(liste_pixels[0:9])                   # Afficher la liste de tous les pixels (ici 9)

```

Réponse dans l'interpréteur Python :

```

(255,255,255)
[(255, 0, 0), (255, 255, 255), (0, 0, 0), (255, 255, 255), (255, 255, 255), (255, 0, 0), (0, 0,
255), (255, 255, 255), (255, 255, 0)]

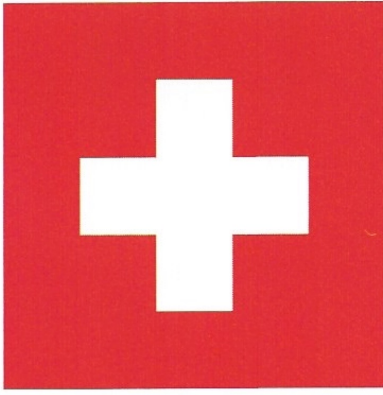
```



## DOCUMENT 3

## Boucles imbriquées

Pour créer ou modifier une image, il faut la parcourir pixel par pixel. L'utilisation de deux boucles imbriquées permet de parcourir tous les pixels d'une zone rectangulaire (lignes et colonnes). Pour chaque pixel d'une ligne, on parcourt toute la colonne de pixels puis on continue avec le pixel suivant jusqu'à la fin de la ligne. On aura ainsi accompli le balayage complet de l'image.

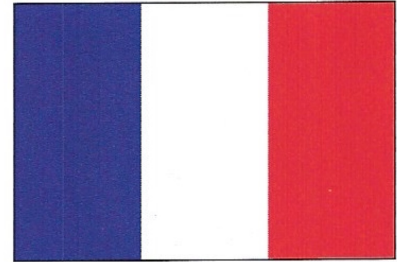


```
1 from PIL import Image
2 im = Image.new("RGB", (500,500), "RGB(255,0,0)")
3 # Créer une image de 500x500 pixels en rouge
4 # Début partie à commenter
5 for x in range(200,300):
6     for y in range(100,400):
7         im.putpixel((x,y), (255,255,255))
8 for x in range(100,400):
9     for y in range(200,300):
10        im.putpixel((x,y), (255,255,255))
11 # Fin partie à commenter
12 im.save("suisse.png")
13 im.show()
```

## DOCUMENT 4

## Créer une image par bandes de pixels

```
1 from PIL import Image
2 im = Image.new("RGB", (300,200), "RGB(0, 255, 0)")
3 # Créer un rectangle de 300x200 pixels en vert
4 for x in range(0,100):
5     for y in range(0,200):
6         im.putpixel((x,y), (0,0,255))
7 # Créer la bande bleue
8 for x in range(100,200):
9     for y in range(0,200):
10        im.putpixel((x,y), (255,255,255))
11 # Créer la bande blanche
12 # Pour x de 200 à 300
13 # Pour y de 0 à 200
14 # Attribuer à chaque pixel (x,y) la couleur rouge
15 # Créer la bande rouge
16 im.save("bleu_blanc_rouge.png")
17 im.show()
```



<1> Donnez la largeur et la hauteur en pixels de l'image. **DOC. 1**

<2> L'image est obtenue grâce à un agrandissement (avec Paint, GIMP, etc.). Expliquez pourquoi elle serait presque invisible sur l'écran d'un ordinateur. **DOC. 1**

<3> Présentez la réponse de l'interpréteur Python sous la forme d'une matrice avec les valeurs RGB. **DOC. 1** et **DOC. 2**

<4> Écrivez la partie manquante du programme. **DOC. 2**

<5> Proposez des lignes de commentaires au programme. **DOC. 3**

<6> Écrivez les trois lignes de programme manquantes : 12 à 14. **DOC. 4**