

La représentation d'une image par une machine

Correction

Exercice 2 : Création d'une image pixel par pixel

Expliquer pourquoi le pixel (5,1) est-il bleu.

La couleur bleue se définit en mode RGB de la façon suivante : (0, 0, 255)

La première boucle « for x in range(7) » dessine une **ligne rouge** en y = 1

La deuxième boucle « for y in range(5) » dessine une **colonne bleue** en x = 5

La deuxième boucle, étant exécutée **après** la première « écrase » les éléments précédemment dessinés.

Exercice 3

Expliquer à quoi sert l'instruction `r,v,b = img.getpixel((x,y))`.

L'instruction `img.getpixel ((x,y))` permet de **recupérer** la couleur d'un pixel en coordonnée (x, y) sous la forme d'un triplet de valeurs (les valeurs des trois composantes couleur).

Expliquer à quoi sert l'instruction `img.putpixel((x,y),(r,0,0))`.

L'instruction `img.putpixel((x,y),(r,0,0))` permet de **dessiner** un pixel en coordonnée (x, y) . Ce pixel sera rouge (variant en fonction de r) car les composantes vertes et bleue sont définies à 0.

Exercice 4

`Image.new(...)` permet d'initialiser une nouvelle image **avec une taille précisée et une couleur de base**.

`.save()` permet de sauvegarder les modifications dans un nouveau fichier image.

Exercice 5

Pour avoir une image jpg la commande doit être `im.save ("trace.png", "JPEG")` et non `im.save ("trace.png", "PNG")`.

Dans le deuxième cas, nous aurions une image s'appelant `trace.jpg` dont Windows considérera qu'elle est au format .jpg alors qu'elle sera en fait au format .png.

L'image jpg est floue (format avec compression) au contraire du png (format sans compression).

Une image enregistrée au format .jpg prenant moins de place qu'une au format .png.

Exercise 6



Import : from PIL import Image

Correction python

```
im = Image . new ( "RGB" , (800 ,800), "grey" )
for k in range (800) :
    im . putpixel (( k ,400) ,(255 ,255 ,255) )
im . save ( "image.png" ,"png" )
```

Exercise 7



Import : from PIL import Image

Correction python

```
im = Image . new ( "RGB" , (800 ,800), "grey" )
for k in range (800) :
    im . putpixel (( k , k ) ,(0 ,0 ,0) )
im . save ( "image.png" ,"png" )
```

Exercise 8



Import : from PIL import Image

```
from PIL import Image
im = Image.new("RGB", (800,600), "grey")
for k in range(800):
    im.putpixel((799 -k, k), (0,0,0) )
im.save("image.png", "png")
```

Exercise 9

↳ ✎

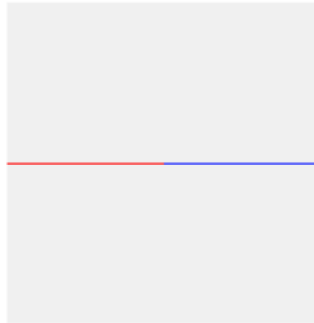


from PIL import Image

Correction python

```
im = Image . new ( "RGB" , (800 ,800), "grey" )
for k in range (0 ,800 ,2) :
    im . putpixel (( k ,400) ,(0 ,0 ,0) )
im . save ( "image.png" ,"png" )
```

Exercice 10



Import : from PIL import Image

Correction python

Méthode 1

```
im = Image . new ( "RGB" , (800 ,800), "grey" )
for k in range (400) :
    im . putpixel (( k ,400) ,(255 ,0 ,0) )
for k in range (400 ,800) :
    im . putpixel (( k ,400) ,(0 ,0 ,255) )
im . save ( "image.png" ,"png" )
```

Méthode 2

```
im = Image . new ( "RGB" , (800 ,800), "grey" )
for k in range (800) :
    if k<400 :
        im . putpixel (( k ,400) ,(255 ,0 ,0) )
    else :
        im . putpixel (( k ,400) ,(0 ,0 ,255) )
im . save ( "image.png" ,"png" )
```

Exercice 11



Import : from PIL import Image

Correction python drapeau français

""" dimension et couleur : source wikipedia """

```
im = Image . new ( "RGB" , (900 ,450) , "grey" )
```

```
for l in range (450) :
```

```
    for c in range (900) :
```

```
        if c<300 :
```

```
            im . putpixel (( c, l ) ,(5 ,20 ,64) )
```

```
        elif c<600 :
```

```
            im . putpixel (( c, l ) ,(255 ,255 ,255) )
```

```
        else :
```

```
            im . putpixel (( c, l ) ,(236 ,25 ,32) )
```

```
im . save ( "image.png" ,"png" )
```