

Créer des images avec Python



On souhaite créer des images avec le langage de programmation Python.
Pour utiliser ce langage, on lance le programme Pyzo ou EduPython par exemple.

1. Créer (et enregistrer) un nouveau fichier dans le logiciel Pyzo ou EduPython.
Pour pouvoir manipuler des images, on importe le module PIL (Python Image Library).
Voici le contenu du fichier pour commencer :

```
from PIL import Image
im = Image.new('RGB', (100,100))
im.show()
```

2. À la lecture du code, avant même de l'exécuter, que peut-on dire de l'image qui est créée ?
3. Recopier le code dans le fichier et exécuter le code.
On observe que l'image est noire. C'est parce que l'image est créée avec uniquement des 0.
4. Pour modifier des pixels de l'image, on injecte des pixels avec la méthode `putpixel`.
Cette méthode prend deux arguments : les coordonnées (x,y) du pixel à modifier, et la couleur (r,g,b) où r,g et b sont des entiers de 0 à 255.

- (a) Modifier le fichier pour que son contenu soit le suivant :

```
from PIL import Image
im = Image.new('RGB', (100,100))
for x in range(100):
    for y in range(100):
        im.putpixel((x,y), (0,255,0))
im.show()
```

- (b) Expliquer ce que font les deux boucles Pour (`for`) imbriquées dans le programme.
Pour information, avec l'instruction :

```
for x in range(100):
```

la variable x va prendre 100 valeurs au total, de 0 jusqu'à $(100-1)=99$.

- (c) Modifier les dimensions de l'image, pour qu'elle ait 256 pixels de largeur et de hauteur.
- (d) Modifier le code afin que l'image soit un dégradé vertical de bleu, avec la première ligne noire pur et la dernière ligne bleue pur.
Aide : Penser à utiliser la valeur d'une variable dans le code de la couleur.
- (e) Modifier le code afin que l'image soit un dégradé horizontal de rouge (première colonne noire pur et dernière colonne rouge pur).
- (f) Modifier le code pour créer une image de dimensions 90×90 ayant trois bandes verticales de trois couleurs différentes quelconques.

Pour cela, il faut savoir que la fonction `range` peut prendre deux arguments. Avec :

```
for x in range(20,50):
```

la variable x va prendre ses valeurs entre 20 et $(50-1)=49$, ce qui fait $(50-20)=30$ valeurs.

(g) Enregistrer l'image dans un fichier en ajoutant l'instruction suivante :

```
im.save('nom_du_fichier.ppm')
```

Ouvrir alors le fichier avec GIMP.

5. (**Facultatif** :) Créer des dégradés plus complexes, par exemple à l'envers, ou même faisant varier à la fois en ligne et en colonne, sur les trois composantes RVB.
6. (**Facultatif** :) Faire des dégradés similaires dans une image de dimensions 600×600 (les valeurs doivent là encore aller de 0 à 255). On pourra penser à utiliser des formules de proportionnalité, tout en gardant des valeurs entières.

Il peut être utile de savoir que $a // b$ donne le quotient entier de la division de a par b .

On peut par exemple obtenir quelque chose comme ci-après :

