

ICN : Vers les boucles

Objectifs

Créer des lignes de pixels de même couleur à l'aide d'une **boucle**.
Découvrir et comprendre les **boucles For**.

Un descriptif de vos premières lignes de code

Revenons sur votre premier programme pour afficher une image :

```
1 from PIL import Image
2
3 #Creation d'une image blanche RGB 8x8
4 Mario=Image.new("RGB", (8,8), (255,255,255))
5
6 #definition d'une couleur
7 rouge=(255,0,0)
8
9 #remplissage de l'image pixel par pixel
10 Mario.putpixel((0,2),rouge)
11 Mario.putpixel((7,7),(0,255,0))
12
13 Mario.show()
```

On importe une **librairie** permettant de manipuler des images.
Cette ligne doit être présente au début de chacun de vos programmes.

On crée une image de dimension 8x8
dont les pixels sont définis en couleurs RGB,
initialisés en blanc.
Cette image est stockée sous le nom « Mario ».

On crée une **variable** appelée « rouge ».
C'est un triplet.

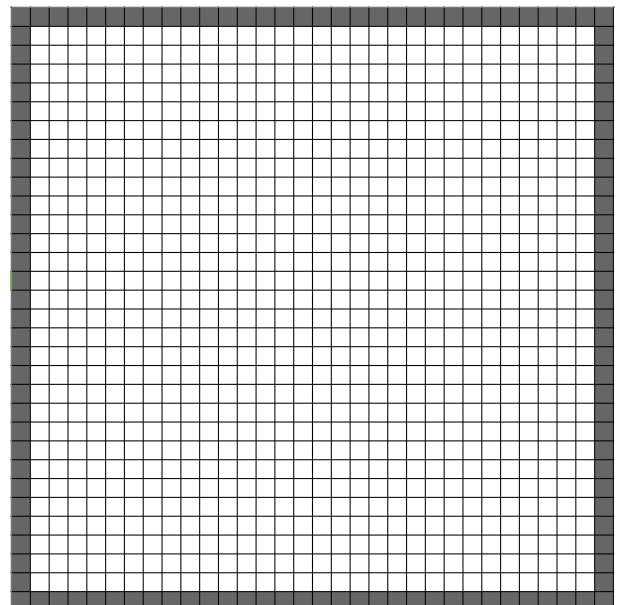
L'image « Mario » possède une **méthode** *putpixel()*.
Cette méthode demande **deux arguments** dans l'ordre :
Les coordonnées du pixel et la couleur RGB.

Cela permet de modifier la couleur du pixel ciblé
sur l'image « Mario ».

« show » se traduit par « montrer ».
Cette ligne permet d'**afficher** l'image « Mario »
à la fin de votre programme

Créer des lignes

1. Fabriquez l'image des drapeaux de l'Italie et de l'Allemagne.
(Vous choisirez une hauteur de 3 pixels et une largeur adaptée.)
2. Fabriquez un cadre gris de largeur 1 pixel sur une image de dimension 32 par 32 comme ci-contre.



Boucles, compteur et affichage

1. Dans **Pyzo**, créez un nouveau fichier que vous appellerez *boucles.py*.

2. Écrivez puis exécutez le programme ci-contre :

```
1 for i in range(0,5):
2     print("La valeur de i est : ",i)
```

3. Écrivez puis exécutez le programme ci-contre :

```
1 for i in range(0,5):
2     print("Bonjour")
3     print("Au revoir")
```

☞ Les deux premières lignes constituent une **boucle For**.

☞ Il ne faut pas oublier les " : " à la fin de la première ligne.



☞ Le contenu de la **boucle** est **indenté** par la touche :



☞ La fin de l'**indentation** signifie la fin du contenu de la boucle.

☞ *i* est le **compteur** de cette boucle.

☞ `for i in range(0,5):` peut se traduire par « Pour *i* allant de 0 à 4 faire ... ».

4. À la suite du programme précédent, écrivez puis exécutez le code ci-contre :

```
1 for i in range(0,7):
2     print(i)
3     print("Fin")
```

5. Toujours à la suite de votre fichier, créez un programme qui affiche les **nombre pairs** de 0 à 100.

6. Créez un programme qui affiche les **multiples de 7** de 0 à 700.

7. Créez un programme qui affiche les **carrés parfaits** de 0 à 256.

(Les carrés parfaits sont : 0, 1, 4, 9, 16, ... 256, ...)

Retournons à notre cadre gris

Modifiez et complétez le code suivant afin d'obtenir le cadre gris souhaité :

```
1 from PIL import Image
2
3 #Creation d'une image blanche RGB 32 par 32
4 Cadre=Image.new("RGB", (32,32), (255,255,255))
5
6 #definition d'une couleur
7 gris=(120,120,120)
8
9 #creation du cadre avec une boucle
10 for i in range(0,20):
11     Cadre.putpixel((i,0),gris)
12
13 #Affichage de l'image
14 Cadre.show()
```