

1 Paradigmes

Quels noms ne correspondent pas à un paradigme de programmation ?

1. objet
2. redondant
3. assujéti
4. impératif
5. fonctionnel

2 Analyse de code

Identifier le paradigme mis en œuvre dans chacun des extraits suivants :

— Extrait 1 :

```
image_1 = [(120, 13, 15), (255, 0, 200), (100, 145, 13)],  
[(10, 130, 15), (155, 140, 20), (1, 145, 13)],  
[(120, 130, 15), (0, 10, 100), (200, 245, 143)]  
  
def negatif(pixel):  
    R, V, B = pixel  
    return (255-R, 255-V, 255-B)  
  
def filtrer_image(image, filtre):  
    nouvelle_image = []  
    for i in range(len(image)):  
        ligne = []  
        for j in range(len(image[0])):  
            ligne.append(filtre(image[i][j]))  
        nouvelle_image.append(ligne)  
    return nouvelle_image  
  
image_2 = filtrer_image(image_1, negatif)
```

— Extrait 2 :

```
image_2 = Image()  
image_2.definir_contenu(image_1)  
image_2.appliquer_negatif()  
print(image_2.contenu)
```

— Extrait 3 :

```
for i in range(len(image_1)):  
    for j in range(len(image_1[0])):  
        image_1[i][j] = negatif(image_1[i][j])
```

3 Effet de bord

Saurez-vous identifier l'effet de bord qui apparaît dans la suite d'instructions suivantes ?

```
1 | liste = [1, 2, 3]
2 | matrice = [liste for i in range(2)]
3 | print(matrice)
4 | liste[1] = 0
5 | print(matrice)
```

Que valent les variable `liste` et `matrice` à l'issue des instructions ?

Quel paradigme de programmation est « violé » ici ?

Exécuter ce code dans PythonTutor : <http://pythontutor.com/visualize.html>

4 Découvrir les fonctions built-in `map` et `filter`

<https://docs.python.org/fr/3/library/functions.html>

Lire attentivement les instructions suivantes :

```
1 | liste_0 = [i for i in range(10)]
2 |
3 | def double(n):
4 |     return 2 * n
5 |
6 | def est_pair(n):
7 |     return n % 2 == 0
8 |
9 | liste_1 = list(map(double, liste_0))
10 | liste_2 = list(filter(est_pair, liste_0))
```

Lire la documentation sur les fonctions `map` et `filter` puis expliquer ce que valent `liste_1` et `liste_2` (*remarque* : `list(iterable)` crée une liste-python avec les éléments d'un iterable).

Quel paradigme de programmation est illustré par l'utilisation de `map` et `filter` ?

Que vaut `liste_0` en fin d'exécution ?