

4 – Guide de style

On oublie souvent qu'un programme sera relu et modifié par un humain, que ce soit son auteur ou un autre développeur.

C'est pourquoi, adopter un style d'écriture standard facilite cette relecture.

1 – Choisir correctement des noms

Syntaxiquement, les noms de variables, des fonctions, des classes, des méthodes, des attributs, peuvent contenir des lettres, des chiffres, des caractères "_" et ne doivent pas commencer par un chiffre.

Quel que soit le langage, on choisira un nom d'autant plus évocateur qu'il est utilisé dans une grande portion de code.

La PEP8 ajoute les conventions suivantes :

(guide de style Python accessible sur <https://www.python.org/dev/peps/pep-0008>)

- Noms de modules courts, en minuscule et de préférence sans "_"
Exemple : `crypto`
- Noms de classes ou de type en CamelCase, c'est à dire en minuscules, sans "_", avec une majuscule en début de chaque mots
Exemple : `class NombrePremier`
- Noms des fonctions, des méthodes, des attributs ou des variables en minuscules, les mots séparés par des "_"
Exemple : `def decomposition_facteurs_premiers`
- Les variables utilisés comme constantes sont en majuscule, les mots séparés par des "_"
Exemple : `TOTAL_MAX = 10`

2 – Espaces, indentations, lignes blanches

Indentation

En Python, les blocs sont délimités par l'indentation. La PEP8 propose d'indenter les blocs à l'aide de 4 "caractères" espace (utiliser par la plupart des IDE Python).

Une ligne ne devrait pas excéder 79 caractères (pas toujours respecté). Quand une instruction doit courir sur plusieurs lignes, on facilite la lecture en indentant.

Exemple tiré de la PEP8 :

```
foo = nom_de_fonction_long(var_one, var_two,  
                             var_three, var_four)
```

Espaces

Les règles suivantes permettent de bien distribuer les espaces :

- Pas d'espace avant ":"
- Espace après mais pas avant les ",", " (dans les appels ou les définitions de fonction)
- Espace autour des "="
- Pas d'espace avant les "(", "[", "{", ni après
- pas d'espace autour des ":" dans les slices
 - `print(lst[2:6])`

Lignes blanches

- On laisse 2 lignes vides entre les différentes fonctions ou classes à l'intérieur d'un module
- Dans une même classe, les méthodes sont séparées par une seule ligne blanche

Outils de validation

Des outils sont souvent intégrés dans les IDE, mais aussi disponible séparément :

- *pep8* et *flake8* permettent de vérifier la conformité avec la PEP8 (ainsi que d'autres choses)
- *autopep8* et *black* peuvent reformater automatiquement le code en suivant ces règles

Utilisation de *autopep8* : dans la console

```
pip install --upgrade autopep8
```

```
autopep8 --in-place --aggressive --aggressive <filename>
```

Regarder aussi comment fonctionnent : *pep8* / *flake8* / *black*