La norme PEP8

La norme **PEP8** rassemble un ensemble de règles d'écritures en Python, concernant essentiellement les espaces, les retours chariots ainsi que les noms des variables. Elle constitue une réelle harmonisation des méthodes et des conventions de "codage".

Utilité

La communauté Python, grandissant exponentiellement, il semble nécessaire de fixer des "règles", permettant aux utilisateurs d'adopter la même rédaction du code.
Dans les forums notamment StackOverflow et developpez.com, la norme *PEP8* permet une meilleure lisibilité, ainsi elle produit un plus grand attrait de Python. En effet, il est souvent représenté tel un langage peu esthétique et donc moins accessible pour les *néophytes*.

Des normes

Les espaces

Les opérateurs doivent être absolument entourés d'espaces.

```
# Correct
ma_variable = 'value'
1 + 2

# Incorrect
ma_variable='value'
1+2
```

Cependant, il existe une exception fréquente : un argument avec une valeur par défaut.

```
# Correct
fonct(arg='value')
```

Les lignes

Par convention, une ligne doit se limiter à 79 caractères. Ceci est un héritage des petits écrans.

Pour une ligne trop longue, on peut donc utiliser l'indentation, le slash `/` à la fin d'une ligne, ...

```
# Indentation
foo = la_chose_au_nom_si_long_quelle_ne_tient_pas_sur(
      une,
      carte,
      de,
      munchkin)

# Slash
queryset = ModelDjangoALaNoix.objects\
                        .filter(banzai=True)\
                        .exclude(chawarma=False)

# Parenthèses
from file_info import (file_extension, is_compressed, file_name,
       file_count_sheets, path_folder)

# Multiples guillemets
s = ("Les chaînes Python sont automatiquement"
 "concaténées par la VM si elles sont "
 "uniquement séparées par des espaces "
 "ou sauts de lignes.")
```

Les noms

Si deux modules sont importés, ils doivent obligatoirement figuré dans 2 lignes d'importation différentes.

```
# Correct
import geopandas
import xlrd

# Incorrect
import xlrd, geopandas
```

Les variables, dans la globalité, sont composées de lettres minuscules `abcde..yz` et d'underscores `_`.
Retour sur ces règles ...

```
# Boucle et indices
# --> Minuscule simple
for x in range(10):
    print(x)

# Modules, variables, fonctions, ...
# --> Minuscules et underscores
une_variable = 10
def une_fonction():

# Constantes
MAX_LENGTH = 10

# Classe (Camel Case)
class JeSuisUneClasse:
```

Mon code suit-il les conventions ?

il existe une librairie en Python, permettant de vérifier si un fichier est conventionné dans la norme PEP8 : c'est pep8.
On l'installe avec la commande suivante.

```
Installation
$ pip install pep8

Utilisation/exécution
$ pep8 main.py
```

Sources

Ce document a été fortement inspiré du site Sam et Max qui explique le Python avec humour, et surtout de manière très accessible .

Voici quelques détails sur le fonctionnement de pep8 en Python.