How to Write Beautiful Python Code With PEP 8

PEP 8 exists to improve the readability of Python code.

Naming Conventions

- Choosing sensible names will save us time and energy later. Easy to figure out, from the name, what a certain variable, function, or class represents. Makes code readable for other developers
- Never use l, O, or I single letter names as these can be mistaken for 1 and 0

Type	Naming Convention	Examples
Function	Use a lowercase word or words. Separate words by underscores to improve readability.	`function`, `my_function`
Variable	Use a lowercase single letter, word, or words. Separate words with underscores to improve readability.	`x`, `var`, `my_variable`
Class	Start each word with a capital letter. Do not separate words with underscores. This style is called camel case or pascal case.	`Model`, `MyClass`
Method	Use a lowercase word or words. Separate words with underscores to improve readability.	`class_method`, `method`
Constant	Use an uppercase single letter, word, or words. Separate words with underscores to improve readability.	`CONSTANT`, `MY_CONSTANT`, `MY_LONG_CONSTANT`
Module	Use a short, lowercase word or words. Separate words with underscores to improve readability.	`module.py`, `my_module.py`
Package	Use a short, lowercase word or words. Do not separate words with underscores.	`package`, `mypackage`

Ref: https://realpython.com/python-pep8/#why-we-need-pep-8

Choosing Names

- The best way to name your objects in Python is to use descriptive names to make it clear what the object represents.

Blank LInes

- Vertical whitespace, or blank lines, can greatly improve the readability of your code.
- **Surround top-level functions and classes with two blank lines**. Top-level functions and classes should be fairly self-contained and handle separate functionality.
- **Surround method definitions inside classes with a single blank line**. Inside a class, functions are all related to one another.
- **Use blank lines sparingly inside functions to show clear steps.** Leave a blank line between several return statements to show what is being returned clearly.

Maximum Line Length and Line Breaking

- 79 characters per line.
- Use \ backslashes to break lines.
- If line breaking needs to occur around binary operators, like + and * before the operator.

Indentation

- Use 4 consecutive spaces to indicate indentation.
- Prefer spaces over tabs.
- Indent line breaks to improve readability.
- For nested code blocks, add comment after the final condition or add extra indentation on the line continuation.

Comments

Block comments

- Indent block comments to the same level as the code they describe.
- Start each line with a # followed by a single space.
- Separate paragraphs by a line containing a single #.

Inline comments

- Use inline comments sparingly.
- Write inline comments on the same line as the statement they refer to.
- Separate inline comments by two or more spaces from the statement.
- Start inline comments with a # and a single space, like block comments.
- Don’t use them to explain the obvious.

Linters

Linters are programs that analyse code and flag errors. They provide suggestions on how to fix the error.

Autoformatters

Autoformatters are programs that refactor your code to conform with PEP 8 automatically. Once such a program is black, which formats code following most of the rules in PEP 8.

Programming Recommendations

- Don’t compare Boolean values to True or False using the equivalence operator.
- Use the fact that empty sequences are falsy in if statements.
- Use is not rather than not ... is in if statements.
- Don’t use if x: when you mean if x is not None:.
- Use .startswith() and .endswith() instead of slicing.