

Python Coding Conventions

References:

- [PEP 8 -- Style Guide for Python Code | Python.org](#) (official from Python.org, co-written by Guido van Rossum - creator of Python)
- [How to Write Beautiful Python Code With PEP 8 – Real Python](#) (easy to follow and quality Python resource page)
- [Python Variable Names \(w3schools.com\)](#) (from w3school, short and concise, quality resource)

These coding conventions are the best practice, style guide for Python programming project, some of it are:

- Naming and declaration rules for variables and functions.
- Rules for the use of white space, indentation, and comments.
- Programming practices and principles

By following the coding convention, is to enhance and ensure the consistency, readability for Python project:

- Improves code readability
- Make code maintenance easier

Hint:

- For VS Code User, you might want to checkout:
 - [Beautify - Visual Studio Marketplace](#)
 - [Prettier - Code formatter - Visual Studio Marketplace](#)
- For general python user:
 - [autopep8](#) · [PyPI](#)

Variable Names

Use **underscore_case** for identifier names (variables and functions).

```
first_name = "Nhan Kiet"
last_name = "To"

# Aligned with opening delimiter.
foo = long_function_name(var_one, var_two,
                          var_three, var_four)

# Add 4 spaces (an extra level of indentation) to distinguish arguments from the rest.
def long_function_name(
    var_one, var_two, var_three,
    var_four):
    print(var_one)

# Hanging indents should add a level.
foo = long_function_name(
    var_one, var_two,
    var_three, var_four)
```

Long list or iterators with many items

Preferably 3 items per row:

```
# Both are correct
my_list = [
    1, 2, 3,
    4, 5, 6,
]
result = some_function_that_takes_arguments(
    'a', 'b', 'c',
    'd', 'e', 'f',
)

my_list = [
    1, 2, 3,
    4, 5, 6,
]
result = some_function_that_takes_arguments(
    'a', 'b', 'c',
    'd', 'e', 'f',
)
```

Import external libraries

Each library should be in a line, alphabetically order:

```
# Correct:
import os
import sys

# Wrong:
import sys, os

# Correct:
from subprocess import Popen, PIPE
```

Whitespace in Expressions and Statements

Avoid extraneous whitespace in the following situations:

```
# Correct:
spam(ham[1], {eggs: 2})

# Wrong:
spam( ham[ 1 ], { eggs: 2 } )

# Correct:
foo = (0,)

# Wrong:
bar = (0, )

# Correct:
x = 1
y = 2
long_variable = 3

# Wrong:
x             = 1
y             = 2
long_variable = 3
```

Comment

Documetion Strings (DocString) - for function, class or file documentation

```
"""Return a foobang

Optional plotz says to frobnicate the bizbaz first.
"""
```

Inline comment

```
x = x + 1           # Increment x
```