

Pytho code conventions: A summary

General

This document is a summary of code conventions in Python programming language. The document is based on PEP8¹ and contains only a part of the conventions.

Variable names and functions

General

All names are to be meaningful and indicative.

Name conventions

- 1) **Variable Names**

Lowercase letters, where words are separated by an underscore.
Variable names should not start with a nondescript prefix.

Correct: `name_of_student = 'Moshe'`
Incorrect: `str_name_of_student = 'Moshe'`
`nameOfSutdent = 'Moshe'`
`n = 'Moshe'`
- 2) **Function Names**

Lowercase latin letters, where words are separated by an underscore.

Correct: `get_student_grade(student_name)`
Incorrect: `getStudentGrade(studentName)`
- 3) **Constants**

Variables whose values are not to be changed will be written in uppercase, separated by an underscore.

Correct: `BEST_PTOGRAM = 'ITC Fellows'`
Incorrect: `best_program = 'ITC Fellows'`

Documentation

- 4) Documentation strings:
 - a. All files (modules) should include a short docstring including the author's name and a short description of the module's functionality.
 - b. All methods and functions must include a docstring.
 - c. **Note:** Docstrings do not come in place of indicative variable names.
- 5) Documenting using comments:
 - a. The code should be documented so that it is easily understandable.
 - b. There is no need to document every line, and no need to document the obvious.

Style

- 6) Don't use 'magic numbers'. Instead, create an appropriate, meaningful constant.


```
print sys.argv[2]
USER_NAME = 2
print sys.argv[USER_NAME]
```

¹ <https://www.python.org/dev/peps/pep-0008>

Omer Rosenbaum, Shay Sadovsky

- 7) Use the built-in constants True and False, not 0 and 1.
- 8) Every variable in the code will be used for one purpose only.
- 9) Operators are to be enclosed by one space on each side (apart from the negativity minus)


```
1 + 2 * 3 == -3 + 10
1+2*3== -3+10
birthday_string = "happy birthday!"
birthday_string="sad birthday :("
```
- 10) A space will come after a comma and not before it.


```
example_function(1, 2, 3)
example_function(1 , 2 , 3)
```
- 11) Do not write lines too long. The maximum characters in a line should be **79**.

Rules of thumb

- 12) When checking if an object is None, use the *is* operator.


```
if object is None:
if object == None:
```
- 13) Do not affix mutable objects (like a list or a dictionary) as default parameters of a function.


```
def function(default=None):
    if default is None:
        default={}

def function(default={}):
```
- 14) Do not use Python keywords as variable names.


```
list = [1, 2]
numbers_list = [1, 2]
```