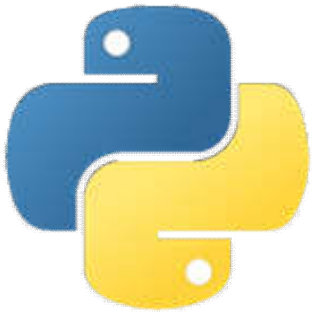

Coding convention

Nội dung

1. **PEP 8**
2. **Convention cơ bản**
1. **Q&A**

1. What’s PEP 8?

- Là tiêu chuẩn coding convention, bao gồm 1 bộ các quy tắc trong coding.
- Python sử dụng chuẩn PEP 8 làm conding convention.

2. Coding convention

❖Indentation (thụt lùi đầu dòng): Sử dụng 4 khoảng trắng hay 1 tab cho mỗi cấp lùi đầu dòng

❑ Yes:

```
Ví dụ 1:
foo = long_function_name(
    var_one, var_two,
    var_three, var_four)
Ví dụ 2:
for letter in 'Phuong':
    if letter == 'o':
        pass
        print('Pass block')
    print('Current Letter:', letter)
print('Out of for!')
```

❑No:

```
45 for letter in 'Phuong':
46     if letter == 'o':
47         pass
48         print('Pass block')
49         print('Current Letter :', letter)
50
51 print('Out of for!')
```

Bad Indentation (2 spaces)

→ Giải pháp: Sử dụng Format code (Trong Pycharm phím tắt là tổ hợp Ctrl + Alt + L).

- Đối với 1 dòng code, chiều dài tối đa nên là < 80 kỳ tự.
- Đối với khối ghi chú (comment): mỗi dòng nên giới hạn 72 ký tự.
- Nên ngắt thành nhiều dòng đối với dòng code quá dài, và bổ sung thêm dấu “\”. Chú ý khoảng cách thụt đầu dòng vẫn có cấp ngang với dòng trên.

❑Yes:

```
with open('/path/to/some/file/you/want/to/read') as file_1, \
     open('/path/to/some/file/being/written', 'w') as file_2:
    file_2.write(file_1.read())
```

```
income = (gross_wages
          + taxable_interest
          + (dividends - qualified_dividends)
          - ira_deduction
          - student_loan_interest)
```

❑No:

```
with open('/path/to/some/file/you/want/to/read') as file_1,
open('/path/to/some/file/being/written', 'w') as file_2:
file_2.write(file_1.read())
```

```
income = (gross_wages +

          taxable_interest +

          (dividends - qualified_dividends) -

          ira_deduction -

          student_loan_interest)
```

- Blank line (dòng trống):
  - Giữa các phương thức cách nhau bởi 1 dòng trống
  - Giữa lớp và phương thức cách bởi 2 dòng trống
  - Các khối code riêng biết cách nhau bằng 1 dòng trống

❑Yes:

```
class Point:
    '''
    classdocs
    '''


    def __init__(self, x=0,y=0):
        '''
        Constructor
        '''
        self.x = x
        self.y = y

    def __str__(self):
        return "x-value: " + str(self.x) + " y-value: " + str(self.y)

    def __add__(self, other):
        p = Point()
        p.x = self.x + other.x
        p.y = self.y + other.y

        return p
```

❑No:

```
class Point:
    '''
    classdocs
    '''
    def __init__(self, x=0,y=0):
        '''
        Constructor
        '''
        self.x = x
        self.y = y
    def __str__(self):
        return "x-value: " + str(self.x) + " y-value: " + str(self.y)
    def __add__(self, other):
        p = Point()
        p.x = self.x + other.x
        p.y = self.y + other.y

        return p
```

❖Import: Mỗi thư viện/module import được đặt trên một dòng riêng biệt.

❑Yes:

```
import os
import sys
```

```
from subprocess import Popen, PIPE
```

❑No:

```
import sys, os
```

❖Đối với tên biến bắt đầu bởi 2 dấu _ liên tục, đặt sau phần comment giới thiệu vê module nhưng đặt trước import. Nội dung trong ghi chú nên dùng dấu nháy đôi “”.

```
"""This is the example module.
This module does stuff.
"""

from __future__ import barry_as_FLUFL

__all__ = ['a', 'b', 'c']
__version__ = '0.1'
__author__ = 'Cardinal Biggles'

import os
import sys
```

- White space (khoảng trắng): Nên sử dụng khoảng trắng phía sau , hoặc ; hoặc :
- Không dùng khoảng trắng giữa tên hàm và dấu ngoặc () hoặc trước dấu []

❖Ghi chú:

- ❑Thông tin: Tác giả, version, ngày viết
- ❑Giải thích cho những đoạn code khó và nên sử dụng English
- ❑Ghi chú ngắn gọn

```
"""
@author: KTPHUONG
@version: 1.0
@since: Jan 05, 2017
"""
```

```
# First comment
print "Hello, Python!" # second comment
x = x + 1              # Increment x
```

- Naming convention:
  - module_name
  - package_name
  - ClassName
  - method_name
  - ExceptionName
  - function_name

- GLOBAL_CONSTANT_NAME
- global_var_name
- instance_var_name
- function_parameter_name
- local_var_name


Q
&
A