

Coding Standards (Python)

1 Naming convention

- **variable name**

Variable names should be in camelCase.

Examples:

myName,
myWordFile,
isTeacher

- **Constant Name**

In Python, constants are usually declared and assigned in a module. Inside the module, constants are written in all capital letters and underscores separating the words.

Examples:

PI,
GRAVITY,
MAX.VALUE

- **Method Name**

Method name should be in camelCase.

Example:

```
def myFunction():  
    print('Hello world!')
```

- **Parameter Name**

Parameter names should be in camelCase.

Example:

```
function addNum(numX, numY):  
    sum = numX + numY
```

- **Class Name**

Class names should be in camelCase.

Example:

```
class myName:
    def __init__(self, name, age):
        self.name = name
        self.age = age
```

- **Interface Name**

Interface names should be in camelCase.

Example:

```
import zope.interface
class myInterface(zope.interface.Interface):
    x = zope.interface.Attribute("foo")
    def method1(self, x):
        pass
    def method2(self):
        pass
@zope.interface.implementer(myInterface)
class myClass:
    def method1(self, x):
        return x**2
    def method2(self):
        return "foo"
```

- **Exception Name**

In python, The critical operation which can raise an exception is placed inside the try clause. The code that handles the exceptions is written in the except clause. Here, import module sys to get the type of exception.

Example:

```
import sys
randomList = ['a', 0, 2]
for entry in randomList:
    try:
        print('The entry is', entry)
        r = 1/int(entry)
        break
    except:
        print('Oops!', sys.exc_info()[0], 'occurred.')
        print('Next entry.')
        print()
print('The reciprocal of', entry, 'is', r)
```

2 Comments

Comments start with a `#` and a single space.

Examples:

Single line comment:

```
# This is a comment.
```

```
print('Hello, World!')
```

Multi-line comment:

Python does not really have a syntax for multi line comments. To add a multi-line comment, insert a `#` for each line.

```
# This is a comment
```

```
# written in
```

```
# more than just one line.
```

```
print('Hello, World!')
```

Inline comment:

Inline comments should be separated by at least one tab from the statement if necessary.

Example:

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

3 Indentation

The number of tab per indentation level must be one.

Example:

```
def myPet:
    print('This is my pet cat!')
```

4 Variable Ordering

Variables of classes must be in the order: private, protected and public.

5 Method Ordering

Methods of classes must be in the order: constructor, private, protected and public.

6 Import

Imports should always be on separate lines and put at the top of the file.

Example:

```
import os
```

```
import sys
```

Imports should be grouped in the following order:

1. Standard library imports.
2. Related third party imports
3. Local application/library specific imports.

Example:

```
import mypkg.sibling
```

```
from mypkg import sibling
```

```
from mypkg.sibling import example
```

7 Line Break

Formulas within a paragraph should break before binary operator if necessary.

Example:

```
income = (grossWages
          + taxableInterest
          + (dividends - qualifiedDividends)
          - iraDeduction
          - studentLoanInterest)
```