**PYTHON CODING STANDARDS**

1. **Indentation in function arguments and function statements:-**

# Correct

samp = function(var\_one, var\_two,

var\_three, var\_four)

def long\_function\_name(

var\_1, var\_two, var\_three,

var\_four,var\_5):

#wrong

foo = long\_function\_name(var\_one, var\_two,

var\_three, var\_four)

def long\_function\_name(

var\_one, var\_two, var\_three,

var\_four):

print(var\_one)

1. **Tabs or space to use for indentation**

* spaces are the preferred indentation method.
* Python disallows mixing tabs and spaces for indentation.

1. **Operators readability**

# Correct:

# easy to match operators with operands

AJ = (one

+ two

+ (one - two)

- (one + two))

# wrong

AJ = (one +

two +

(one - two) -

(one + two))

**4)Imports format**

# Correct:

import math

import sys

from math import sqrt,mean

# Wrong:

import sys, math

# for module of one class which is in another class

from foo.bar.yourclass import YourClass

**5)Immediately inside parentheses, brackets or braces:**

# Correct:

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

# Wrong:

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

# Correct:

foo = (0,)

# Wrong:

bar = (0, )

6) Multiple statements on same line are not encouraged

# Correct:

if foo == 'blah':

do\_blah\_thing()

do\_one()

do\_two()

do\_three()

# Wrong:

if foo == 'blah': do\_blah\_thing()

do\_one(); do\_two(); do\_three()

7) Trailing Commas

 when a version control system is used, when a list of values, arguments or imported items is expected to be extended over time.

 it does not make sense to have a trailing comma on the same line as the closing delimiter

# Correct:

FILES = [

'setup.cfg',

'tox.ini',

]

# wrong

FILES = ['setup.cfg', 'tox.ini',]

8) DocString

 you should have a comment that describes what the method does. This comment should appear after the def line.

Def fn():

"""Return a nan

A demo function

"""

**9) Naming convention**

Never use the characters 'l' (lowercase letter el), 'O' (uppercase letter oh), or 'I' (uppercase letter eye) as single character variable names.

In some fonts, these characters are indistinguishable from the numerals one and zero. When tempted to use 'l', use 'L' instead.

**10) def vs lambda**

# Correct:

def f(x): return 2\*x

# Wrong:

f = lambda x: 2\*x