Het Deepak Raval
FYIT Roll No - 68

What is PEP?

The PEP is an abbreviation form of Python Enteprise Proposal. Writing code with proper logic is a key factor of programming, but many other important factors can effect the code's quality. The developer's coding style makes the code much relicable, and every developer should keep in mind that Python strictly follows the way of order and formant of the string.

Adaptive a nice coding style makes the code more readable. The code becomes easy for end-user.

PEP 8 is a document that provides various guidelines various guidelinesc to write the readable. Barry Warsaw, and Nick Coghlan. The main aim of PEP is to enchance the readability and consistency of code.

Why PEP 8 is important?

PEP 8 enchances the readability of the Python code, but why is readability so important? Lets understand this concept.

Creator of Python Guido Van Sossum said "Code is much more often than it is written." The code can be written in a few minutes, a few hours, or a whole day but once we have written the code, we will never actually it again. But sometimes we need to read the code again and again.

At this point we must have an idea of why we wrote the particular line in the code. The code should reflect the meaning of each line. That's why readability is so much important.

We will describe Naming Convention

When we write the code, we need to asign name to many things such as variables, functions, classes, packages, and a lot will save time and energy. When we look back to the files after sometime we can easily recall what a certain variables, function or class represents. Developers should avoid choosing inappropriate names. The naming conventions.

Lowercase

```
Var = 10
```

lower_case_with_underscore.
number_of_apple = 5

UPPERCASE
VAR = 6

Name Style
Below is the table that specifies some of the common naming styles in Python

	Type	Naming Convention	Examples
1	Function	We should use the lowercase words or separate words by the underscore	myfunction, my_function
2	Variable	We should use a lowercase letter words or seperate words to enhance the readability	a, var, variable_name.
3	Class	The first letter of your class name should be capitalized use camel case. Do not separate words with the underscore	Myclass, Form Model

Method	We should use a lower case letter, words or seperate words to enchance readability,	My constant, CONSTANT, MY_CONSTANT
Module	We should use a lowercase letter, words, or seperate words to enchance the readability	Module-name.py, module.py.
Package	We should use a lower case letter, words to enchance the durability. Do not separate words with the underscore.	package, mypackage.

These are some common naming conventions that are useful to beauty the Python code.

Code Layout

The code layout defines how much the code is readable. In this section we will learn how to use whitespace to improve code readability.

Indention

Unlike other programming languages, the indentation is used to define the code block in Python. The indentations are the important part of the Python Programming language and it determines the level of lines of code. Generally we use the 4 space for indentation.

Example

```
x = 5
if x == 5
    print ('x is larger than 5')
```

In above example, the in part print statement will get executed if the condition of if statement is true. This indentation defines the code block and tells us what statements execute when a function is called or condition triggers.

Tabs vs Space

We can also use the tabs to provide the consecutive spaces to indicate the indentation, but white spaces are the most preferable. Python 2 allows the mixing of tabs and spaces but we'll get an error in Python 3.

Indention following the Breaks

It is essential to use ~~not~~ identation when using line continuations to keep lines to fewer than 79 characters. It provides the flexibility to determining between two lines of code and a single line of code that extends two lines.

Example

```
obj = func-name (argument-one, argument,
                 two argument-three,
                 argument-four.
```

Adding 4 spaces from the second line to discriminate arguments from the rest

```
def function_name(
    argument_one, argument_two, argument_
    three, argument_four):
        print(argument_two)
```

4 space indentation to add a level.

```
foo = long_function_name(
    var_one_var_two,
    var_three, var_four)
```

Use docstring

Python provides the two types of document Strings or docstring - single line and multiple lines. We use the triple quotes to define a single line or multiline quotes. Basically, these are used to describe the function or particular program. Lets understand with example.

Example :-

```
""" This is a simple add method """
""" This is
        a
```

(8)

Simple add program to add the two numbers " " "

Should a line Break Before or After a Binary Operator?

The lines break before or after a binary operation is a traditional approach. But it affects the readability extensively because the operator are scattered across the different screens, and each operator is kept away from its operand and onto the previous line.

Example

Total marks = (English_marks + math_marks + (science_marks - biology marks) + physics_marks)

Python allows us to break line before or after a binary operator as long as the convention is consistent locally.

Importing module.

We should important the the modulisum the separate line as follows.

Eg: import pygame
import os
import sys.

Instead of following example we can write the import statement as.

Eg: import sys, os.

We can also use the following approach.

Eg from subprocess import Popen, PIPE.

The import statement should be written at the top of the file or just after any module comment are more readable and tend to be better behaved.

import mypkg.siblings
from mypkg import siblings
from mypkg.siblings import example.

Blank lines

Blank lines can be improved the readability of Python code. If my lines of code bunched together the code will become harder to read. we can remove this by using the many blank vertical lines, and the reader might need to

(10)

Instruction to add virtual whitespace

1) Top level function and class with two lines

CODE
```
Class FirstClass:
    pass

class secondclass:
    pass

def main-function():
    return None
```

* Single blank lines inside classes

The function that we defined in the class is related to one another.

Example
```
Class Firstclass:
    def method-one (self):
        return None

    def second-two(self):
        return None
```

Use blank lines inside the function:
sometimes we need to write a complicated function has consists of several steps deefoee the return statement.
So we can add the blank line between each step.

Example:-

```
def cal_variance (in list):
    list_sum = 0
    for n in n_list
      list_sum : list num +n
      men = list_sum / (n_list)

    Squade_sum = 0
     for n in n list:
        squade_sum = squade sum +n**2
        men_squde = squade_num/bn (nlist)

    return men_squades_men**2
```

* Putting the cloving Braces.

Example
space
list_numbers = [
5, 4, 1
4, 6, 3
7, 8, 9
]

* Line up the closing braces with the first characters of line.

```
list_numbers = [
  5, 4, 1,
  4, 6, 3,
  7, 8, 9,
  ]
```

Comments

Comments are the integral part of any programming language. These are the least way to understand the code.
But we should remember the following points.

Block Comment

Block comments are the good choice for the small section of code. Such comments are useful when we describe several line codes to perform a single action such as iterating a loop.
They help us to understand the purpose of the code.

(13)

PEP 8 provides the following rules to write comment block.

1) Indent block comment should be at the same level.

2) Start each line with the # followed by a single space.

3) Seperate line using the single #.

Example

```
for i in range (0,5):
    # loop will iterate over i five times and
      print out the value of i:
    # new line character.
      print (i,'\n')
```

Inline Comments.

1) Start comments with the # and single space

2) Use inline comment carefully.

3) We should seperate the inline comments on the same line as the statement they refer.

Example of inline comment.

```
a=10 # this a is variable that holds
       integer value.
```

Sometimes, we can use the naming convention to replace the inline comment.

```
X='Peter Decosta' # this is a student.
```

We can use the following name convention.

```
Student-name ='Peter Decosta'
```

Inline comment are essential but block comments make the code more readable.

Example :-

```
# Recommended
list 1= [1, 2, 3]
# Not recommended
Lit 1= [1, 2, 3]
```

Programming recommendation

As we know that, there are several methods to perform similar tasks in Python. In this section

(15)

We will see some of the suggestions of PEP 8 to improve the consistency.

Avoid Comparing Boolean values using the equivalence operator.

Example :-

```
# Not Recommended

bool_value = 10>5
if bool_value == True:
    return '10 is bigger then 5'
```

We shouldn't use the == to operator to compare the Boolean value, It can only take the True or False.

Example

```
if my_bool:
    return '10 is bigger then 5'.
```