

Name : Shah Kunjan Bharat Class : FYIT Roll No : 75

## PEP in Python

### What is PEP?

→ The PEP is an abbreviation form of Python Enterprise  
Personal. Writing code with proper logic is a key  
factor of Programming, but many other Important  
factors can affect the code's quality. The  
developer's Coding Style makes the code much  
easier, and Every developer should keep in  
mind that Python Strictly follows the way of  
order and format 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 to write the readable in Python. PEP  
8 describes how the developer can write beautiful code.  
It was officially written in 2021 by Guido Van  
Rossum, Barry Warsaw, and Nick Coghlan. The  
main aim of PEP is to Enhance the readability and  
and Consistency of Code.

Name: Shah Krunjan Bhagat Class: FYIT Roll No: 75

## Why PEP 8 is Important?

→ PEP 8 Enhances the readability of the Python Code, but why is readability so important? Creator of Python, Guido Van Rossum said, "Code is much more often read 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 re-write 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 few important guidelines for writing effective code that can be read by others as well.

## Naming Convention

When we write the code, we need to assign name to many things such as Variables, functions, classes, packages, and a lot more things. Selecting a proper name will save time and energy, when we look back to file after sometime, we can easily recall what a certain

Name: Shah Kunjan Bhagat Class: FYIT Roll No: 75

Variable, function, or class Represents Developers should avoid choosing inappropriate names. The naming convention in Python is slightly messy, but there are certain conventions that we can follow easily; let's see the following naming convention.

Example.

Single lowercase letter,

a = 10

Single uppercase letter

A = 10

Lower Case

Var = 10

lower\_case\_with\_underscores

number\_of\_apples = 5

UPPERCASE

Var = 6

UPPER-CASE-WITH-UNDERSCORES

NUM\_OF\_CHARS = 20

Name: Shah Kunjan Bhagat Class: FYIT Roll No: 75

## Capitalized words (or Camel Case)

Number of Books = 100

### Name Style

Type	Naming Convention	Examples
Function	We should use the lowercase words or separate words by the underscore	my function, my_function

Variable	We should use a lowercase letter, words, or separate words to enhance the readability.	a, vay, variable_name
----------	--	-----------------------

Class	The first letter of class name should be Capitalized; use Camel Case. Do not separate words with the underscore!	MyClass, form, Model
-------	--	----------------------

Method	We should use a short, uppercase, letter, words, or separate words to enhance the readability.	MY_CONSTANT, CONSTANT, my_CONSTANT
--------	--	--

Name: Shah Kunjan Bharat Class: FYIT Reg No: 75

Module We should use a short, uppercase letter, words, or separate words to enhance the readability

MY CONSTANT  
CONSTANT  
MY\_CONSTANT

Package We should use a lower case letter, words or separate words to enhance the readability. Do not separate words with underscores

package,  
myPackage

Above are some common naming conventions that are useful to beautify the Python code:

Name: Shah Kunjan Bhat Class: FYIT Roll No: 75

## 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.

## Indentation

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 used four space per Indentation.

### Example:

$x \leq 5$

If  $x == 5$ :

Print('x is larger than 5')

In above Example, the Indented 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 as condition trigger.

Name : Shah Kunjan Bhagat Class : FYIT Roll No. 75

## Tabs vs Space

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

## Indentation following the Breaks

It is essential to use Indentation when using line continuations to keep lines to fewer than 79 characters. It provides the flexibility to determine 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)`

Name: Shah Kunjan Bharat Class: FYIT Roll No: 75

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

```
def function-name(  
    argument-one, argument-two, argument-three,  
    argument-four):  
    print(argument-two.)
```

4 Space Indentation to add a level.

```
for= long-function-name(  
    var-one, var-two,  
    var-three, var-four)
```

Use docstring

Python provides the two types of document strings as docstring - Single line and multiple lines. We use multi-line quotes - Basically, there are used to describe the function or particular program. let's understand with Example

Example def add(a,b)

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

a

Name: Shaikunjar Bhatat (Class: F.Y.I.T) Roll No: 75

Simple add program to add  
the two numbers " " "

Should a line break Before or After a binary  
operator?

The line break before or after a binary operation  
operation is a traditional approach. But it  
affects the readability extensively because the  
operators are scattered across the different  
screen, 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 use to break line before or after a  
binary operator, as long as the convention is  
consistent locally.

Name: Shah Kunjan Bharat Class: FYIT Roll No: 25

## Importing module

We should Import the modules in the separate lines as follows

Eg:-

Import pgzip

Import os

Import sys

Instead of following Example we can write the Import statement as

Eg: Import sys, os

We can also the following approach

Eg: from subprocess Import Popen, PIPE

The Import Statement should be written at the top of the file as just after any module (and) Absolute Imports are recommended because they are more readable and tend to be better schemed.

Import myPKG.sibling

from myPKG Import sibling

From myPKG.Sibling Import Example.

Name: Shah Krunjan Bhavat Class: PYIT Roll No: 75

R6

However, we can use the explicit relative imports instead of absolute imports, especially dealing with complex packages.

#### \* Blank lines.

Blank lines can be improved the readability of Python code. If my lines of code combined 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 scroll more than necessary.

Instructions to add vertical whitespace.

1) Top level function and classes with two lines.

CODE: class firstClass:  
    pass

class secondClass:  
    pass

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

Name: Shah Rukhan Bhavat Class: XI P Roll No: 275

Single blank lines inside classes

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

Example: Class first class:

```
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 before the return statement. So we can add the blank line between each step.

Eg. def Cal\_Variance(n-list):

```
list - num = 0
```

```
for n in n-list
```

```
list.sum = list.num + n
```

```
mean = list.sum / len(n-list)
```

Name: Shah Kunjan Bhavat Class: FYIT Roll No: 75

Square-Sum = 0

for n in n-list:

Square-Sum = Square-Sum + n\*\*2

mean-Square = Square-Sum / len(n-list)

between mean-Square = mean\*\*2

## \* Putting the Closing Braces

We can break lines inside Parenthesis, brackets  
using the line Continuations. PEP 8 allows us to  
use closing brace in Implicit line Continuations

Example line up the closing brace with the first non-white  
space

list-number = [

5, 4, 1

4, 6, 3

7, 8, 9

]

Name: Shah Krunjan Bharat Class: FYIT Roll No. 175

\* Line up the closing for  
Comments

Comments are the Integral Part of any Programming Language. There are the best way to understand the Code. But we should remember the following Points:

- 1] Start with the Capital letters, and write complete sentence.
- 2] Update the Comment in Case of a Change in Code.
- 3] Limit the line length of Comment and ~~docstrings~~ to 72 Characters.

### Block Comment.

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

Name: Shash Kunjan Bharat Class: FYIT Roll No: 75

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] Separate levels using the single #.

Example for i in range(0,5):

# Loop will iterate over j five times and  
print out the value of i  
# New line character  
print(i, 'In')

### Inline Comment

Inline comments are used to explain the single statement in a piece of code. We can quickly get the idea of why we wrote the particular lines of code. PEP8 specifies the following rules for the inline comment.

- 1] Start comment with the # and single space.
- 2] Use inline comment carefully.
- 3] We should separate the inline comments on the same line as the statement they refer.

Name: Shah Krunjan Bharat Class: FYIT Roll No: 75

### 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 Deconta' # this is a student`

We can use the following the name convention

Student name: 'Peter Deconta'

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

Avoid unnecessary adding whitespace. In some cases, use of whitespace can make the code much harder to read. To much whitespace can make code overly spaced and difficult to understand, we should avoid adding whitespace at the end of a line. This is known as trailing whitespace.

Name: Shah Krunjan Bharat Class: FYIT Roll No: 75

Example: # Recommended

list1 = [1, 2, 3]

# Not recommended

list1 = [1, 2, 3]

## Programming Recommendation

As we know that, there are several methods to perform similar tasks in Python. In this section we will see some of the suggestions of PEP 8 to improve the consistency.

Avoid comparing Boolean values using the equivalence operators

Example # Not recommended

fixed\_value = 10 > 5

If fixed\_value == True:

return '10 is bigger than 5'

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

Example If my feel:

return '10 is bigger than 5'.