

## **PRACTICAL 7**

### **IMPLEMENTING CODING PRACTICES IN PYTHON USING PEP8**

#### **INTRODUCTION :**

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small- and largescale projects.

#### **Why PEP 8 is Important?**

PEP 8 enhances the readability of the Python code, but why is readability so important? Let's understand this concept. Creator of Python, Guido van Rossum 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 rewrite 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.

#### **FUNCTION OF PEP8 IN PYTHON :**

##### **Syntax and semantics :**

Python is meant to be an easily readable language. Its formatting is visually uncluttered, and often uses English keywords where other languages use punctuation. Unlike many other languages, it does not use curly brackets to delimit blocks, and semicolons after statements are allowed but rarely used. It has fewer syntactic exceptions and special cases than C or Pascal.

##### **Naming Conventions :**

When you write Python code, you have to name a lot of things: variables, functions, classes, packages, and so on. Choosing sensible names will save you time and energy later. You'll be able to figure out, from the name, what a certain variable, function, or class represents. You'll also avoid using inappropriate names that might result in errors that are difficult to debug.

## Naming Styles :

The table below outlines some of the common naming styles in Python code and when you should use them:


Type	Naming Convention	Examples
Function	Use a lowercase word or words. Separate words by underscores to improve readability.	function, my_function
Variable	Use a lowercase single letter, word, or words. Separate words with underscores to improve readability.	x, var, my_variable
Class	Start each word with a capital letter. Do not separate words with underscores. This style is called camel case.	Model, MyClass
Method	Use a lowercase word or words. Separate words with underscores to improve readability.	class_method, method
Constant	Use an uppercase single letter, word, or words. Separate words with underscores to improve readability.	CONSTANT, MY_CONSTANT, MY_LONG_CONSTANT
Module	Use a short, lowercase word or words. Separate words with underscores to improve readability.	module.py, my_module.py
Package	Use a short, lowercase word or words. Do not separate words with underscores.	package, mypackag “Beautiful is better than ugly.” — The Zen of Python e

These are some of the common naming conventions and examples of how to use them. But in order to write readable code, you still have to be careful with your choice of letters and words. In addition to choosing the correct naming styles in your code, you also have to choose the names carefully. Below are a few pointers on how to do this as effectively as possible.

How to write the code name :

Choosing names for your variables, functions, classes, and so forth can be challenging. You should put a fair amount of thought into your naming choices when writing code as it will make your code more readable. The best way to name your objects in Python is to use descriptive names to make it clear what the object represents.

When naming variables, you may be tempted to choose simple, single-letter lowercase names, like `x`. But, unless you're using `x` as the argument of a mathematical function, it's not clear what `x` represents. Imagine you are storing a person's name as a string, and you want to use string slicing to format their name differently. You could end up with something like this:



```
python.py - C:/Users/DELL/AppData/Local/Programs/Python/Python39/python.py (3.9.1)
File Edit Format Run Options Window Help
# Python Program to calculate the square root

# Note: change this value for a different result
num = 8

# To take the input from the user
#num = float(input('Enter a number: '))

num_sqrt = num ** 0.5
print('The square root of %0.3f is %0.3f'%(num ,num_sqrt))

IDLE Shell 3.9.1
File Edit Shell Debug Options Window Help
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:/Users/DELL/AppData/Local/Programs/Python/Python39/python.py ====
The square root of 8.000 is 2.828
>>> |
```