PRACTICAL 7

Implementing coding practices in Python using PEP8.

As Guido van Rossum said, “Code is read much more often than it is written.” You may spend a few minutes, or a whole day, writing a piece of code to process user authentication. Once you’ve written it, you’re never going to write it again. But you’ll definitely have to read it again. That piece of code might remain part of a project you’re working on. Every time you go back to that file, you’ll have to remember what that code does and why you wrote it, so readability matters.

PEP stands for Python Enhancement Proposal, and there are several of them. A PEP is a document that describes new features proposed for Python and documents aspects of Python, like design and style, for the community.

Writing clear, readable code shows professionalism. It’ll tell an employer that you understand how to structure your code well.

If you have more experience writing Python code, then you may need to collaborate with others. Writing readable code here is crucial. Other people, who may have never met you or seen your coding style before, will have to read and understand your code. Having guidelines that you follow and recognize will make it easier for others to read your code.

Coding using PEP 8:

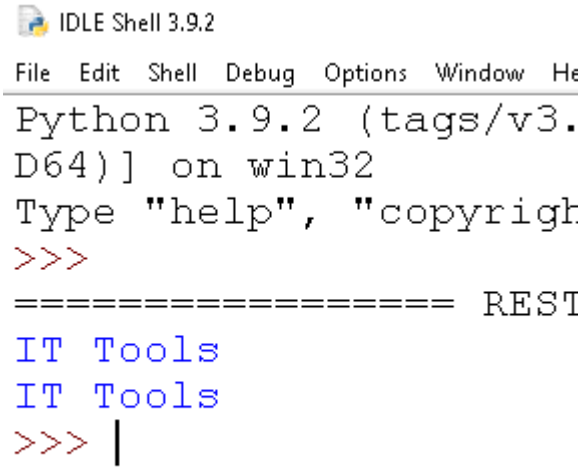
prac7.py - F:/CLG/IT TOOLS/Practical/7/prac7.py (3.9.2)

File Edit Format Run Options Window Help

```
# Correct:
# Add 4 spaces (an extra level of indentation) to distinguish arguments from the rest.
def long_function_name(
        var_one, var_two, var_three,
        var_four):
    print(var_one)

# Hanging indents should add a level.
foo = long_function_name(
    "IT Tools", "OOP",
    "WP", "PP")

# Aligned with opening delimiter.
foo = long_function_name("IT Tools", "OOP",
    "WP", "PP")
```



Another Example:

IMRAN RIZWAN SHAIKH 109 FYIT PRACTICAL 7 IT TOOLS

prac7 2.py - F:/CLG/IT TOOLS/Practical/7/prac7 2.py (3.9.2)

File Edit Format Run Options Window Help

```
# This program adds two numbers

num1 = 1.5
num2 = 6.3

# Add two numbers
sum = num1 + num2

# Display the sum
print('The sum of {0} and {1} is {2}'.format(num1, num2, sum))
```

IDLE Shell 3.9.2

File Edit Shell Debug Options Window Help

```
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19
D64)] on win32
Type "help", "copyright", "credits" or "l
>>>
========== RESTART: F:/CLG/IT TOOLS
The sum of 1.5 and 6.3 is 7.8
>>>
```