M.M.

RPA Master Bootcamp

Function and Scope of Variables

In programming, a function is a reusable block of code that performs a specific task or set of tasks. It helps in organizing code and making it more modular, efficient, and maintainable. Real-life examples of functions can be found in various aspects of everyday activities. Here are some examples:

```
def calculate_area_of_circle(radius):
    pi = 3.14159
    area = pi * radius * radius
    return area
```

PEP 8 is the official style guide for Python code, and it provides guidelines for writing clean, readable, and maintainable code. When writing functions in Python, you should follow these PEP 8 recommendations:

Function Names: Use lowercase letters and separate words with underscores to name functions.

Function Arguments: Separate function arguments with a space after the comma. If a function definition requires multiple lines, put each argument on a new line and align them with the opening parenthesis.

Function Documentation: Use docstrings to provide documentation for your functions. Use triple quotes for multi-line docstrings.

Function Length: Keep your functions short and focused. If a function becomes too long, consider refactoring it into smaller functions.

Whitespace in Functions: Use two blank lines to separate function definitions.

Return Statements: Use a single space after the "return" keyword.

```
# function and scope of variables
print("Testing")
numb = 2


def log_user_details(user_name, user_age,
user_role, number1, number2):
    total_value = sum_number(number1, number2)
```

```
    print("total_value", total_value)
    print("Print user details")
    print("user name ", user_name)
    print("user_age ", user_age)
    print("user_role ", user_role)
    return user_name, user_age, user_role, number1,
number2


def sum_number(number1, number2):
    print("Sum the numbers")

    total_value = number1 + number2
    # print(total_value)
    print(numb)
    return total_value


def log_num_table():
    for i in range(1, 11):
        print(numb * i)


user_nam = "Muneeb"
age = 32
role = "RPA Developer"
num1 = 10
num2 = 30


log_num_table()
name, user_age, user_role, number1, number2 =
log_user_details(user_nam, age, role, num1, num2)

print(name, user_age, user_role, number1, number2,
"name")
```

```
def file_name():
    fil_name = "abc.java"
    return fil_name


def log_file_extension(f_name):
    print("print file extension")
    print("f_name ", f_name)
    file_ext = f_name.split(".")
    print(file_ext)


name_of_file = file_name()
log_file_extension(name_of_file)
```