Overview

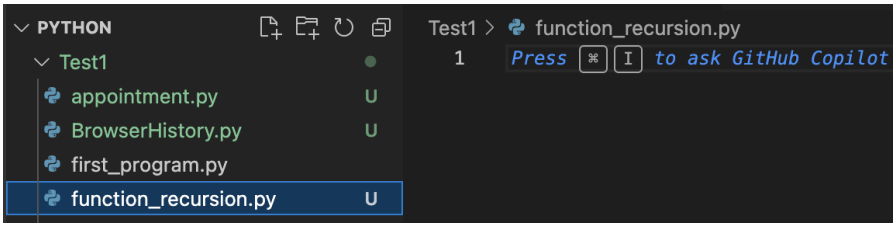
In this worksheet, we'll explore functions and function recursion in Python. Functions are reusable blocks of code that perform specific tasks, while recursion is a technique where a function calls itself repeatedly until a specific condition is met. We will learn about function recursion by solving the problem of calculating the sum of digits in a given number.

Pre-requisites:

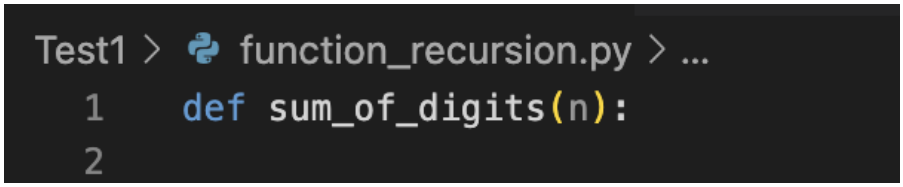
1. Github account
2. Git
3. VSCode
4. Python
5. Basic understanding of Python syntax.
6. Basic understanding of functions, call stacks and recursion
7. Familiarity with git commands

Instructions:

1. **Open the cloned folder in VSCode**.
    a. Create a python file called “function_recursion.py”



2. **Problem Statement**: Write a recursive function sum_of_digits(n) that takes a non-negative integer n as input and returns the sum of its digits.
   For example:
   sum_of_digits(123) should return 6 because 1 + 2 + 3 = 6.
   sum_of_digits(9876) should return 30 because 9 + 8 + 7 + 6 = 30.



3. **Recursive Solution:**

Test1 > function_recursion.py > ...

```python
def sum_of_digits(n):
    if n < 10:      # Base case: If n is a single-digit number, return n
        return n

    else:       # Recursive case: Calculate the sum of digits
        last_digit = n % 10         # Get the last digit of n
        remaining_digits = n // 10  # Get the remaining digits by integer division with 10

        return last_digit + sum_of_digits(remaining_digits) # Recursively call the function with the remaining digits
                                                            # and add the last digit to the result

print(sum_of_digits(123))
```

Explanation:

1. The **`sum_of_digits`** function takes a non-negative integer n as input.

2. The base case is when **n** is a single-digit number (less than 10). In this case, the function simply returns **n**.

3. For the recursive case (when n is not a single-digit number), the function performs the following steps:
   - It extracts the last digit of n using the modulo operator **%** and stores it in the variable **`last_digit`**.
   - It removes the last digit from n using integer division **//** and stores the remaining digits in the variable **`remaining_digits`**.
   - It recursively calls the **`sum_of_digits`** function with **`remaining_digits`** as the argument.
   - It adds the **`last_digit`** to the result returned by the recursive call.
   - The recursion continues until the base case is reached (when **n** becomes a single-digit number).

Exercise:
write a recursive function **`reverse_string(s)`** that reverses the order of characters in a given string.
Examples:

If the input string is "hello", the function should return "olleh".
If the input string is "python", the function should return "nohtyp".
If the input string is an empty string "", the function should return an empty string "".

The function should follow a recursive approach, meaning it should call itself with a smaller portion of the input until it reaches the base case.

Hint:

- The base case for this problem is when the input string is empty or contains only one character. In this case, the function should return the string as it is.
- For the recursive case, the function should separate the first character of the string from the remaining characters. It should then recursively call the reverse_string function with the remaining characters (excluding the first character). After getting the reversed string for the remaining characters, it should append the first character to the end of the reversed string.

Sample Input/Output:

```
hello → olleh
Python → nohtyp
```