

WEEK: 04

1 .Calculate the length of a string (using both inbuilt and without inbuilt method)

Using inbuilt method

```
def length_inbuilt(s):  
    return len(s)
```

Without using inbuilt method

```
def length_without_inbuilt(s):  
    count = 0  
    for char in s:  
        count += 1  
    return count
```

```
s = "Hello World"
```

```
print(length_inbuilt(s))
```

```
print(length_without_inbuilt(s))
```

Output:



```
bash Copy code  
11  
11
```

2 . Count the number of occurrences of a given character in a string (using inbuilt and without inbuilt method)

Using inbuilt method

```
def count_inbuilt(s, char):  
    return s.count(char)
```


Without using inbuilt method

```
def count_without_inbuilt(s, char):  
    count = 0  
    for c in s:
```

```
if c == char:  
    count += 1  
return count
```

```
s = "hello world"  
char = 'l'  
print(count_inbuilt(s, char))  
print(count_without_inbuilt(s, char))
```

OUTPUT:

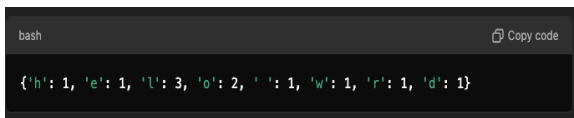
A terminal window with a dark background. The title bar says 'bash' and there is a 'Copy code' button. The output shows the number '3' on two separate lines.

```
bash Copy code  
3  
3
```

3. Count the number of occurrences of each character in a string:

```
from collections import Counter  
  
def count_each_char(s):  
    return dict(Counter(s))  
  
s = "hello world"  
print(count_each_char(s))
```

OUTPUT:

A terminal window with a dark background. The title bar says 'bash' and there is a 'Copy code' button. The output is a dictionary showing the frequency of each character in the string 'hello world'.

```
bash Copy code  
{'h': 1, 'e': 1, 'l': 3, 'o': 2, ' ': 1, 'w': 1, 'r': 1, 'd': 1}
```

4. Create a new string by appending s2 in the middle of s1:

```
def append_in_middle(s1, s2):  
    middle_index = len(s1) // 2  
    return s1[:middle_index] + s2 + s1[middle_index:]  
  
s1 = "hello"  
print(append_in_middle(s1, s2))
```

OUTPUT:

```
bash Copy code
helloworld
```

5. Count all letters, digits, and special symbols from a given string:

```
def count_characters(s):
    letters = digits = special = 0
    for char in s:
        if char.isalpha():
            letters += 1
        elif char.isdigit():
            digits += 1
        else:
            special += 1
    return letters, digits, special

s = "Hello123@#"
letters, digits, special = count_characters(s)
print(f"Letters: {letters}, Digits: {digits}, Special: {special}")
```

OUTPUT:

```
bash Copy code
Letters: 5, Digits: 3, Special: 2
```

6. Check if two strings are balanced:

```
def are_balanced(s1, s2):
    return all(char in s2 for char in s1)

s1 = "abc"
s2 = "aabbcc"
print(are_balanced(s1, s2))
```

OUTPUT:

```
bash Copy code
True
```

7. Find all occurrences of a substring in a given string by ignoring the case:

```
def find_substring(s, sub):
```

```
    s_lower = s.lower()
```

```
    sub_lower = sub.lower()
```

```
    return [i for i in range(len(s)) if s_lower.startswith(sub_lower, i)]
```

```
s = "Hello hello HeLlO"
```

```
sub = "hello"
```

```
print(find_substring(s, sub))
```

OUTPUT:

```
bash Copy code
[0, 6, 12]
```

8. Calculate the sum and average of the digits present in a string:

```
def sum_and_average(s):
```

```
    digits = [int(char) for char in s if char.isdigit()]
```

```
    total_sum = sum(digits)
```

```
    average = total_sum / len(digits) if digits else 0
```

```
    return total_sum, average
```

```
s = "abc1234xyz"
```

```
total, avg = sum_and_average(s)
```

```
print(f"Sum: {total}, Average: {avg}")
```

OUTPUT:

```
bash Copy code
Sum: 10, Average: 2.5
```

9. Reverse a string using slicing operator:

```
def reverse_string_slicing(s):
    return s[::-1]
```

```
s = "Hello"
print(reverse_string_slicing(s))
```

OUTPUT:

```
bash Copy code
oLleH
```

10. Reverse a string using loops:

```
def reverse_string_loop(s):
    reversed_string = ""
    for char in s:
        reversed_string = char + reversed_string
    return reversed_string
```

```
s = "Hello"
print(reverse_string_loop(s))
```

OUTPUT:

```
bash Copy code
oLleH
```

11. Remove all characters from a string except integers:

```
def remove_non_digits(s):
```

```
return ''.join(char for char in s if char.isdigit())
```

```
s = "abc123xyz"
```

```
print(remove_non_digits(s))
```

OUTPUT:

A screenshot of a terminal window with a dark background. The top bar shows 'bash' on the left and a 'Copy code' button on the right. The terminal displays the output '123' on a new line.

```
bash Copy code
123
```

12. Apply all possible operators (+, *, in, is, ==, >):

```
s1 = "Hello"
```

```
s2 = "World"
```

Concatenation

```
print(s1 + s2)
```

Repetition

```
print(s1 * 3)
```

Membership

```
print('H' in s1)
```

Identity

```
print(s1 is s2)
```

Equality

```
print(s1 == s2)
```

Comparison

```
print(s1 > s2)
```

OUTPUT:

```
bash Copy code
HelloWorld
HelloHelloHello
True
False
False
True
```

13. Create a function with two arguments (one of them is a default argument):

```
def my_function(a, b=10):
```

```
    return a + b
```

```
print(my_function(5))
```

```
print(my_function(5, 15))
```

OUTPUT:

```
bash Copy code
15
20
```

14. Reverse a string using recursion:

```
def reverse_string_recursion(s):
```

```
    if len(s) == 0:
```

```
        return s
```

```
    else:
```

```
        return reverse_string_recursion(s[1:]) + s[0]
```

```
s = "Hello"
```

```
print(reverse_string_recursion(s))
```

OUTPUT:

```
bash Copy code
olleH
```