Day 2: Strings in Python

Topics Covered:

- *Introduction to Strings*
- *String Operations*
- *String Methods*

1. Introduction to Strings:

In Python, strings are sequences of characters enclosed in single, double, or triple quotes. Strings are one of the most common data types used to handle textual data.

Example of a string:

```
str1 = "Hello, World!"
print(str1)
```

Characteristics of Strings:

- *Strings are immutable: once created, their content cannot be changed.*
- *You can create strings using single quotes ('), double quotes (") or triple quotes (''' or """).*

Example of a Multi-line String:

```
str2 = """This is a
multi-line string"""
```

2. String Operations:

Strings support many operations such as concatenation, repetition, and slicing.

Concatenation:

```
str3 = "Hello" + " " + "World!"
print(str3)  # Output: Hello World!
```

Repetition:

```
str4 = "Hello" * 3
print(str4)  # Output: HelloHelloHello
```

Slicing:

```
str5 = "Hello, World!"
print(str5[0:5])  # Output: Hello
```

3. String Methods:

Python provides several built-in methods for strings that help in manipulation and analysis.

len():

```
str6 = "Hello, World!"
print(len(str6))  # Output: 13
```

lower():

```
print(str6.lower())  # Output: hello, world!
```

upper():

```
print(str6.upper())  # Output: HELLO, WORLD!
```

replace():

```
print(str6.replace("World", "Python"))  # Output: Hello, Python!
```

strip():

```
str7 = "  Hello, World!  "
print(str7.strip())  # Output: Hello, World!
```

Summary of Day 2:

Today, we learned about strings in Python. We explored string operations such as concatenation, repetition, and slicing. Additionally, we covered common string methods like lower(), upper(), replace(), and strip. Mastering strings is crucial as they are used frequently in many programming tasks, from text manipulation to data processing.