Chapter 3 – Strings

What is a String?

A string is a data type in Python used to represent a sequence of characters enclosed in quotes. You can define a string in Python using single quotes, double quotes, or triple quotes.

Examples:

```
a = 'shivam'       # Single quoted string
b = "shivam"       # Double quoted string
c = '''shivam'''   # Triple quoted string
```

String Slicing

String slicing allows you to extract a portion of a string. The syntax for slicing is:

```
string[start:stop:step]
```

`start`: The index from where the slice begins (inclusive).
`stop`: The index where the slice ends (exclusive).
`step`: The number of characters to skip.

Examples:

```
word = "amazing"
print(word[1:6])     # Output: "mazi" (from index 1 to 5)
print(word[1:6:2])   # Output: "mzn" (from index 1 to 5, skipping every 2nd character)
print(word[:7])      # Output: "amazing" (from the start to index 6)
print(word[0:])      # Output: "amazing" (from index 0 to the end)
```

String Functions

Python provides several built-in functions to perform operations on strings. Consider the following string:

```
str = "shivam"
```

Common String Functions:

1. `len()` **Function**: Returns the length of the string.

```
print(len(str))  # Output: 6
```

2. `endswith()` **Method**: Checks if the string ends with the specified suffix.

```
print(str.endswith("vam"))  # Output: True
```

3. `count()` **Method**: Counts the occurrences of a specified character or substring.

```
print(str.count("a"))  # Output: 2
```

4. `capitalize()` **Method**: Capitalizes the first character of the string.

```
capitalized_string = str.capitalize()
print(capitalized_string)  # Output: "Shivam"
```

5. `find()` **Method**: Finds the index of the first occurrence of a substring.

```
index = str.find("va")
print(index)  # Output: 3
```

6. `replace()` **Method**: Replaces occurrences of a substring with a new substring.

```
replaced_string = str.replace("a", "o")
print(replaced_string)  # Output: "shivom"
```

Escape Sequence Characters

Escape sequence characters are used to represent special characters in strings. They begin with a backslash (`\`).

Common Escape Sequences:

- `\\`: Backslash
- `\'`: Single quote
- `\"`: Double quote
- `\n`: Newline
- `\t`: Tab

Examples:

```
escaped_string = "This is a line break:\nThis is a new line."
print(escaped_string)
```