Introduction To Python Strings

- What is a string?

➢ A string is a sequence of characters enclosed within single (‘ ’), double (“ ”), or triple quotes (‘’’ ’’’ or “”” ”””).

➢ In Python, strings are immutable, meaning they are can not be changed after they are created.

- Example:

➢ text = "Hello, World!”

Creating Strings In Python

- Single and Double Quotes:
  - Both can be used to create strings in python.
  - Example: str1 = 'Hello'

    str2 = "World"
- Triple Quotes:
  - Used for multiline strings or docstrings
  - Example:
  - Input: multiline_str = '''This is

    a multiline string.'''
  - Output: This is

    a multiline string.

Strings Indexing And Slicing

- String Indexing:
  - Access individual characters using index positions (starting from 0).
  - Example:
  - Input:
  - text = "Python"

    Print(text[0])
  - Output:
  - P
- String Slicing:
  - Extract part of string using start : stop : step .
  - Example:
  - Input:- print(text[0:3]) # Output:- Pyt
  - Input:- print(text[::2]) # Output:- Pto

String Methods

- Common String Methods:
  - Len() : Returns the length of string.
  - Lower() : Converts all character to lower character.
  - Upper() : Converts all character to upper character.
  - Strip() : Remove whitespace from both ends.
  - Replace() : Replace a substring with another
  - Example:
  - my_string = " Hello World "
  - Input: print(my_string.strip()) # Output: "Hello World"
  - Input: print(my_string.upper()) # Output: "HELLO WORLD"

String Concatenation And Formatting

- Concatenation:
  - Use + to combine strings.
  - Example:
  - Input: greeting = “Hello” + “ “ + “World” # Output: Hello World
- String Formatting:
  - f-strings: Easier way to embed expressions inside string literals.
  - Example:
  - Input: name = “John”

    print(name) # Output: John
  - Input: print(f”Hello, {name}!”) # Output: Hello, John!

Escape Sequences

- Special characters in strings: Backslash is used to introduce escape sequences.
  - **\n** : New line
  - **\t** : Tab
  - **** : Backslash
  - Example:
  - Input: print("Hello\nWorld") # Output: Hello (New line) World
  - Input: print("Hello\tWorld") # Output: Hello World
  - Input: print("Hello\\World") # Output: Hello\World

Strings Are Immutable

- Immutability:
  - Strings cannot be changed after creation. Any operation on string creates a new string.
  - Example:
  - Input: text = "Hello"

```
text[0] = "J"        # This will raise an error
```

Advanced String Operations

- Joining a List of Strings:
  - Use join() to combine list element into a single string.
  - Example:
  - Input: words = "Hello", "World"

```
sentence = " ".join(words)     # Output: "Hello World"
```

- Splitting a String:
  - Use split() to break a string into a list.
  - Example:
  - Input: sentence = "Hello World"

```
words = sentence.split()       # Output: ['Hello', 'World']
```

Strings Validation Methods

- Common validation maethods:
  - isalpha() : Checks is all characters are alphabetic.
  - Isdigit() : Checks if all characters are digits.
  - Isalnum() : Checks If all characters are alphanumeric.
  - Example:
  - Input: print("Hello World".isalpha()) # Output: True
  - Input: print("12345".isdigit()) # Output: True
  - Input: print("Hello123.isalnum()) # Output: True

Conclusion

- Key Takeaways:
  - Strings are a fundamental data type in python.
  - They are immutable and support various operations and methods for manipulation.
  - String formatting, slicing, and advanced operations like joining and splitting are useful in many applications.