String Data Type in Python

Strings are one of the most commonly used data types in Python. They are sequences of characters enclosed in quotes. Python supports single, double, and triple quotes for string creation.

Creating Strings

```
# Single-quoted string
str1 = 'Hello, World!'

# Double-quoted string
str2 = "Hello, World!"

# Triple-quoted string (used for multi-line strings)
str3 = """Hello,
World!"""

print(str1)
print(str2)
print(str3)
```

String Indexing and Slicing

Strings can be indexed and sliced to access specific characters or substrings.

```
str1 = 'Hello, World!'

# Indexing
print(str1[0])   # Output: H
print(str1[-1])  # Output: !

# Slicing
print(str1[0:5])  # Output: Hello
print(str1[7:])   # Output: World!
print(str1[:5])   # Output: Hello
print(str1[::2])  # Output: Hlo ol!
```

Common String Methods

1. len()

Returns the length of the string.

```
str1 = 'Hello, World!'
print(len(str1))  # Output: 13
```

2. str.lower()

Converts all characters in the string to lowercase.

```python
str1 = 'Hello, World!'
print(str1.lower())  # Output: hello, world!
```

3. str.upper()

Converts all characters in the string to uppercase.

```python
str1 = 'Hello, World!'
print(str1.upper())  # Output: HELLO, WORLD!
```

4. str.capitalize()

Capitalizes the first character of the string.

```python
str1 = 'hello, world!'
print(str1.capitalize())  # Output: Hello, world!
```

5. str.title()

Converts the first character of each word to uppercase.

```python
str1 = 'hello, world!'
print(str1.title())  # Output: Hello, World!
```

6. str.strip()

Removes leading and trailing whitespace.

```python
str1 = '   Hello, World!   '
print(str1.strip())  # Output: Hello, World!
```

7. str.replace(old, new)

Replaces all occurrences of a substring with another substring.

```python
str1 = 'Hello, World!'
print(str1.replace('World', 'Python'))  # Output: Hello, Python!
```

8. str.split(separator)

Splits the string into a list of substrings based on a specified separator.

```python
str1 = 'Hello, World!'
print(str1.split(', '))  # Output: ['Hello', 'World!']
```

9. str.join(iterable)

Joins the elements of an iterable (e.g., list) into a single string, separated by the string calling the method.

```
list1 = ['Hello', 'World']
print(' '.join(list1))  # Output: Hello World
```

10. str.find(substring)

Returns the index of the first occurrence of the substring, or -1 if not found.

```
str1 = 'Hello, World!'
print(str1.find('World'))  # Output: 7
```

11. str.count(substring)

Returns the number of occurrences of the substring.

```
str1 = 'Hello, World!'
print(str1.count('o'))  # Output: 2
```

12. str.startswith(prefix)

Returns `True` if the string starts with the specified prefix, otherwise `False`.

```
str1 = 'Hello, World!'
print(str1.startswith('Hello'))  # Output: True
```

13. str.endswith(suffix)

Returns `True` if the string ends with the specified suffix, otherwise `False`.

```
str1 = 'Hello, World!'
print(str1.endswith('World!'))  # Output: True
```

14. str.isalpha()

Returns `True` if all characters in the string are alphabetic, otherwise `False`.

```
str1 = 'Hello'
print(str1.isalpha())  # Output: True
```

15. str.isdigit()

Returns `True` if all characters in the string are digits, otherwise `False`.

```
str1 = '12345'
print(str1.isdigit())  # Output: True
```

16. str.isalnum()

Returns `True` if all characters in the string are alphanumeric, otherwise `False`.

```
str1 = 'Hello123'
print(str1.isalnum())  # Output: True
```

String Formatting

1. Using f-strings

Introduced in Python 3.6, f-strings are a concise and convenient way to embed expressions inside string literals.

```
name = 'Alice'
age = 30
print(f'My name is {name} and I am {age} years old.')
```

2. Using str.format()

The `str.format()` method allows you to format strings with placeholders.

```
name = 'Bob'
age = 25
print('My name is {} and I am {} years old.'.format(name, age))
```

3. Using % Operator

The `%` operator can be used for string formatting similar to C-style string formatting.

```
name = 'Charlie'
age = 35
print('My name is %s and I am %d years old.' % (name, age))
```

Stay Updated

Be sure to this repository to stay updated with new examples and enhancements!

License

Contact

Panagiotis Moschos - pan.moschos86@gmail.com

Note: This is a Python script and requires a Python interpreter to run.

Happy Coding

Made with by Panagiotis Moschos (https://github.com/pmoschos)