

String Methods

Python provides a variety of built-in methods for manipulating strings. These methods allow you to perform common operations such as searching, modifying, and formatting strings. Here are some essential string methods in Python:

1. `len()`

- Returns the length of the string (the number of characters).

```
```python
my_string = "Hello, World!"
length = len(my_string) # Result: 13
```
```

2. `lower()` and `upper()`

- `lower()`: Converts all characters in the string to lowercase.
- `upper()`: Converts all characters in the string to uppercase.

```
```python
my_string = "Hello, World!"
lowercase = my_string.lower() # Result: "hello, world!"
uppercase = my_string.upper() # Result: "HELLO, WORLD!"
```
```

3. `strip()`

- Removes leading and trailing whitespace (including spaces, tabs, and newline characters).

```
```python
my_string = " Hello, World! "
stripped_string = my_string.strip() # Result: "Hello, World!"
```
```

4. `replace(old, new)`

- Replaces occurrences of the old substring with the new substring.

```
```python
my_string = "Hello, World!"
new_string = my_string.replace("Hello", "Hi") # Result: "Hi, World!"
```
```

5. `find(substring)`

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

```
```python
my_string = "Hello, World!"
index = my_string.find("World") # Result: 7
```
```

6. ****count(substring)****

- Returns the number of occurrences of the substring in the string.

```
```python
my_string = "abcaabcba"
count = my_string.count("a") # Result: 4
```
```

7. ****startswith(prefix) and endswith(suffix)****

- `startswith(prefix)`: Checks if the string starts with the specified prefix.
- `endswith(suffix)`: Checks if the string ends with the specified suffix.

```
```python
my_string = "Hello, World!"
starts_with_hello = my_string.startswith("Hello") # Result: True
ends_with_world = my_string.endswith("World!") # Result: True
```
```

8. ****split(separator)****

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

```
```python
my_string = "apple,orange,banana"
fruits_list = my_string.split(",") # Result: ['apple', 'orange', 'banana']
```
```

9. ****join(iterable)****

- Joins the elements of an iterable (e.g., a list) into a single string, using the string as a separator.

```
```python
fruits = ['apple', 'orange', 'banana']
joined_string = ",".join(fruits) # Result: 'apple,orange,banana'
```
```

10. ****format()****

- Formats a string by replacing placeholders with values.

```
```python
name = "Alice"
age = 30
formatted_string = "My name is {} and I am {} years old.".format(name, age)
Result: "My name is Alice and I am 30 years old."
```
```

11. ****startswith(prefix) and endswith(suffix)****

- `startswith(prefix)`: Checks if the string starts with the specified prefix.
- `endswith(suffix)`: Checks if the string ends with the specified suffix.

```
```python
my_string = "Hello, World!"
starts_with_hello = my_string.startswith("Hello") # Result: True
ends_with_world = my_string.endswith("World!") # Result: True
```
```

These are just a few examples of the many string methods available in Python. Understanding and using these methods will make it easier to work with strings in a variety of applications.