# Python String Methods

## 1. `str.capitalize()`

- **Purpose**: Converts the first character of the string to uppercase and the rest to lowercase.
- **Syntax**: `string.capitalize()`
- **Example**:

```
text = "hello world"
print(text.capitalize())  # Output: "Hello world"
```

---

## 2. `str.lower()`

- **Purpose**: Converts all characters in the string to lowercase.
- **Syntax**: `string.lower()`
- **Example**:

```
text = "HELLO WORLD"
print(text.lower())  # Output: "hello world"
```

---

## 3. `str.upper()`

- **Purpose**: Converts all characters in the string to uppercase.

- **Syntax**: `string.upper()`
- **Example**:

```
text = "hello world"
print(text.upper())  # Output: "HELLO WORLD"
```

---

## 4. `str.title()`

- **Purpose**: Converts the first character of each word to uppercase and the rest to lowercase.
- **Syntax**: `string.title()`
- **Example**:

```
text = "hello world"
print(text.title())  # Output: "Hello World"
```

---

## 5. `str.swapcase()`

- **Purpose**: Swaps the case of all characters in the string (uppercase becomes lowercase and vice versa).
- **Syntax**: `string.swapcase()`
- **Example**:

```
text = "Hello World"
print(text.swapcase())  # Output: "hELLO wORLD"
```

---

## 6. `str.strip()`

- **Purpose**: Removes leading and trailing whitespace (or specified characters) from the string.
- **Syntax**: `string.strip([chars])`
- **Example**:

```
text = "   hello world   "
print(text.strip())  # Output: "hello world"
```

## 7. str.lstrip()

- **Purpose**: Removes leading whitespace (or specified characters) from the string.
- **Syntax**: `string.lstrip([chars])`
- **Example**:

```
text = "   hello world   "
print(text.lstrip())  # Output: "hello world   "
```

## 8. str.rstrip()

- **Purpose**: Removes trailing whitespace (or specified characters) from the string.
- **Syntax**: `string.rstrip([chars])`
- **Example**:

```
text = "   hello world   "
print(text.rstrip())  # Output: "   hello world"
```

## 9. str.replace()

- **Purpose**: Replaces all occurrences of a substring with another substring.
- **Syntax**: `string.replace(old, new[, count])`
- **Example**:

```
text = "hello world"
print(text.replace("world", "Python"))  # Output: "hello Python"
```

## 10. str.split()

- **Purpose**: Splits the string into a list of substrings based on a delimiter.
- **Syntax**: `string.split([sep[, maxsplit]])`
- **Example**:

  ```
  text = "hello world"
  print(text.split())  # Output: ['hello', 'world']
  ```

---

## 11. `str.join()`

- **Purpose**: Joins elements of an iterable (e.g., list) into a single string using the string as a separator.
- **Syntax**: `string.join(iterable)`
- **Example**:

  ```
  words = ["hello", "world"]
  print(" ".join(words))  # Output: "hello world"
  ```

---

## 12. `str.find()`

- **Purpose**: Returns the lowest index of the substring if found, otherwise returns `1`.
- **Syntax**: `string.find(sub[, start[, end]])`
- **Example**:

  ```
  text = "hello world"
  print(text.find("world"))  # Output: 6
  ```

---

## 13. `str.index()`

- **Purpose**: Similar to `find()`, but raises a `ValueError` if the substring is not found.
- **Syntax**: `string.index(sub[, start[, end]])`
- **Example**:

```
text = "hello world"
print(text.index("world"))  # Output: 6
```

## 14. str.count()

- **Purpose**: Returns the number of non-overlapping occurrences of a substring in the string.
- **Syntax**: `string.count(sub[, start[, end]])`
- **Example**:

```
text = "hello world"
print(text.count("l"))  # Output: 3
```

## 15. str.startswith()

- **Purpose**: Checks if the string starts with a specified prefix.
- **Syntax**: `string.startswith(prefix[, start[, end]])`
- **Example**:

```
text = "hello world"
print(text.startswith("hello"))  # Output: True
```

## 16. str.endswith()

- **Purpose**: Checks if the string ends with a specified suffix.
- **Syntax**: `string.endswith(suffix[, start[, end]])`
- **Example**:

```
text = "hello world"
print(text.endswith("world"))  # Output: True
```

## 17. str.isalpha()

- **Purpose**: Checks if all characters in the string are alphabetic (letters).

- **Syntax**: `string.isalpha()`
- **Example**:

```
text = "hello"
print(text.isalpha())  # Output: True
```

## 18. `str.isdigit()`

- **Purpose**: Checks if all characters in the string are digits.
- **Syntax**: `string.isdigit()`
- **Example**:

```
text = "123"
print(text.isdigit())  # Output: True
```

## 19. `str.isalnum()`

- **Purpose**: Checks if all characters in the string are alphanumeric (letters or digits).
- **Syntax**: `string.isalnum()`
- **Example**:

```
text = "hello123"
print(text.isalnum())  # Output: True
```

## 20. `str.islower()`

- **Purpose**: Checks if all characters in the string are lowercase.
- **Syntax**: `string.islower()`
- **Example**:

```
text = "hello"
print(text.islower())  # Output: True
```

## 21. `str.isupper()`

- **Purpose**: Checks if all characters in the string are uppercase.
- **Syntax**: `string.isupper()`
- **Example**:

```
text = "HELLO"
print(text.isupper())  # Output: True
```

---

## 22. `str.isspace()`

- **Purpose**: Checks if all characters in the string are whitespace.
- **Syntax**: `string.isspace()`
- **Example**:

```
text = "   "
print(text.isspace())  # Output: True
```

---

## 23. `str.zfill()`

- **Purpose**: Pads the string with zeros on the left until it reaches the specified length.
- **Syntax**: `string.zfill(width)`
- **Example**:

```
text = "42"
print(text.zfill(5))  # Output: "00042"
```

---

## 24. `str.format()`

- **Purpose**: Formats the string by replacing placeholders `{}` with specified values.
- **Syntax**: `string.format(*args, **kwargs)`
- **Example**:

```
text = "Hello, {}!"
print(text.format("world"))  # Output: "Hello, world!"
```

## 25. `str.center()`

- **Purpose**: Centers the string in a field of a specified width.
- **Syntax**: `string.center(width[, fillchar])`
- **Example**:

```
text = "hello"
print(text.center(10, "-"))  # Output: "--hello---"
```

## 26. `str.ljust()`

- **Purpose**: Left-justifies the string in a field of a specified width.
- **Syntax**: `string.ljust(width[, fillchar])`
- **Example**:

```
text = "hello"
print(text.ljust(10, "-"))  # Output: "hello-----"
```

## 27. `str.rjust()`

- **Purpose**: Right-justifies the string in a field of a specified width.
- **Syntax**: `string.rjust(width[, fillchar])`
- **Example**:

```
text = "hello"
print(text.rjust(10, "-"))  # Output: "-----hello"
```

## 28. `str.expandtabs()`

- **Purpose**: Replaces tab characters (`\t`) with spaces.

- **Syntax**: `string.expandtabs(tabsize=8)`
- **Example**:

```
text = "hello\tworld"
print(text.expandtabs(4))  # Output: "hello   world"
```

## 29. `str.encode()`

- **Purpose**: Encodes the string into bytes using a specified encoding (default is `utf-8`).
- **Syntax**: `string.encode(encoding="utf-8", errors="strict")`
- **Example**:

```
text = "hello"
print(text.encode())  # Output: b'hello'
```

## 30. `str.translate()`

- **Purpose**: Translates the string using a translation table (created with `str.maketrans()`).
- **Syntax**: `string.translate(table)`
- **Example**:

```
text = "hello"
table = str.maketrans("el", "EL")
print(text.translate(table))  # Output: "hELLo"
```