## Python String Methods: Definitions, Purpose, Syntax, and Examples

**1. str.lower()**

**Definition:** Converts all characters in the string to lowercase. **Purpose:** To make a string lowercase, often for case-insensitive comparisons. **Syntax:**

string.lower()

**Example:**

"HELLO".lower()  # Output: "hello"

---

**2. str.upper()**

**Definition:** Converts all characters in the string to uppercase. **Purpose:** To make a string uppercase. **Syntax:**

string.upper()

**Example:**

"hello".upper()  # Output: "HELLO"

---

**3. str.capitalize()**

**Definition:** Capitalizes the first character of the string. **Purpose:** Formats the string for readability. **Syntax:**

string.capitalize()

**Example:**

"python".capitalize()  # Output: "Python"

---

**4. str.title()**

**Definition:** Capitalizes the first character of each word. **Purpose:** Formats a string to title case. **Syntax:**

string.title()

**Example:**

"hello world".title()  # Output: "Hello World"

---

**5. str.swapcase()**

**Definition:** Swaps the case of each character. **Purpose:** Converts lowercase characters to uppercase and vice versa. **Syntax:**

string.swapcase()

**Example:**

"Hello".swapcase()  # Output: "hELLO"

---

**6. str.find(sub)**

**Definition:** Returns the lowest index of the substring. Returns -1 if not found. **Purpose:** To locate a substring within a string. **Syntax:**

string.find(substring)

**Example:**

"hello world".find("world")  # Output: 6

---

**7. str.index(sub)**

**Definition:** Like find(), but raises a ValueError if the substring is not found. **Purpose:** Locate a substring and ensure it exists. **Syntax:**

string.index(substring)

**Example:**

"hello world".index("world")  # Output: 6

---

**8. str.startswith(prefix)**

**Definition:** Checks if the string starts with the given prefix. **Purpose:** To validate prefixes. **Syntax:**

string.startswith(prefix)

**Example:**

"hello".startswith("he")  # Output: True

---

**9. str.endswith(suffix)**

**Definition:** Checks if the string ends with the given suffix. **Purpose:** To validate suffixes. **Syntax:**

string.endswith(suffix)

**Example:**

"hello".endswith("lo")  # Output: True

---

**10. str.count(sub)**

**Definition:** Counts the occurrences of a substring. **Purpose:** To count how many times a substring appears in a string. **Syntax:**

string.count(substring)

**Example:**

"banana".count("a")  # Output: 3

---

**11. str.replace(old, new)**

**Definition:** Replaces occurrences of a substring with another. **Purpose:** To modify strings by replacing content. **Syntax:**

string.replace(old, new)

**Example:**

"hello world".replace("world", "Python")  # Output: "hello Python"

---

**12. str.strip([chars])**

**Definition:** Removes leading and trailing whitespace (or specified characters). **Purpose:** Cleans up strings by removing unwanted characters or spaces. **Syntax:**

string.strip([chars])

**Example:**

"  hello  ".strip()  # Output: "hello"

---

**13. str.lstrip([chars])**

**Definition:** Removes leading whitespace (or specified characters). **Purpose:** Cleans up strings by removing unwanted leading characters. **Syntax:**

string.lstrip([chars])

**Example:**

"  hello".lstrip()  # Output: "hello"

---

**14. str.rstrip([chars])**

**Definition:** Removes trailing whitespace (or specified characters). **Purpose:** Cleans up strings by removing unwanted trailing characters. **Syntax:**

string.rstrip([chars])

**Example:**

"hello  ".rstrip()  # Output: "hello"

---

**15. str.split(sep)**

**Definition:** Splits the string into a list by the specified separator. **Purpose:** To break a string into parts. **Syntax:**

string.split(separator)

**Example:**

"a,b,c".split(",")  # Output: ['a', 'b', 'c']

---

**16. str.join(iterable)**

**Definition:** Joins elements of an iterable into a single string with a specified separator. **Purpose:** To combine multiple strings into one. **Syntax:**

separator.join(iterable)

**Example:**

" ".join(["a", "b", "c"])  # Output: "a b c"

---

**17. str.isalpha()**

**Definition:** Returns True if all characters are alphabetic. **Purpose:** Validates if a string contains only letters. **Syntax:**

string.isalpha()

**Example:**

"abc".isalpha()  # Output: True

---

**18. str.isdigit()**

**Definition:** Returns True if all characters are digits. **Purpose:** Validates if a string contains only numbers. **Syntax:**

string.isdigit()

**Example:**

"123".isdigit()  # Output: True

---

**19. str.isalnum()**

**Definition:** Returns True if all characters are alphanumeric. **Purpose:** Validates if a string contains only letters and numbers. **Syntax:**

string.isalnum()

**Example:**

"abc123".isalnum()  # Output: True

---

**20. str.isspace()**

**Definition:** Returns True if all characters are whitespace. **Purpose:** Checks for whitespace-only strings. **Syntax:**

string.isspace()

**Example:**

"   ".isspace()  # Output: True

---

**21. str.islower()**

**Definition:** Checks if all characters are lowercase. **Purpose:** Validates lowercase-only strings. **Syntax:**

string.islower()

**Example:**

"hello".islower()  # Output: True

---

**22. str.isupper()**

**Definition:** Checks if all characters are uppercase. **Purpose:** Validates uppercase-only strings. **Syntax:**

string.isupper()

**Example:**

"HELLO".isupper()  # Output: True

---

**23. str.istitle()**

**Definition:** Checks if the string is title-cased. **Purpose:** Validates if each word's first character is uppercase. **Syntax:**

string.istitle()

**Example:**

"Hello World".istitle()  # Output: True

27. `str.zfill(width)`

Pads the string with zeros on the left to fill the specified width.

```python
"42".zfill(5)  # Output: "00042"
```

---