[Python String Methods] (CheatSheet)

1. Case Conversion

- **lower**: str.lower()
- **upper**: str.upper()
- **capitalize**: str.capitalize()
- **title**: str.title()
- **swapcase**: str.swapcase()
- **casefold**: str.casefold()

2. Checking Content

- **isalpha**: str.isalpha()
- **isdigit**: str.isdigit()
- **isnumeric**: str.isnumeric()
- **isalnum**: str.isalnum()
- **isspace**: str.isspace()
- **istitle**: str.istitle()
- **islower**: str.islower()
- **isupper**: str.isupper()
- **isdecimal**: str.isdecimal()
- **isidentifier**: str.isidentifier()
- **isprintable**: str.isprintable()

3. Searching and Replacing

- **startswith**: str.startswith(substring)
- **endswith**: str.endswith(substring)
- **count**: str.count(substring)
- **find**: str.find(substring)
- **index**: str.index(substring)
- **rfind**: str.rfind(substring)
- **rindex**: str.rindex(substring)
- **replace**: str.replace(old, new[, count])

By: Waleed Mousa

4. Character and Substring Manipulation

- **strip**: `str.strip([chars])`
- **rstrip**: `str.rstrip([chars])`
- **lstrip**: `str.lstrip([chars])`
- **split**: `str.split([sep[, maxsplit]])`
- **rsplit**: `str.rsplit([sep[, maxsplit]])`
- **partition**: `str.partition(sep)`
- **rpartition**: `str.rpartition(sep)`
- **join**: `separator.join(iterable)`
- **expandtabs**: `str.expandtabs(tabsize)`
- **center**: `str.center(width[, fillchar])`
- **ljust**: `str.ljust(width[, fillchar])`
- **rjust**: `str.rjust(width[, fillchar])`
- **zfill**: `str.zfill(width)`

5. Text Formatting

- **format**: `str.format(*args, **kwargs)`
- **format_map**: `str.format_map(mapping)`
- **encode**: `str.encode(encoding='utf-8', errors='strict')`
- **translate**: `str.translate(table)`

6. Escape Characters

- **Escape Single Quote**: `'Don\\'t'`
- **Escape Double Quote**: `"He said, \"Hello\""`
- **Newline**: `'Hello\\nWorld'`
- **Tab**: `'Hello\\tWorld'`
- **Backslash**: `'Use \\\\ to represent backslash'`

7. Regular Expressions

- **re.match**: `re.match(pattern, string)`
- **re.search**: `re.search(pattern, string)`
- **re.findall**: `re.findall(pattern, string)`
- **re.finditer**: `re.finditer(pattern, string)`

- **re.sub**: `re.sub(pattern, repl, string)`
- **re.compile**: `regex = re.compile(pattern)`

8. Working with Whitespace

- **Remove Leading Whitespace**: `str.lstrip()`
- **Remove Trailing Whitespace**: `str.rstrip()`
- **Remove Both Leading and Trailing Whitespace**: `str.strip()`
- **Split Lines**: `str.splitlines([keepends])`

9. String Testing

- **Check for Substring**: `'substring' in str`
- **Check for Absence of Substring**: `'substring' not in str`
- **Check String Equality**: `str1 == str2`
- **Check String Inequality**: `str1 != str2`

10. String Information

- **Length of String**: `len(str)`
- **Minimum Character**: `min(str)`
- **Maximum Character**: `max(str)`

11. String Literals

- **Raw String**: `r'raw\string'`
- **Multiline String**: `'''Line 1\nLine 2'''`
- **Concatenation**: `'Hello ' + 'World'`
- **Repetition**: `'Repeat ' * 3`

12. Advanced String Formatting

- **String Interpolation (f-strings)**: `f'Hello, {name}!'`
- **String Template**: `from string import Template; t = Template('Hello, $name!'); t.substitute(name='World')`

13. Unicode Handling

- **Unicode String**: `'unicode string'`
- **Encode Unicode**: `'str'.encode('utf-8')`
- **Decode Byte to String**: `b'byte'.decode('utf-8')`

14. String Conversion

- **String to List**: `'str'.split()`
- **List to String**: `''.join(['s', 't', 'r'])`
- **String to Int**: `int('42')`
- **String to Float**: `float('4.2')`

15. Slice and Dice

- **Substring Extraction**: `'string'[start:end]`
- **Reverse String**: `'string'[::-1]`
- **Skip Characters while Slicing**: `'string'[start:end:step]`

16. String Iteration

- **Iterate over Characters**: `[char for char in 'string']`
- **Enumerate Characters**: `[(i, char) for i, char in enumerate('string')]`

17. String Comparison

- **Lexicographical Comparison**: `str1 < str2`
- **Case-Insensitive Comparison**: `str1.lower() == str2.lower()`

18. String Memory and Identity

- **String Identity (is)**: `str1 is str2`
- **String Identity (is not)**: `str1 is not str2`

19. Debugging Strings

- **Printable Representation**: `repr('str\n')`

20. String Methods with Keywords

- **startswith with Tuple of Prefixes**: `'string'.startswith(('s', 'st'))`
- **endswith with Tuple of Suffixes**: `'string'.endswith(('g', 'ng'))`

21. String Methods and ASCII

- **Get ASCII Value of Character**: `ord('a')`
- **Get Character from ASCII Value**: `chr(97)`

22. String Constants

- **String of ASCII Letters**: `string.ascii_letters`
- **String of ASCII Lowercase Letters**: `string.ascii_lowercase`
- **String of ASCII Uppercase Letters**: `string.ascii_uppercase`
- **String of Digits**: `string.digits`
- **String of Hexadecimal Digits**: `string.hexdigits`
- **String of Octal Digits**: `string.octdigits`
- **String of Punctuation**: `string.punctuation`
- **String of Printable Characters**: `string.printable`
- **String of Whitespace Characters**: `string.whitespace`

23. String Parsing and Extraction

- **Extract Substring by Index**: `'string'[1:4]`
- **Extract Last n Characters**: `'string'[-n:]`
- **Extract First n Characters**: `'string'[:n]`

24. Checking String Characteristics

- **Check if String is All Lowercase**: `'string'.islower()`
- **Check if String is All Uppercase**: `'string'.isupper()`
- **Check if String is Capitalized (First letter uppercase, rest lowercase)**: `'string'.istitle()`

25. String Mutability

- **Immutable Nature of Strings**: `s = 'string'; s = 'new' + s[5:]`
- **Creating a New String from the Old One**: `new_s = s[:5] + 'new' + s[8:]`

26. Converting Between Strings and Lists

- **Splitting a String into a List of Words**: `'The quick brown fox'.split()`
- **Joining a List of Words into a String**: `' '.join(['The', 'quick', 'brown', 'fox'])`

27. Cleaning Strings

- **Removing Leading and Trailing Spaces**: `' string '.strip()`
- **Removing Leading Spaces Only**: `' string '.lstrip()`
- **Removing Trailing Spaces Only**: `' string '.rstrip()`

28. Aligning Strings

- **Left Align String**: `'string'.ljust(10)`
- **Right Align String**: `'string'.rjust(10)`
- **Center Align String**: `'string'.center(10)`

29. String Repetition and Concatenation

- **Repeating Strings**: `'string' * 3`
- **Concatenating Strings**: `'string1' + 'string2'`

30. String Interpolation (Advanced Formatting)

- **Old Style (% operator)**: `'Hello %s' % ('World',)`
- **New Style (.format)**: `'Hello {}'.format('World')`

31. String Escape Sequences

- **New Line**: `print('line1\\nline2')`
- **Tab**: `print('column1\\tcolumn2')`
- **Backslash**: `print('Backslash: \\\\')`

32. String Literals (Advanced)

- **Byte String**: `b'byte string'`

- **Raw String (suppresses escape sequence processing)**: `r'raw\string\n'`

33. String Unpacking

- **Unpacking Characters into Variables**: `a, b, c = 'abc'`

34. Dealing with Quotes in Strings

- **Single Quotes Inside Double Quotes**: `"That's a quote"`
- **Double Quotes Inside Single Quotes**: `'He said, "Hello"'`

35. String Documentation (Docstrings)

- **Triple-Quoted Multiline Strings as Docstrings**: `"""This is a docstring"""`

36. String Encoding/Decoding

- **Encoding a String**: `'string'.encode('utf-8')`
- **Decoding Bytes to String**: `b'string'.decode('utf-8')`

37. String Memory Interning

- **Interning Strings**: `sys.intern('string')`

38. Making a String "Safe" for Filenames or URLs

- **Escape for URL**: `urllib.parse.quote('string')`
- **Safe Filenames**: `re.sub(r'[^\w\s-]', '', 'string').strip().lower()`

39. Multi-Line Strings

- **Define Multi-Line String**: `'''Line 1\nLine 2'''`

40. Text Wrapping and Filling

- **Text Wrapping**: `textwrap.wrap('long string', width=50)`
- **Text Filling**: `textwrap.fill('long string', width=50)`