Training Report Day-7

13 June 2024

➢ What is python String?

In Python, strings are used for representing textual data. A string is a sequence of characters enclosed in either single quotes ('') or double quotes (“”). The Python language provides various built-in methods and functionalities to work with strings efficiently.

➢ Python String Methods:-

1. Capitalize ()

Description: Converts the first character to uppercase and the rest to lowercase.

Example:

s = "hello world"

print(capitalize()) # Output: "Hello world"

2. Case fold ()

Description: Converts the string to lowercase.

Example:

s = "Hello World"

print(s.casefold()) # Output: "hello world"

3. `center(width[, fillchar])`

Description : Centers the string in a field of given width.

Example :

s = "hello"

print(s.center(10, ' ')) # Output: " hello "

4. `count(sub[, start[, end]])`

Description : Returns the number of non overlapping occurrences of substring sub.

Example :

s = "hello world"

print(s.count('l')) # Output: 3

5. `encode(encoding='utf 8', errors='strict')`

Description : Encodes the string using the specified encoding.

Example :

s = "hello"

print(s.encode()) # Output: b'hello'

6. `endswith(suffix[, start[, end]])`

Description : Returns True if the string ends with the specified suffix.

Example :

s = "hello world"

print(s.endswith("world")) # Output: True

7. `expandtabs(tabsize=8)`

Description : Replaces tabs with spaces.

Example :

s = "hello\tworld"

print(s.expandtabs()) # Output: "hello world"

8. `find(sub[, start[, end]])`

Description : Returns the lowest index where the substring is found.

Example :

s = "hello world"

print(s.find("world")) # Output: 6

9. `format(*args, kwargs)`

Description : Formats the string using the given arguments.

Example :

s = "Hello {}"

print(s.format("world")) # Output: "Hello world"

10. `format_map(mapping)`

Description : Formats the string using the dictionary provided.

Example :

s = "Hello {name}"

print(s.format_map({"name": "world"})) # Output: "Hello world"

11. `index(sub[, start[, end]])`

Description : Returns the lowest index where the substring is found.

Example :

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

12. `isalnum()`

Description : Returns True if all characters are alphanumeric.

Example :

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

13. `isalpha()`

Description : Returns True if all characters are alphabetic.

Example :

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

14. `isascii()`

Description : Returns True if all characters are ASCII.

Example :

```
s = "hello"
print(s.isascii()) # Output: True
```

15. `isdecimal()`

Description : Returns True if all characters are decimal.

Example :

```
s = "123"
print(s.isdecimal()) # Output: True
```

16. `isdigit()`

Description : Returns True if all characters are digits.

Example :

s = "123"

print(s.isdigit()) # Output: True

17. `isidentifier()`

Description : Returns True if the string is a valid identifier.

Example :

s = "hello_world"

print(s.isidentifier()) # Output: True

18. `islower()`

Description : Returns True if all characters are lowercase.

Example :

s = "hello"

print(s.islower()) # Output: True

19. `isnumeric()`

Description : Returns True if all characters are numeric.

Example :

s = "123"

print(s.isnumeric()) # Output: True

20. `isprintable()`

Description : Returns True if all characters are printable.

Example :

s = "hello"

print(s.isprintable()) # Output: True

21. `isspace()`

Description : Returns True if all characters are whitespace.

Example :

s = " "

print(s.isspace()) # Output: True

22. `istitle()`

Description : Returns True if the string is titlecased.

Example :

s = "Hello World"

print(s.istitle()) # Output: True

23. `isupper()`

Description : Returns True if all characters are uppercase.

Example :

s = "HELLO"

print(s.isupper()) # Output: True

24. `join(iterable)`

Description : Concatenates the strings in the iterable.

Example :

s = " "

print(s.join(["hello", "world"])) # Output: "hello world"

25. `ljust(width[, fillchar])`

Description : Left justifies the string in a field of given width.

Example :

s = "hello"

print(s.ljust(10, ' ')) # Output: "hello "

26. `lower()`

Description : Converts the string to lowercase.

Example :

s = "Hello World"

print(s.lower()) # Output: "hello world"

27. `lstrip([chars])`

Description : Removes leading characters.

Example :

s = " hello"

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

28. `maketrans(x[, y[, z]])`

Description : Returns a translation table for use with `str.translate()`.

Example :

table = str.maketrans("abc", "123")

s = "abc"

print(s.translate(table)) # Output: "123"

29. `partition(sep)`

Description : Splits the string at the first occurrence of sep.

Example :

s = "hello world"

print(s.partition(" ")) # Output: ('hello', ' ', 'world')

30. `replace(old, new[, count])`

Description : Replaces occurrences of a substring.

Example :

s = "hello world"

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

31. `rfind(sub[, start[, end]])`

Description : Returns the highest index where the substring is found.

Example :

s = "hello world"

print(s.rfind("o")) # Output: 7

32. `rindex(sub[, start[, end]])`

Description : Returns the highest index where the substring is found.

Example :

s = "hello world"

print(s.rindex("o")) # Output: 7

33. `rjust(width[, fillchar])`

Description : Right justifies the string in a field of given width.

Example :

s = "hello"

print(s.rjust(10, ' ')) # Output: " hello"

34. `rpartition(sep)`

Description : Splits the string at the last occurrence of sep.

Example :

s = "hello world"

print(s.rpartition(" ")) # Output: ('hello', ' ', 'world')

35. `rsplit(sep=None, maxsplit= 1)`

Description : Splits the string from the right.

Example :

s = "hello world"

print(s.rsplit()) # Output: ['hello', 'world']

36. `rstrip([chars])`

Description : Removes trailing characters.

Example :

s = "hello "

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

37. `split(sep=None, maxsplit= 1)`

Description : Splits the string at the separator.

Example :

s = "hello world"

print(s.split()) # Output: ['hello', 'world']

38. `splitlines([keepends])`

Description : Splits the string at line breaks.

Example :

s = "hello\nworld"

print(s.splitlines()) # Output: ['hello', 'world']

39. `startswith(prefix[, start[, end]])`

Description : Returns True if the string starts with the specified prefix.

Example :

s = "hello world"

print(s.startswith("hello")) # Output: True

40. `strip([chars])`

Description : Removes leading and trailing characters.

Example :

s = " hello "

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

41. `swapcase()`

Description : Swaps case, converting lowercase to uppercase and vice versa.

Example :

s = "Hello World"

print(s.swapcase()) # Output: "hELLO wORLD"

42. `title()`

Description : Converts the string to title case.

Example :

s = "hello world"

print(s.title()) # Output: "Hello World"

43. `translate(table)`

Description : Translates the string using the given translation table.

Example :

table = str.maketrans("abc", "123")

s = "abc"

print(s.translate(table)) # Output: "123"

44. `upper()`

Description : Converts the string to uppercase.

Example :

s = "hello world"

print(s.upper()) # Output: "HELLO WORLD"

45. `zfill(width)`

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

Example :

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