Programming and Problem Solving through Python Language
O Level / A Level

Chapter - 6 : Functions

String Functions

A string is a sequence of characters enclosed in quotation marks.

12. Replace ( ) Function

This function replaces a specified phrase with another specified phrase.

Syntax string.replace(oldValue, newValue, count)

oldValue(Required)	- A String	-Old string want to replace.
newValue(Required)	- A String	-New string want to place.
count(Optional)	- An Integer	- number of times to replace.

Example

```
txt = "I love apples, apple are my favorite apple"
x = txt.replace("apple", "orange", 1)
print(x)

txt = "I love apples, apple are my favorite apple"
x = txt.replace("apple", "orange")
print(x)
```

Output

```
I love oranges, apple are my favorite apple
I love oranges, orange are my favorite orange
```

13. Strip( ) Function

This function removes any leading and trailing characters. By default it removes spaces.

Syntax string.strip(character)

character(Optional)	- A String	- A characters to remove.

Example

```
txt = "   I love apples, apple are my favorite apple  "
x = txt.strip()
print(x)

txt = "apples, apple are my favorite apple,apple"
x = txt.strip( "apple,")
print(x)
```

Output

```
I love apples, apple are my favorite apple
s, apple are my favorite
```

14. Lstrip( ) Function

This function removes any leading characters. By default it removes spaces.

Syntax string.lstrip(character)

character(Optional) - A String - A characters to remove.

Example

```
txt = "   I love apples, apple are my favorite apple  "
x = txt.lstrip()
print(x)

txt = "apples, apple are my favorite apple,apple"
x = txt.lstrip( "apple,")
print(x)
```

Output

```
I love apples, apple are my favorite apple
s, apple are my favorite apple,apple
```

15. Rstrip( ) Function

This function removes any trailing characters. By default it removes spaces.

Syntax string.rstrip(character)

character(Optional) - A String - A characters to remove.

Example

```
txt = "   I love apples, apple are my favorite apple  "
x = txt.rstrip()
print(x)

txt = "apples, apple are my favorite apple,apple"
x = txt.rstrip( "apple,")
print(x)
```

Output

```
    I love apples, apple are my favorite apple
apples,, apple are my favorite
```

16. Split( ) Function

This function breaks the string at the specified separator, and returns a list. The default separator is space.

Syntax string.split(character, maxsplit)

character(Optional) - A String -A characters used as separator.
maxsplit(Optional) -An Integer -Count of splits to do. Default is -1(All).

If maxsplit is given, the list contains the specified number of elements plus one.

Example

```
txt = "I love apples, apple are my favorite apple"
x = txt.split()
print(x)

txt = "I love apples, apple are my favorite apple"
x = txt.split(" ",5)
print(x)

txt = "apples, apple are my favorite apple,apple"
x = txt.split( ",")
print(x)
```

Output

```
['I', 'love', 'apples,', 'apple', 'are', 'my', 'favorite', 'apple']
['I', 'love', 'apples,', 'apple', 'are', 'my favorite apple']
['apples', ' apple are my favorite apple', 'apple']
```

17. Partition( ) Function

This function searches for a string, and splits the string into a **tuple** having 3 elements.

1. The first part contains the part before the specified string.
2. The second part contains the specified string.
3. The third part contains the part after the string.

If the specified value is not found, then the tuple contains:

1 - the whole string,   2 - an empty string,   3 - an empty string

Syntax string.partition(character)

character(Required)   - A String   -A characters used to partition.

Example

```
txt = "I love apples, apple are my favorite apple"
x = txt.partition("apple")
print(x)

txt = "I love apples, apple are my favorite apple"
x = txt.partition("my")
print(x)

txt = "I love apples, apple are my favorite apple"
x = txt.partition("this")
print(x)
```

Output

```
('I love ', 'apple', 's, apple are my favorite apple')
('I love apples, apple are ', 'my', ' favorite apple')
('I love apples, apple are my favorite apple', ' ', ' ')
```

18. Join( ) Function

This function joins all items in a iterable (tuple, list, dictionary, set) into a string, using a hash character as separator

Syntax string.join(Object)

Object(Required) - A iterable - It may be tuple, list , set or dictionary **where all the values must be strings.**

If a dictionary used as an iterable, the returned values are the keys, not the values.

Example

```
obj = ("Apple", "Orange", "Mango")
x = "#".join(obj)
print(x)

obj = ["Apple", "Orange", "Mango"]
x = "#".join(obj)
print(x

obj = {"Fruit": "Apple", "Cost": "100","Weight": "500"}
mySeparator = "--"
x = mySeparator.join(obj)
print(x)
```

Output

```
Apple#Orange#Mango
Apple#Orange#Mango
Fruit--Cost--Weight
```

19. isspace( ) Function

This method returns True if all the characters are spaces, otherwise False.

Syntax string.isspace( )

Example

```
a = "   "
b = "  h  "
print(a.isspace())
print(b.isspace())
```

Output

```
True
False
```

20. isalpha( ) Function

This method returns True if all the characters are alphabet[A-Z,a-z], otherwise False.

Syntax string.isalpha( )

Example

```
obj = "    "
x=obj.isalpha()
print(x)

obj = "Abc"
```

```
x=obj.isalpha()
print(x)

obj = "Abc7"
x=obj.isalpha()
print(x)
```

Output

```
False
True
False
```

21. isdigit( ) Function

This method returns True if all the characters are digits[0-9], otherwise False.

Syntax string.isalpha( )

Example

```
obj = "12"
x=obj.isdigit()
print(x)

obj = "12.98"
x=obj.isdigit()
print(x)

obj = "78a"
x=obj.isdigit()
print(x)

obj = "\u0030"          #unicode for 0
x=obj.isdigit()
print(obj, "--",x)

obj = "\u00B2"          #unicode for exponent 2
x=obj.isdigit()
print(obj, "--",x)
```

Output

```
True
False
False
0—True
² -- True
```