What are strings?

In python, anything that you enclose between single or double quotation marks is considered a string. A string is essentially a sequence or array of textual data. Strings are used when working with Unicode characters.

Example

```
name = "Harry"
print("Hello, " + name)
```

Output

```
Hello, Harry
```

Note: It does not matter whether you enclose your strings in single or double quotes, the output remains the same.

Sometimes, the user might need to put quotation marks in between the strings. Example, consider the sentence: He said, “I want to eat an apple”.

How will you print this statement in python?: He said, "I want to eat an apple". We will definitely use single quotes for our convenience

```
print('He said, "I want to eat an apple".')
```

Multiline Strings

If our string has multiple lines, we can create them like this:

```
a = """Lorem ipsum dolor sit amet,
consectetur adipiscing elit,
sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua."""
print(a)
```

Accessing Characters of a String

In Python, string is like an array of characters. We can access parts of string by using its index which starts from 0.
Square brackets can be used to access elements of the string.

```
print(name[0])
print(name[1])
```

Looping through the string

We can loop through strings using a for loop like this:

```
for character in name:
   print(character)
```

Above code prints all the characters in the string name one by one!

String Slicing & Operations on String

Length of a String

We can find the length of a string using len() function.

Example:

```
fruit = "Mango"
len1 = len(fruit)
print("Mango is a", len1, "letter word.")
```

Output:

```
Mango is a 5 letter word.
```

String as an array

A string is essentially a sequence of characters also called an array. Thus we can access the elements of this array.

Example:

```
pie = "ApplePie"
print(pie[:5])
print(pie[6])           #returns character at specified index
```

Output:

```
Apple
i
```

Note: This method of specifying the start and end index to specify a part of a string is called slicing.

Slicing Example:

```
pie = "ApplePie"
print(pie[:5])      #Slicing from Start
print(pie[5:])      #Slicing till End
print(pie[2:6])     #Slicing in between
print(pie[-8:])     #Slicing using negative index
```

Output:

```
Apple
Pie
pleP
ApplePie
```

Loop through a String:

Strings are arrays and arrays are iterable. Thus we can loop through strings.

Example:

```
alphabets = "ABCDE"
for i in alphabets:
    print(i)
```

Output:

```
A
B
C
D
E
```

String methods

Python provides a set of built-in methods that we can use to alter and modify the strings.

upper() :

The upper() method converts a string to upper case.

Example:

```
str1 = "AbcDEfghIJ"
print(str1.upper())
```

Output:

```
ABCDEFGHIJ
```

lower()

The lower() method converts a string to lower case.

Example:

```
str1 = "AbcDEfghIJ"
print(str1.lower())
```

Output:

```
abcdefghij
```

strip() :

The strip() method removes any white spaces before and after the string.

Example:

```
str2 = " Silver Spoon "
print(str2.strip)
```

Output:

```
Silver Spoon
```

rstrip() :

the rstrip() removes any trailing characters. Example:

```
str3 = "Hello !!!"
```

```
print(str3.rstrip("!"))
```

Output:

```
Hello
```

replace() :

The replace() method replaces all occurences of a string with another string. Example:

```
str2 = "Silver Spoon"
print(str2.replace("Sp", "M"))
```

Output:

```
Silver Moon
```

split() :

The split() method splits the given string at the specified instance and returns the separated strings as list items.

Example:

```
str2 = "Silver Spoon"
print(str2.split(" "))      #Splits the string at the whitespace " ".
```

Output:

```
['Silver', 'Spoon']
```

There are various other string methods that we can use to modify our strings.

capitalize() :

The capitalize() method turns only the first character of the string to uppercase and the rest other characters of the string are turned to lowercase. The string has no effect if the first character is already uppercase.

Example:

```
str1 = "hello"
capStr1 = str1.capitalize()
print(capStr1)
str2 = "hello WorlD"
capStr2 = str2.capitalize()
print(capStr2)
```

Output:

```
Hello
Hello world
```

center() :

The center() method aligns the string to the center as per the parameters given by the user.

Example:

```
str1 = "Welcome to the Console!!!"
print(str1.center(50))
```

Output:

```
            Welcome to the Console!!!
```

We can also provide padding character. It will fill the rest of the fill characters provided by the user.

Example:

```
str1 = "Welcome to the Console!!!"
print(str1.center(50, "."))
```

Output:

```
............Welcome to the Console!!!.............
```

count() :

The count() method returns the number of times the given value has occurred within the given string.

Example:

```
str2 = "Abracadabra"
countStr = str2.count("a")
print(countStr)
```

Output:

```
4
```

endswith() :

The endswith() method checks if the string ends with a given value. If yes then return True, else return False.

Example :

```
str1 = "Welcome to the Console !!!"
print(str1.endswith("!!!"))
```

Output:

```
True
```

We can even also check for a value in-between the string by providing start and end index positions.

Example:

```
str1 = "Welcome to the Console !!!"
print(str1.endswith("to", 4, 10))
```

Output:

```
True
```

find() :

The find() method searches for the first occurrence of the given value and returns the index where it is present. If given value is absent from the string then return -1.

Example:

```
str1 = "He's name is Dan. He is an honest man."
print(str1.find("is"))
```

Output:

```
10
```

As we can see, this method is somewhat similar to the index() method. The major difference being that index() raises an exception if value is absent whereas find() does not.

Example:

```
str1 = "He's name is Dan. He is an honest man."
print(str1.find("Daniel"))
```

Output:

```
-1
```

index() :

The index() method searches for the first occurrence of the given value and returns the index where it is present. If given value is absent from the string then raise an exception.

Example:

```
str1 = "He's name is Dan. Dan is an honest man."
print(str1.index("Dan"))
```

Output:

```
13
```

As we can see, this method is somewhat similar to the find() method. The major difference being that index() raises an exception if value is absent whereas find() does not.

Example:

```
str1 = "He's name is Dan. Dan is an honest man."
```

```
print(str1.index("Daniel"))
```

Output:

```
ValueError: substring not found
```

isalnum() :

The isalnum() method returns True only if the entire string only consists of A-Z, a-z, 0-9. If any other characters or punctuations are present, then it returns False.

Example 1:

```
str1 = "WelcomeToTheConsole"
print(str1.isalnum())
```

Output:

```
True
```

isalpha() :

The isalnum() method returns True only if the entire string only consists of A-Z, a-z. If any other characters or punctuations or numbers(0-9) are present, then it returns False.

Example :

```
str1 = "Welcome"
print(str1.isalpha())
```

Output:

```
True
```

islower() :

The islower() method returns True if all the characters in the string are lower case, else it returns False.

Example:

```
str1 = "hello world"
print(str1.islower())
```

Output:

```
True
```

isprintable() :

The isprintable() method returns True if all the values within the given string are printable, if not, then return False.

Example :

```
str1 = "We wish you a Merry Christmas"
print(str1.isprintable())
```

Output:

```
True
```

isspace() :

The isspace() method returns True only and only if the string contains white spaces, else returns False.

Example:

```
str1 = "        "       #using Spacebar
print(str1.isspace())
str2 = "        "       #using Tab
print(str2.isspace())
```

Output:

```
True
True
```

istitle() :

The istitile() returns True only if the first letter of each word of the string is capitalized, else it returns False.

Example:

```
str1 = "World Health Organization"
print(str1.istitle())
```

Output:

```
True
```

Example:

```
str2 = "To kill a Mocking bird"
print(str2.istitle())
```

Output:

```
False
```

isupper() :

The isupper() method returns True if all the characters in the string are upper case, else it returns False.

Example :

```
str1 = "WORLD HEALTH ORGANIZATION"
```

```
print(str1.isupper())
```

Output:

```
True
```

startswith() :

The endswith() method checks if the string starts with a given value. If yes then return True, else return False.

Example :

```
str1 = "Python is a Interpreted Language"
print(str1.startswith("Python"))
```

Output:

```
True
```

swapcase() :

The swapcase() method changes the character casing of the string. Upper case are converted to lower case and lower case to upper case.

Example:

```
str1 = "Python is a Interpreted Language"
print(str1.swapcase())
```

Output:

```
pYTHON IS A iNTERPRETED lANGUAGE
```

title() :

The title() method capitalizes each letter of the word within the string.

Example:

```
str1 = "He's name is Dan. Dan is an honest man."
print(str1.title())
```

Output:

```
He'S Name Is Dan. Dan Is An Honest Man.
```