String

The string can be defined as the sequence of characters represented in the quotation marks. In python, we can use single, double, or triple quotes to define a string.

Subsets of strings can be taken using the slice operator ([ ] and [:]) with indexes starting at 0 in the beginning of the string and working their way from -1 at the end.

The plus (+) sign is the string concatenation operator and the asterisk (*) is the repetition operator.

Example	Output
str = 'Hello Python' print str print str[0] print str[1:4] print str[4:] print str * 3 print str + "PYTHON"	Hello Python H ell o Python Hello PythonHello PythonHello Python Hello PythonPYTHON

String Concatenation Operator (+)

The concatenation operator (+) concatenates two Strings and creates a new String.

Example	Output

str = 'Hello' str1 = 'Python' print(str+str1)	HelloPython

String Replication Operator (*)

The Replication operator is used to repeat a string number of times. The string will be repeated the number of times which is given by the integer value.

Example	Output
str = 'Hello' print str * 3	HelloHelloHello

String Slice Notation

Python String slice can be defined as a substring which is the part of the string. Therefore further substring can be obtained from a string.

There can be many forms to slice a string, as string can be accessed or indexed from both the direction and hence string can also be sliced from both the directions.

Example	Output
str = 'Hello Python' print str[0] print str[1:4]	H ell o Python

print str[4:]	

strip() method

The strip() method removes any whitespace from the beginning or the end.

Example	Output
str = " Hello, Python " print(str.strip())	Hello, Python

len() method

The len() method returns the length of a string.

Example	Output
str = "Hello" print(len(str))	5

lower() method

The lower() method returns the string in lower case.

Example	Output
str = "HELLO" print(str.lower())	hello

upper() method

The upper() method returns the string in upper case.

Example	Output
str = "hello" print(str.upper())	HELLO

replace() method

The replace() method replaces a string with another string.

Example	Output
str = "Hello" print(str.replace("H", "i"))	iello

split() method

The split() method splits the string into substrings if it finds instances of the separator

Example	Output
str = "Hello, Python" print(str.split(","))	['Hello', ' Python']

capitalize() Method

This method capitalizes the first character of the String.

Example	Output
str = "hello, Python" print(str.capitalize())	Hello Python

isalnum() Method

Returns true if string has at least 1 character and all characters are alphanumeric and false otherwise.

Example	Output
str = "HelloPython3"; print(str.isalnum()) str = "Hello Python3"; print(str.isalnum())	True False

isalpha()

Returns true if string has at least 1 character and all characters are alphabetic and false otherwise.

Example	Output
str = "Hello"; print(str.isalpha()) str = "Hello Python"; print(str.isalpha())	True False

isdigit()

Returns true if string contains only digits and false otherwise.

Example	Output
str = "Hello"; print(str.isdigit()) str = "123"; print(str.isdigit())	False True

count() Method

The method **count()** returns the number of occurrences of substring sub in the range [start, end]. Optional arguments start and end are interpreted as in slice notation.

Example	Output
str = "Hello"; sub = "l"; print(str.count(sub, 1, 4))	2