

# String Data Type

## What is String?

Any sequence of characters within either single quotes or double quotes or triple quotes is considered as a String.

### Syntax

```
str1="debdip"  
str1='debdip'  
str1=""debdip""
```

## How to define multi-line String

```
str1="" nsec is at  
      garia, garia  
      is at kolkata""
```

## use of backslash in case of string

```
s='This is ' single quote symbol' # invalid  
s='This is \' single quote symbol' #valid s="This is '  
single quote symbol" #valid s='This is " double  
quotes symbol' #valid  
s="" This is \' single quote symbol"" #valid
```

## How to access characters of a String:

We can access characters of a string by using the following ways.

1. By using index
2. By using slice operator

### 1. By using index:

Python supports both +ve and -ve index.  
+ve index means left to right(Forward direction)  
-ve index means right to left(Backward direction)

```
str1=""debdip bhanja Chowdhury""
```

```
print(str1[0])  
'd'
```

```
print(str1[-1])  
'y'
```

```
print(str1[100])  
IndexError: string index out of range
```

## 2.Accessing characters by using slice operator:

**Syntax:** str1 [begin index : end index : step]

**Begin index:** From where we have to consider slice(substring) **end index:** We have to terminate the slice(substring) at endindex-1

**step:** incremented value

```
str4="debdip bhanja Chowdhury"
```

```
print(str4[0:6])
```

```
debdip
```

```
str1="debdip"
```

```
print(str1[0:6:2])
```

```
dbi
```

```
print(str1[:]) # full string will be printed
```

```
debdip
```

```
print(str1[0:]) # starting from 0 index to end of the string
```

```
debdip
```

slicing by reverse index

```
str1="debdip"
```

```
print(str1[-4:])
```

```
bdip
```

d	e	b	d	e	f
0	1	2	3	4	5
-	-	-	-	-	-
6	5	4	3	2	1

Positive and negative index

## Mathematical Operators for String:

We can apply the following mathematical operators for Strings.

1. + operator for concatenation
2. \* operator for repetition

```
print("debdip"+" nsec") #debdipnsec
print("nsec"*3) #nsecnsecnsec
```

### Note:

1. To use + operator for Strings, compulsory both arguments should be str type
2. To use \* operator for Strings, compulsory one argument should be str and other argument should be int

## len() in-built function:

We can use len() function to find the number of characters present in the string.

Eg: s='debdip'  
print(len(s)) #6

## Checking Membership:

We can check whether the character or string is the member of another string or not by using in and not in operators

```
s='debdip'
print('d' in s) #True
print('r' in s) #False
```

## Finding Substrings:

### find():

```
s.find(substring)
```

Returns index of first occurrence of the given substring. If it is not available then we will get -1

```
s.find(substring,begin,end)
```

It will always search from begin index to end-1 index

## Counting substring in the given String:

We can find the number of occurrences of substring present in the given string by using count() method.

1. s.count(substring) # It will search through out the string
2. s.count(substring, begin, end) # It will search from begin index to end-1 index

```
s="abcabcabcadda"  
print(s.count('a'))  
print(s.count('ab'))  
print(s.count('a',3,7))
```

### Output:

```
6  
4  
2
```

## Replacing a string with another string:

```
s.replace(oldstring,newstring)
```

inside s, every occurrence of oldstring will be replaced with newstring.

### Eg1:

```
s="debdip bhanja chowdhury"  
s1=s.replace("debdip","subhadip")  
print(s1)
```

### Output:

```
Subhadip bhanja chowdhury
```

### Eg2: All occurrences will be replaced

```
s="ababababababab"  
s1=s.replace("a","b")  
print(s1)
```

Output: bbbbbbbbbbbbbbb

## Q. String objects are immutable then how we can change the content by using replace() method.

Once we create a string object, we cannot change the content. This non-changeable behaviour is nothing but immutability. If we are trying to change the content by using any method, then with those changes a new object will be created and changes won't be happen in existing object.

Hence with replace() method also a new object got created but existing object won't be changed.

**Eg:** `s="abab"`  
`s1=s.replace("a","b")`

## Splitting of Strings:

We can split the given string according to specified separator by using split() method.

`l=s.split(separator)`

The default separator is space. The return type of split() method is List

```
s="07-04-2021"
print(s.split('-'))
```

```
['07', '04', '2021']
```

## Joining of Strings:

We can join a group of strings(list or tuple) wrt the given separator.

```
s=separator.join(group of strings)
```

**Eg:** `t=('07','04','2021')`  
`s='-'.join(t)`  
`print(s)`

**Output:** 07-04-2021

```
str1="amal kumar dutta"
x=(str1.split(' '))
print(x)
```

```
['amal', 'kumar', 'dutta']
```

## Changing case of a String:

We can change case of a string by using the following 4 methods.

1. `upper()` To convert all characters to upper case
2. `lower()` To convert all characters to lower case
3. `swapcase()` converts all lower case characters to upper case and all upper case characters to lower case
4. `title()` To convert all character to title case. i.e first character in every word should be upper case and all remaining characters should be in lower case.
5. `capitalize()` Only first character will be converted to upper case and all remaining characters can be converted to lower case

```
s='Debdip Bhanja Chowdhury'
print(s.upper())
print(s.lower())
print(s.swapcase())
print(s.title())
print(s.capitalize())
```

### Output:

```
DEBDIP BHANJA CHOWDHURY
```

## Checking starting and ending part of the string:

Python contains the following methods for this purpose

1. `s.startswith(substring)`
2. `s.endswith(substring)`

### Eg:

```
s="debdip bhanja chowdhury"
print(s.startswith('debdip'))
print(s.endswith('debdip'))
print(s.endswith('chowdhury'))
```

output:-

True

False

True

## To check type of characters present in a string:

Python contains the following methods for this purpose.

- 1) **isalnum()**: Returns True if all characters are alphanumeric( a to z , A to Z ,0 to9 )
- 2) **isalpha()**: Returns True if all characters are only alphabet symbols(a to z,A to Z)
- 3) **isdigit()**: Returns True if all characters are digits only( 0 to 9)
- 4) **islower()**: Returns True if all characters are lower case alphabet symbols
- 5) **isupper()**: Returns True if all characters are upper case alphabet symbols
- 6) **istitle()**: Returns True if string is in title case
- 7) **isspace()**: Returns True if string contains only spaces

## Formatting the Strings:

We can format the strings with variable values by using replacement operator {} and format() method.

```
Eg: name='debdip'  
dept='mca'  
age=38  
print("{} is in {} and his age is {}".format(name,dept,age))  
print("{0} is in {1} and his age is {2}".format(name,dept,age))  
print("{x} is in {y} and his age is {z}".format(z=age,y=dept,x=name))
```

### **Output:**

```
debdip is in mca and his age is 38  
debdip is in mca and his age is 38  
debdip is in mca and his age is 38
```

```
In [1]: a='debdip'  
print(a)
```

debdip

```
In [2]: a="debdip"  
print(a)
```

debdip

```
In [3]: a='''debdip'''  
print(a)
```

debdip

```
In [4]: str1="what's your name?"  
print(str1)
```

what's your name?

```
In [5]: str1='what's your name?'  
print(str1)
```

```
File "<ipython-input-5-20645885f3e2>", line 1  
str1='what's your name?'
```

```
^  
SyntaxError: invalid syntax
```

```
In [6]: str1='what\'s your name?'      # use of escape character \  
print(str1)
```

what's your name?

```
In [1]: str1="debdip"  
str2="bhanja"  
str3="chowdhury"  
str4=str1+" "+str2+" "+str3  
print(str4)
```

debdip bhanja chowdhury

```
In [2]: print(str4[0])
```

d

```
In [3]: print(str4[0:6])
```

debdip

```
In [6]: str1="debdip"  
print(str1[0:6:2])
```

dbi

```
In [7]: print(len(str1))
```

6

```
In [9]: 'd' in str1
```

```
Out[9]: True
```

```
In [10]: str1="nsec is at garia"  
print(str1.find("garia"))
```

```
11
```

```
In [11]: print(str1.find("kolkata"))
```

```
-1
```

```
In [12]: print(str1.find('a'))
```

```
8
```

```
In [13]: print(str1.find('a',9,15))
```

```
12
```

```
In [14]: print(str1.count('a'))
```

```
3
```

```
In [15]: str2="nsec is at garia , nsec is an engineering college"  
print(str2.count('nsec',4,30))
```

```
1
```

```
In [16]: print(str2.count('nsec'))
```

```
2
```

```
In [17]: s="07-04-2021"  
print(s.split('-'))
```

```
['07', '04', '2021']
```

```
In [18]: s=('07','04','2021')  
t='-'.join(s)  
print(t)
```

```
07-04-2021
```

```
In [33]: str1="amal kumar dutta"  
x=(str1.split(' '))  
print(x)
```

```
['amal', 'kumar', 'dutta']
```

```
In [34]: print(str1.upper())
```

```
AMAL KUMAR DUTTA
```

```
In [35]:
```

```
print(str1.lower())
```

amal kumar dutta

```
In [36]: print(str1.swapcase())
```

AMAL KUMAR DUTTA

```
In [37]: print(str1.title())
```

Amal Kumar Dutta

```
In [38]: print(str1.capitalize())
```

Amal kumar dutta

```
In [40]: str1="nsec is at garia"  
str1.startswith('nsec')
```

Out[40]: True

```
In [41]: str1.endswith('garia')
```

Out[41]: True

```
In [42]: str1.isalnum()
```

Out[42]: False

```
In [44]: str1.isalpha()
```

Out[44]: False

```
In [45]: str1.isdigit()
```

Out[45]: False

```
In [46]: str1.islower()
```

Out[46]: True

```
In [47]: str1.isupper()
```

Out[47]: False

```
In [48]: str1.istitle()
```

Out[48]: False

```
In [49]: str1.isspace()
```

Out[49]: False

In [ ]:

```
In [1]: str1="HELLO"  
str1.casefold()
```

```
Out[1]: 'hello'
```

```
In [2]: str1.swapcase()
```

```
Out[2]: 'hello'
```

```
In [3]: str2=str1.casefold()  
print(str2)
```

```
hello
```

```
In [4]: print(str1)
```

```
HELLO
```

```
In [ ]:
```