



rishabh-mishra ▾

Reading: String Operations

[Bookmark this page](#)

In Python, a String is a sequence of characters.

A character is a symbol. It can be the 26 characters of English alphabets, digits, spaces or special characters.

How to create a String in Python?

Strings in Python are created by enclosing characters in double quotes or single quotes. Triple quotes are also used, generally, to represent multi-line strings.

As an example:

```
[1]: #defining strings in Python using single quotes
my_string = 'Hello'
print(my_string)

Hello

[2]: #defining strings in Python using double quotes
my_string = "Hello"
print(my_string)

Hello

[3]: #defining strings in Python using triple quotes
my_string = '''Hello'''
print(my_string)

Hello

[4]: #defining multiline strings in Python using triple quotes
my_string = """Hello, welcome to
    the world of Python"""
print(my_string)

Hello, welcome to
    the world of Python
```

How to access characters in a String?

We can access individual characters of a string using Indexing. The index starts at 0. It must always be an integer value. Using any other type will raise a **TypeError**.

```
[6]: my_string = "Hello"  
      print(my_string[0])
```

H

```
[7]: my_string = "Hello"  
      print(my_string[0.0])
```

TypeError

Traceback (most recent call last)

```
<ipython-input-7-c78a7808f7be> in <module>  
      1 my_string = "Hello"  
----> 2 print(my_string[0.0])
```

TypeError: string indices must be integers

If we access a character out of the index range, this will raise an ***IndexError***.

```
my_string = "Hello"  
print(my_string[5])
```

IndexError

Traceback (most recent call last)

```
<ipython-input-9-def207b20ed9> in <module>  
      1 my_string = "Hello"  
----> 2 print(my_string[5])
```

IndexError: string index out of range

Clearly, the length of the string "Hello" here is 5 i.e. (0-4), so my_string[5] will throw an error.

Python also allows negative indexing for its sequences. The index of -1 refers to the last item, -2 to the second last and so on.

```
my_string = "Hello"  
print(my_string[-1])
```

o

```
my_string = "Hello"  
print(my_string[-2])
```

l

A range of characters can be accessed using Slicing (slicing operator ':' (colon)).

```
my_string = "Hello"  
#slicing 1st to 4th character  
print('str[1:4] = ', my_string[1:4])  
  
#slicing 2nd to 2nd Last character  
print('str[2:-2] = ', my_string[1:-2])  
  
str[1:4] = ell  
str[2:-2] = el
```

H	E	L	L	O
0	1	2	3	4
-5	-4	-3	-2	-1

String Slicing in Python

String Slicing also accepts a third parameter, called the **stride**, which refers to as to how many characters you want to move forward after the first character is retrieved from the string. The value of stride is set to 1 by default. Look at the following example:

```
my_string = "1234567890"
#striding by setting value=2
print(my_string[::2])
#striding by setting value=4
print(my_string[::4])
```

```
13579
159
```

How to change/ delete a string in Python?

Strings are immutable. This means that you cannot change the value of the string once it is assigned. You can either create a new string or assign a different string to the existing string name.

```
my_string = "Hello"  
print(my_string)  
  
#assign new name to the string  
my_string = "Welcome to the course"  
print(my_string)
```

```
Hello  
Welcome to the course
```

We cannot delete the characters of the string. Instead, we may delete the entire string using the **del** keyword.

```
my_string = "Hello"  
print(my_string)  
del my_string  
print(my_string)
```

```
Hello
```

```
NameError Traceback (most recent call last)  
<ipython-input-22-e954c81fbe6a> in <module>  
      2 print(my_string)  
      3 del my_string  
----> 4 print(my_string)  
  
NameError: name 'my_string' is not defined
```

Python String Operations

There are several operations that can be performed on strings. This makes it one of the most used data types in Python.

A. Concatenation of Strings: Joining two or more strings into a single string is called concatenation.

The `+` operator does this in Python.

The `*` operator can be used to repeat the string for a given number of times.

Writing two string literals together concatenates them like `+` operator.

```
# Concatenation on strings
str1 = 'Hello'
str2 = 'World!'

# using +
print('str1 + str2 = ', str1 + str2)

# using *
print('str1 * 3 =', str1 * 3)

#two string literals together
'Hello ''World!'

str1 + str2 = HelloWorld!
str1 * 3 = HelloHelloHello
'Hello World!'
```

B. String Membership: We can check if a substring exists within a string or not using the ***in*** keyword.

```
'a' in 'hello'
```

```
False
```

```
'a' in 'apple'
```

```
True
```

C. Built-in functions: Some of the commonly used built-in functions are ***len()*** and ***enumerate()***.

The ***len()*** function returns the total number of characters in the string.

The ***enumerate()*** function returns an enumerate object. It contains the index-value pair of all the items in the string.

As an example:

```
sample_str = 'Hello'

#character count
print('len(sample_str) = ', len(sample_str))

# enumerate() #is represented as a list
list_enumerate = list(enumerate(sample_str))
print('list(enumerate(sample_str)) = ', list_enumerate)

len(sample_str) = 5
list(enumerate(sample_str)) = [(0, 'H'), (1, 'e'), (2, 'l'), (3, 'l'), (4, 'o')]
```

D. Escape Sequence: An escape sequence in Python starts with a backslash (\) and is interpreted differently from the usual strings.

Using single quotes or double quotes will result in a **SyntaxError**.

As an example:

```
print("She commanded, "Let's go"")

File "<ipython-input-33-8e2af9ac4ec1>", line 1
    print("She commanded, "Let's go""")
                           ^
SyntaxError: invalid syntax
```

We will now see how the same is implemented using an escape sequence.

```
print("She commanded, \"Let's go\"")
```

She commanded, "Let's go"

E. Common string methods: Some of the commonly used string methods are:

- a. lower():** This method converts the upper case characters to lower case characters.

```
sample_string = 'WelcomeToPython'

#lower()
print('lowercase:',sample_string.lower())

lowercase: welcometopython
```

- b. **upper()**: This method converts the lower case characters to upper case characters.

```
sample_string = 'WelcomeToPython'

#upper()
print('uppercase:',sample_string.upper())

uppercase: WELCOMETOPYTHON
```

- c. **join()**: This method returns a string by joining all the elements of an iterable, separated by a string operator.

```
sample_string = 'WelcomeToPython'

#join()
str1 = '12345'
str2 = 'abcde'
str3 = str1.join(str2)
print('join:',str3)

join: a12345b12345c12345d12345e
```

d. **split()**: This method breaks up a string at the specified separator and returns a list of strings.

```
sample_string = 'Welcome To Python'
sample_string2 = 'Welcome, To, Python'
#upper()
print('split:',sample_string.split()) #split at space

print('split:',sample_string2.split(',')) #split at ,
split: ['Welcome', 'To', 'Python']
split: ['Welcome', ' To', ' Python']
```

e. **find()**: This method finds a substring in the given string.

```
text = 'Welcome to Python Course'
print(text.find('t')) # first 't' find at 8th position

print(text.find('Course')) #the word 'Course' found at 18th position
```

8
18

f. **replace()**: This method replaces a segment of the string (called a substring) with a new string.

```
text = 'Welcome to Python Course'  
#replace 'Python' with 'Artificial Intelligence'  
print(text.replace('Python','Artificial Intelligence'))
```

Welcome to Artificial Intelligence Course



In today's modern age of disruption, SkillUp Online is your ideal learning platform that enables you to upskill to the most in-demand technology skills like Data Science, Big Data, Artificial Intelligence, Cloud, Front-End Development, DevOps & many more. In your journey of evolution as a technologist, SkillUp Online helps you work smarter, get to your career goals faster and create an exciting technology led future.

Corporate

- ▶ Home
- ▶ About Us
- ▶ Enterprise
- ▶ Blog
- ▶ Press

Support

- ▶ Contact us
- ▶ Terms of Service
- ▶ Privacy Policy

Copyright ©2020 [Skillup](#). All Rights Reserved