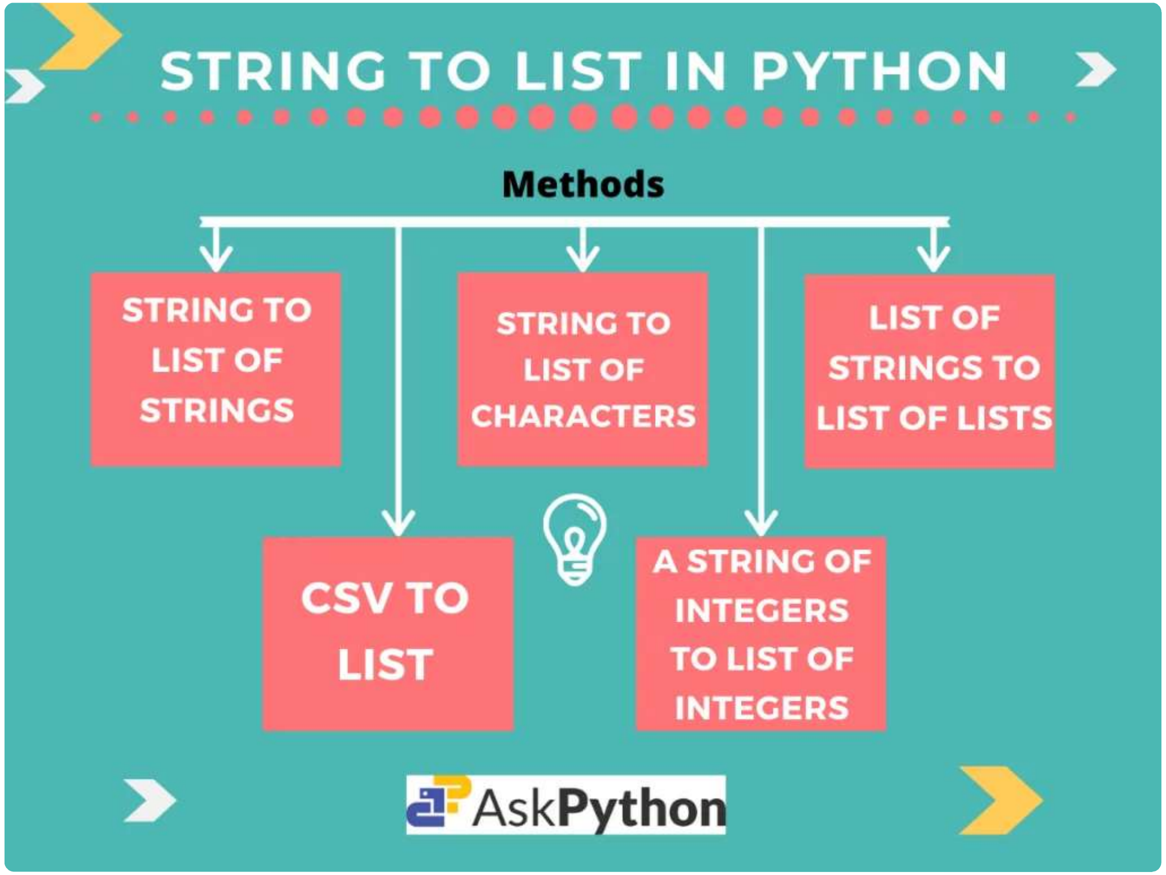
AskPython

Convert String to List in Python

Search ...

Recent Posts

An Ultimate Guide On Insider Threat Risks And Their Prevention



STRING TO LIST IN PYTHON

Methods

STRING TO LIST OF STRINGS

STRING TO LIST OF CHARACTERS

LIST OF STRINGS TO LIST OF LISTS

CSV TO LIST

A STRING OF INTEGERS TO LIST OF INTEGERS

AskPython

Introduction

While programming we may need to convert a **string** to **list** in Python. That could be for any other reason. But, a question arises here, how can we convert a string to different forms of lists?

A beginner's tutorial to train a classifier model with unlabeled data using semi-supervised learning (SSL)

When To Use Colon (:) in Python?

Animating Data in Python – A Simple Guide

GamStop API: Worth the Wait for Open Use

Why do Traders Need to Start Learning Python?

Resize the Plots and Subplots in Matplotlib Using figsize

Diagnose Fever In Python [Easy CLI Method]

Designing State Machines using Python [A Quick Guide]

Tkinter Create Oval – A Quick Guide

Favorite Sites

So, here in this tutorial, we are going to learn how we can convert a string into a list in **Python**.

Methods of converting a string to list in Python

Conversion of a string to list in Python is a pretty easy job. It can be achieved by following different methods as per our own requirements.

Here in this tutorial, we are going to deal with all the methods using which we can convert a string to list in Python for different cases. Below we have listed all the methods:

- **String to List of Strings**
- **String to List of Characters**
- **List of Strings to List of Lists**
- **CSV to List**
- **A string consisting of Integers to List of integers**

Now we are going to discuss each one of the above-mentioned techniques one-by-one.

1. String to List of Strings

When we need to convert a string to list in Python containing the constituent strings of the parent string(previously separated by some separator like **‘,’** or space), we use this method to accomplish the task.

For example, say we have a string **“Python is great”**, and we want a list which would contain only the given names previously separated by spaces, we can get the required list just by splitting the string into parts on the basis of the position of space.

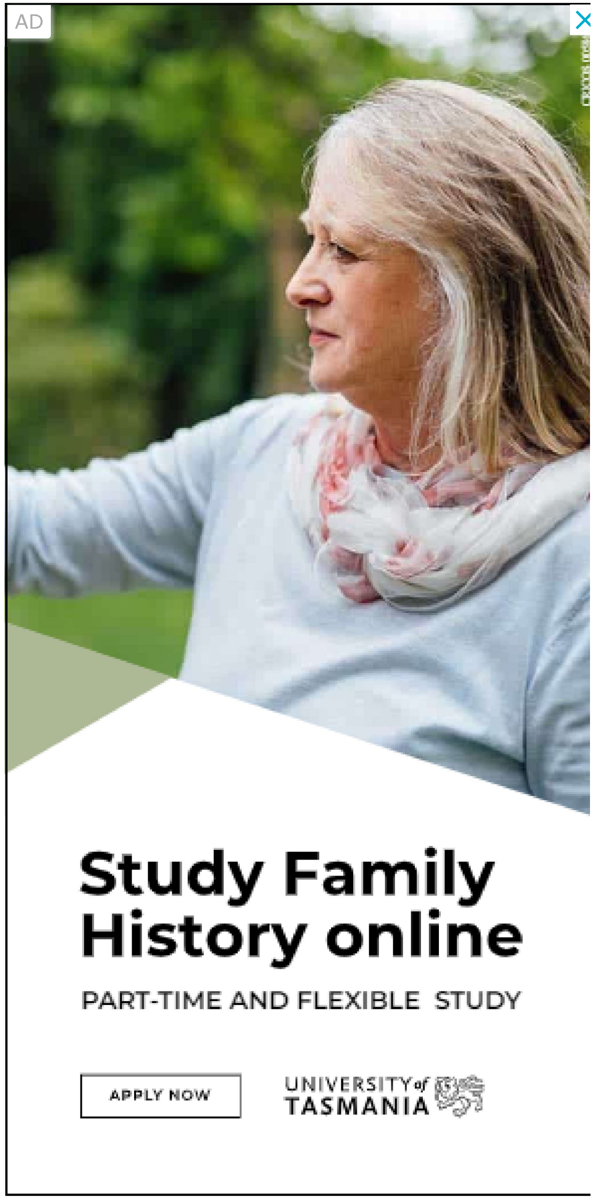
Let us look at an example to understand it better.

```
#given string
string1="Python is great"

#printing the string
print("Actual String: ",string1)

#gives us the type of string1
print("Type of string: ",type(string1))

print("String coverted to list :",string1.split())
#prints the list given by split()
```



Output:

String To List Of Strings

In the above code:

- We consider a string, `string1="Python is great"` and try to convert the same a list of the constituent strings
- `type()` gives us the type of object passed to the method, which in our case was a string
- `split()` is basically used to split a string into a list on the basis of the given separator. In our code, the words were separated by

spaces. By default, if we do not pass anything to the **split()** method it splits up the string on the basis of the position of spaces

Hence though we have not mentioned the separator parameter, the `split()` method gives us a list of the respective strings

2. String to List of Characters

What if we need a **list of characters** present in a string? In that case, direct type conversion from string to list in Python using the `list()` method does the job for us.

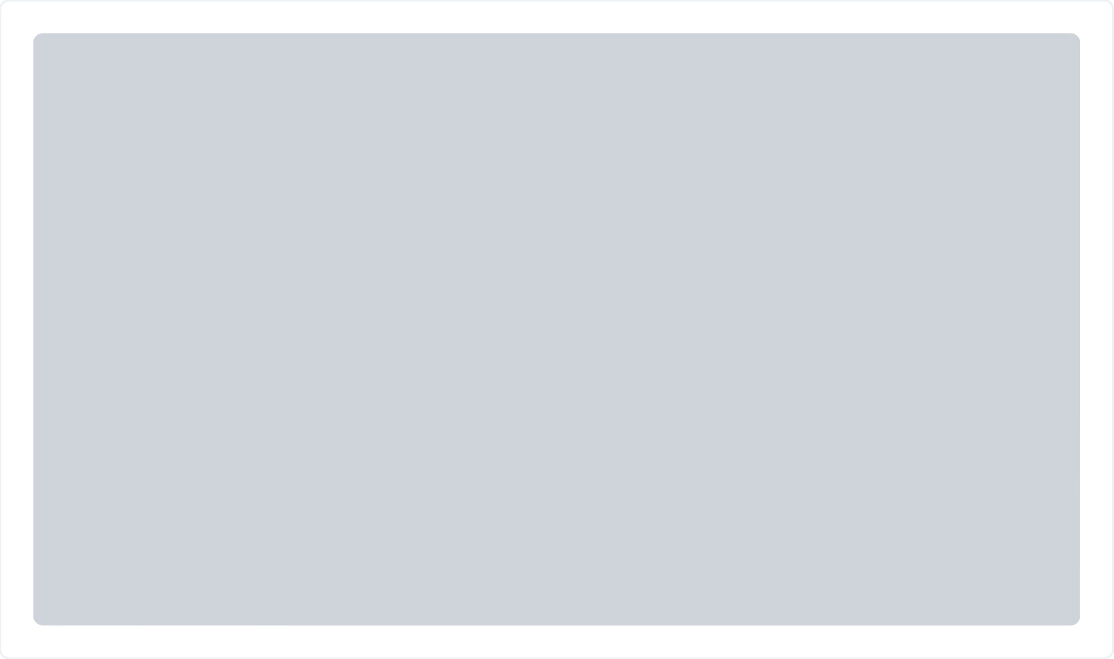
Certainly, if the input string is something like **"abcd"**, typecasting the string into a list using the `list()` method gives us a list having the individual characters **'a', 'b', 'c', 'd'** as its elements. Take a look at the given example code below.

```
#given string
string1="AskPython"

#printing the string
print("Actual String: ",string1)
#confirming the type()
print("Type of string: ",type(string1))
```

```
#type-casting the string into list using list()
print("String coverted to list :\n",list(string1))
```

Output:



String To List Of Characters

Understanding the code:

> Firstly here, we initialize a string, `string1` as **"AskPython"** and print its type using the `type()` method

And as we can observe, typecasting the string using the `list()` method gives us a list of the member characters, as required

3. List of Strings to List of Lists

Here, we are going to see how we can combine both the above methods to **convert a string to a list of character lists**.

Look at the below-given example carefully,

```
#Given string
string1="This is Python"

print("The actual string:",string1)

#converting string1 into a list of strings
string1=string1.split()

#applying list method to the individual elements of the lis
list1=list(map(list,string1))

#printing the resultant list of lists
print("Converted to list of character list :\n",list1)
```

Output:

String To List Of Character Lists

Understand the code:

> In this case, after the initialization of the string `string1`, we use the first method and convert it into a list of strings
>
> That is, at this point string1 is a list of strings given by `[ 'This', 'is', 'Python' ]`
>
> Then we apply the `list()` method to all the elements of the list
>
> **string1**. As we saw in our previous case this gives us a list consisting of character lists. Note, mass type-casting was performed using the map() function

4. CSV to List

A **CSV(Comma Separated Values)** string, as its name suggests is a string consisting of values or data separated by commas.

Let us look at how we can convert such type of string to list in Python.

```
#given string
string1="abc,def,ghi"
print("Actual CSV String: ",string1)
print("Type of string: ",type(string1))

#spliting string1 into list with ',' as the parameter
print("CSV coverted to list :",string1.split(','))
```

Output:

CSV To List

Here:

- Similarly, we initiate by considering a string **string1** with various data or values separated by commas(**‘,’**)
- After printing it and its `type()`, we proceed by splitting it on the basis of the parameter **‘,’**
- This makes the values **‘abc’**, **‘def’**, and **‘ghi’** the elements of a list. In this way, we were actually able to extract values from a given **CSV**

5. A string consisting of Integers to List of integers

Now we are going to convert a string consisting of only integers separated by some space, comma or etc., to a list with **integer** type elements.

For example, look at the code below,

```
#string with integers sepated by spaces
string1="1 2 3 4 5 6 7 8"
print("Actual String containing integers: ",string1)
print("Type of string: ",type(string1))

#coverting the string into list of strings
list1=list(string1.split())
print("Converted string to list : ",list1)

#typecasting the individual elements of the string list int
list2=list(map(int,list1))
print("List of integers : ",list2)
```

Output:

String With Integers To List

Now:

We took a string, `string1` as **“1 2 3 4 5 6 7 8”** and print it and its **type()** consecutively

Then we split it using the `split()` method and store the resultant list into a list, **list1**. At this point, list1 holds **[‘1’, ‘2’ , ‘3’, ‘4’, ‘5’, ‘6’, ‘7’, ‘8’]** as we can see from the output, as expected

Now we map the function `int()` throughout the list, typecasting each one of the elements into integers. And further, we store the typecasted mapped list into **list2** and print the same

As a result, we get a list consisting of the integer elements on which now we can perform arithmetic operations.

Conclusion

That’s all now, this was all about converting strings into different lists using various methods. Try to use the one which suits your code and solves your purpose as well as meets up to your requirements. Questions in the comments are appreciated.

References

https://www.askpython.com/python/string

← Previous Post Next Post →

AskPython is part of JournalDev IT Services Private Limited