

Python Tutorials and Notes

Python String find() Method with Examples



What is Python String find?

The Python string `find()` method helps to find the index of the first occurrence of the substring in the given string. It will return `-1` if the substring is not present.

In this tutorial, you will learn:

- What is Python String `find()`?
- Syntax
- Using `find()` method with default values
- Using `start` argument in `find()`
- Using `start` and `end` arguments in `find()`
- To find the position of a given substring in a string using `find()`
- Python string `rfind()`
- Python string `index()`

Syntax

```
string.find(substring,start,end)
```

Parameters

Here, are three parameters of Python string `find()`:

- **substring:** The substring you want to search in the given string.
- **start:** [optional] The start value from where the search for substring will begin. By default, it is `0`.
- **end:** [optional] The end value where the search for substring will end. By default, the value is the length of the string.

Using find() method with default values

The parameters passed to find() method are substring i.e the string you want to search for, start, and end. The start value is 0 by default, and the end value is the length of the string.

In this example, we will use the find() method with default values.

The find() method will search for the substring and give the position of the very first occurrence of the substring. Now, if the substring is present multiple times in the given string, still it will return you the index or position of the first one.

Example:

```
mystring = "Meet Guru99 Tutorials Site.Best site for Python Tutorials!"  
print("The position of Tutorials is at:", mystring.find("Tutorials"))
```

Output:

```
The position of Tutorials is at: 12
```

Using start argument in find()

You can search the substring in the given string and specify the start position, from where the search will begin. The start parameter can be used for the same.

The example will specify the start position as 15, and the find() method will begin the search from position 15. Here, the end position will be the length of the string and will search till the end of the string from 15 positions onwards.

Example:

```
mystring = "Meet Guru99 Tutorials Site.Best site for Python Tutorials!"  
print("The position of Tutorials is at:", mystring.find("Tutorials", 20))
```

Output:

```
The position of Tutorials is at 48
```

Using start and end arguments in find()

Using the start and end parameter, we will try to limit the search, instead of searching the entire string.

Example:

```
mystring = "Meet Guru99 Tutorials Site.Best site for Python Tutorials!"  
print("The position of Tutorials is at:", mystring.find("Tutorials", 5, 30))
```

Output:

```
The position of Tutorials is at 12
```

To find the position of a given substring in a string using find()

We know that find() helps us to find the index of the first occurrence of substring. It returns -1 if the substring is not present in the given string. The example below shows the index when the string is present and -1 when we don't find the substring we are searching for.

Example:

```
mystring = "Meet Guru99 Tutorials Site.Best site for Python Tutorials!"  
print("The position of Best site is at:", mystring.find("Best site", 5, 40))  
print("The position of Guru99 is at:", mystring.find("Guru99", 20))
```

Output:

```
The position of Best site is at: 27  
The position of Guru99 is at: -1
```

Python string rfind()

The Python function `rfind()` is similar to `find()` function with the only difference is that `rfind()` gives the highest index for the substring given and `find()` gives the lowest i.e the very first index. Both `rfind()` and `find()` will return -1 if the substring is not present.

In the example below, we have a string "Meet Guru99 Tutorials Site. Best site for Python Tutorials!" and will try to find the position of substring Tutorials using `find()` and `rfind()`. The occurrence of Tutorials in the string is twice.

Here is an example where both `find()` and `rfind()` are used.

```
mystring = "Meet Guru99 Tutorials Site.Best site for Python Tutorials!"  
print("The position of Tutorials using find() : ", mystring.find("Tutorials"))  
print("The position of Tutorials using rfind() : ", mystring.rfind("Tutorials"))
```

Output:

```
The position of Tutorials using find() : 12  
The position of Tutorials using rfind() : 48
```

The output shows that `find()` gives the index of the very first `Tutorials` substring that it gets, and `rfind()` gives the last index of substring `Tutorials`.

Python string index()

The Python string `index()` is function that will give you the position of the substring given just like `find()`. The only difference between the two is, `index()` will throw an exception if the substring is not present in the string and `find()` will return `-1`.

Here is a working example that shows the behaviour of `index()` and `find()`.

```
mystring = "Meet Guru99 Tutorials Site.Best site for Python Tutorials!"
print("The position of Tutorials using find() : ", mystring.find("Tutorials"))
print("The position of Tutorials using index() : ", mystring.index("Tutorials"))
```

Output:

```
The position of Tutorials using find() : 12
The position of Tutorials using index() : 12
```

We are getting same position for both `find()` and `index()`. Let us see an example when the substring given is not present in the string.

```
mystring = "Meet Guru99 Tutorials Site.Best site for Python Tutorials!"
print("The position of Tutorials using find() : ", mystring.find("test"))
print("The position of Tutorials using index() : ", mystring.index("test"))
```

Output:

```
The position of Tutorials using find() : -1
Traceback (most recent call last):
  File "task1.py", line 3, in <module>
    print("The position of Tutorials using index() : ", mystring.index("test"))
ValueError: substring not found
```

In the above example, we are trying to find the position of substring "test". The substring is not present in the given string, and hence using `find()`, we get the position as `-1`, but for `index()`, it throws an error as shown above.

To find the total occurrence of a substring

To find the total number of times the substring has occurred in the given string we will make use of `find()` function. Will loop through the string using for-loop from 0 till the end of the string. Will make use of `startIndex` parameter for `find()`.

Variables `startIndex` and `count` will be initialized to 0. Inside for-loop will check if the substring is present inside the string given using `find()` and `startIndex` as 0.

The value returned from `find()` if not `-1`, will update the `startIndex` to the index where the string is found and also increment the `count` value.

Here is the working example:

Output:

```
The total count of substring test is: 6
```

Summary

- The Python string `find()` method helps to find the index of the first occurrence of the substring in the given string. It will return -1 if the substring is not present.
- The parameters passed to `find()` method are substring i.e the string you want to search for, start, and end. The start value is 0 by default, and the end value is the length of the string.
- You can search the substring in the given string and specify the start position, from where the search will begin. The start parameter can be used for the same.
- Using the start and end parameter, we will try to limit the search, instead of searching the entire string.
- The Python function `rfind()` is similar to `find()` function with the only difference is that `rfind()` gives the highest index for the substring given and `find()` gives the lowest i.e the very first index. Both `rfind()` and `find()` will return -1 if the substring is not present.
- The Python string `index()` is yet another function that will give you the position of the substring given just like `find()`. The only difference between the two is, `index()` will throw an exception if the substring is not present in the string and `find()` will return -1.
- We can make use of `find()` to find the count of the total occurrence of a substring in a given string.