

# The slice () function in Python

The slice () function in Python returns a slice object that helps you determine how to cut an existing string.

Continuing with the topic of built-in functions in Python, the article will introduce you to the slice () function with syntax, usage, and specific examples. Invites you to read the track.

**The slice () function** in Python returns a slice object. This object helps you determine how to cut an existing string (be it string, bytes, tuple, list, range) or objects that support string protocols (such as `__getitem__ ()`, `__len__ ()`).

## The syntax of the slice () function in Python:

```
slice(stop) slice(start, stop, step)
```

## The parameters of the slice () function

The slice () function can have up to 3 parameters:

1. `start` : An integer specifying the position to start cutting the object. The default is 0.
2. `stop` : An integer specifying the position where the object ends.
3. `step` : An integer that specifies the increment between cuts. The default value is 1.

If only one parameter is passed into the slice (), then start and step have no value.

## The value returned from slice ()

The slice () function returns a slice object used to slice a string according to the corresponding indexes passed in the function.

## Example 1: Create a slice object to cut

```
# ch?a các ch? s? (0, 1, 2) print(slice(3)) # ch?a các ch? s?  
(1, 3) print(slice(1, 5, 2))
```

When you run the program, the output will be:

```
slice(None, 3, None) slice(1, 5, 2)
```

## Example 2: Receiving substring from a given string using slice object

```
pyString = 'Python' # ch?a các ch? s?  
(0, 1, 2) # vd P, y và t sObject = slice(3) print(pyString[sObject]) # ch?  
a các ch? s?  
(1, 3) # vd y và h sObject = slice(1, 5, 2) print(pyString[sObject])
```

Return value:

```
Pyt yh
```

## Example 3: Get substring from a given string using negative index

```
pyString = 'Python' # ch?a các ch? s?  
(-1, -2, -3) # vd n, o và h sObject = slice(-1, -4, -1) print(pyString[sObject])
```

The output will be:

```
noh
```

## Example 4: Slicing lists and tuples

```
pyList = ['P', 'y', 't', 'h', 'o', 'n'] pyTuple = ('P', 'y', 't', 'h', 'o', 'n')  
?a các ch? s? (0, 1, 2) # vd P, y và t sObject = slice(3) # c?  
t list print(pyList[sObject]) # ch?a các ch? s?  
(1, 3) # vd y và h sObject = slice(1, 5, 2) # c?  
t tuple print(pyTuple[sObject])
```

Results returned:

```
['P', 'y', 't'] ('y', 'h')
```

See also: Python built-in functions

You finished reading the article "**The slice () function in Python**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.