

Wool () function in Python

How is the syntax of len () function, what parameters do it have and how is it used? Invites you to read the track.

The len () function in Python returns the length (number of characters) of the specified string. How is the syntax of len () function, what parameters do it have and how is it used? Invites you to read the track.

Wool () function syntax in Python

```
len(chuoi)
```

Wool function parameters ()

Wool () has 1 parameter:

1. string : string (can be string, bytes, tuple, list, range, dictionary, set or frozen set) that you want to calculate the length.

Value returned from wool ()

1. The len function returns the number of characters of the input string. If the parameter or parameter is not passed, the program generates a TypeError exception.

Example 1: How does the function len () work with the parameters of tuple, list and range?

```
testList = []
print(testList, 'co do dai la', len(testList))

testList = [1, 2, 3]
print(testList, 'co do dai la', len(testList))

testTuple = (1, 2, 3)
print(testTuple, 'co do dai la', len(testTuple))

testRange = range(1, 10)
print('Do dai cua', testRange, 'la', len(testRange))
```

Run the program, the result is returned

```
[] co do dai la 0
[1, 2, 3] co do dai la 3
(1, 2, 3) co do dai la 3
Do dai cua range(1, 10) la 9
```

Example 2: How does the len () function work with parameters that are string and byte?

```
testString = ''
print('Do dai cua', testString, 'la', len(testString))

testString = 'Quantrimang'
print('Do dai cua', testString, 'la', len(testString))

# byte object
testByte = b'Quantrimang'
print('Do dai cua', testByte, 'la', len(testByte))

testList = [1, 2, 3]

# converting to bytes object
testByte = bytes(testList)
print('Do dai cua', testByte, 'la', len(testByte))
```

Running the program we get the result is:

```
Do dai cua la 0
Do dai cua Quantrimang la 11
Do dai cua b'Quantrimang' la 11
Do dai cua b'x01x02x03' la 3
```

Example 3: How does the len () function work with parameters that are dictionary and set?

```
testSet = {1, 2, 3}
print(testSet, 'co do dai la', len(testSet))

# Empty Set
testSet = set()
print(testSet, 'co do dai la', len(testSet))

testDict = {1: 'mot', 2: 'hai'}
print(testDict, 'co do dai la', len(testDict))

testDict = {}
print(testDict, 'co do dai la', len(testDict))

testSet = {1, 2}

# frozenSet
frozenTestSet = frozenset(testSet)
```

```
print(frozenTestSet, 'co do dai la', len(frozenTestSet))
```

The result returned is:

```
{1, 2, 3} co do dai la 3
set() co do dai la 0
{1: 'mot', 2: 'hai'} co do dai la 2
{} co do dai la 0
frozenset({1, 2}) co do dai la 2
```

When executed, len () calls the `__len__` method of the object, so you can also imagine the len () function as:

```
def len(s):
    return s.__len__()
```

Therefore, you can assign custom lengths to objects (if needed).

Example 4: How does the len () function work with custom objects?

```
class Session:
    def __init__(self, number = 0):
        self.number = number

    def __len__(self):
        return self.number
```

```
# ?? dài m?c ??nh là 0
# vi?t b?i TipsMake.com
s1 = Session()
print(len(s1))

# gán chi?u dài tùy ch?nh
s2 = Session(6)
print(len(s2))
```

Run the program, the returned result is:

```
0
6
```

Previous lesson: int () function in Python

Next lesson: function getattr () in Python

You finished reading the article "**len () 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.