The map () function in Python

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

The map () function built into Python works to browse all the elements of an iterable (list, tuple, dictionary .) through a given function and returns a list of results after execution.

So the syntax of the map () function like, what parameters it has and how to use it? Invites you to read the track.

Syntax of map () function in Python

```
map(function, iterable, .)
```

The parameters of the map () function

1. `function` The `function` executes for each element in the iterable.
2. `iterable` : a list, tuple, dictionary . want to browse.

You can pass multiple iterablecho map () functions.

Return value from map ()

The map () function browses all iterable elements through the function and returns a list of results after execution.

The value returned from map () is called map object. This object can be passed to list () functions (to create lists in Python), or set () (to create a new set of elements) .

Example 1: How does map () work?

```
def binhphuong(n): return n*n # viet boi TipsMake.com number = (25, 100, 225, 40
```

When you run the program, the output will be:

```
[25, 100, 225, 400]
```

In the above example, every element in the original tuple is square.

Example 2: How to use lambda functions with map ()

Because map () always needs parameters to pass, lambda functions are often used with map ().

In Python, lambda functions or anonymous functions are defined without a name. If normal functions are defined using the keyword `def` , then anonymous functions are defined using the keyword `lambda` .

Read more: Anonymous functions, Lambda in Python.

```
# viet boi TipsMake.com number = (5, 10, 15, 20) result = map(lambda x: x*x, numb
```

Running the program, the result is returned:

```
[25, 100, 225, 400]
```

The result is no different from **example 1** .

Example 3: Passing multiple iterator parameters to map () using lambda

In this example, the corresponding elements of the two lists are added.

```
num1 = [4, 5, 6] num2 = [5, 6, 7] result = map(lambda n1, n2: n1+n2, num1, num2)
```

The result is:

```
[9, 11, 13]
```

See also: Python built-in functions

You finished reading the article "**The map () 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.