

The object () function in Python

The object () function in Python returns an empty object. You cannot add new properties or methods to this object.

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

The object () function in Python returns an empty object. You cannot add new properties or methods to this object. This return object is the basis for all classes, and contains default built-in properties and methods for all classes.

The syntax of the object () function in Python:

```
o = object()
```

Parameters of the object () function

The object () function contains no parameters.

The value returned from object ()

The object () function returns an empty object, the object has no features (Featureless Object).

Example 1: how does object () work?

```
test = object() print(type(test)) print(dir(test))
```

When you run the program, the output will be:

```
['__class__', '__delattr__', '__dir__', '__doc__', '__eq__', '__format__', '__g
```

In the above code, we create the test object and use the built-in functions type () and dir () to get the object's valid types and properties. The result has no __dict__ in the output. Therefore, you cannot assign arbitrary properties to instances of this class.

See also: Python built-in functions

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