Type () function in Python

The built-in function type () in Python returns the type of object passed as parameter.

The built-in function type () in Python returns the type of object passed as parameter. The type () function is mainly used for debugging purposes.

Syntax

There are two different types of parameters that can be passed to the type () function, which are called single parameters and three parameters.

```
type(object)
type(name, bases, dict)
```

1. If the type of single parameter (object) is passed, the result will return the given object type.
2. If the third parameter (name, bases, dict) is passed, it returns a new type object.

type () with a single object parameter (single parameter)

If the type of single parameter (object) is passed, the result will return the given object type.

Example: How to get the object type?

```
numberList = [1, 2]
print(type(numberList))

numberDict = {1: 'one', 2: 'two'}
print(type(numberDict))

class Foo:
a = 0

InstanceOfFoo = Foo()
print(type(InstanceOfFoo))
```

Run the program, the result is:

If you want to test the object type, you can also use the Python function *isinstance (), because isinstance ()* function is also responsible for checking whether the given object is an instance of the subclass.

type () with three parameters name, bases, dict (parameter three)

If the third parameter (name, bases, dict) is passed, it returns a new type object.

1. `name` : class name will become attribute __name__.
2. `bases` : tuple of classes, corresponding to the __base__ attribute.
3. `dict` : a dictionary contains namespaces for the class, corresponding to the __dict__ attribute.

Example: Create object type using type ()

```
o1 = type('X', (object,), dict(a='Foo', b=12))

print(type(o1))
print(vars(o1))

class test:
a = 'Foo'
b = 12

o2 = type('Y', (test,), dict(a='Foo', b=12))
print(type(o2))
print(vars(o2))
```

Run the program, the result is:

```
{'a': 'Foo', 'b': 12, '__weakref__': objects>, '__dict__': , '__module__':
 '__main__', '__doc__': None}

{'a': 'Foo', 'b': 12, '__weakref__': objects>, '__dict__': , '__module__':
 '__main__', '__doc__': None}
{'a': 'Foo', 'b': 12, '__weakref__': objects>, '__dict__': , '__module__':
 '__main__', '__doc__': None}

 {'a': 'Foo', '__module__': '__main__', 'b': 12, '__doc__': None}
```

In the above example, the vars () function is used to return the __*dict*__ attribute . __*dict*__ is used to store writable properties of the object.

See also: Built-in Python functions

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