

图解

Python

可能是迄今为止

最易懂的Python视频教程



```
"""获取对象的信息之内置函数type()"""
```

```
.....
```

```
    内置函数type()用于获得指定对象的类型。
```

```
.....
```

```
.....
```

```
    实例对象的类型是其对应的类对象。
```

```
.....
```

```
class MyClass(object):  
    pass
```

```
mc = MyClass()
```

```
# mc的类型是MyClass, mc是MyClass的一个实例对象  
print(type(mc)) # <class '__main__.MyClass'>
```

```
# 整数对象18的类型是int, 整数对象18是int的一个实例对象  
print(type(18)) # <class 'int'>
```

```
# 字符串对象'abc'的类型是str, 字符串对象'abc'是str的一个实例对象  
print(type('abc')) # <class 'str'>
```

.....

类对象的类型是**type**，也就是说，类对象是**type**的一个实例对象。

.....

```
# 类对象MyClass的类型是type，类对象MyClass是type的一个实例对象
print(type(MyClass))      # <class 'type'>
```

```
# 类对象int的类型是type，类对象int是type的一个实例对象
print(type(int))          # <class 'type'>
```

```
# 类对象str的类型是type，类对象str是type的一个实例对象
print(type(str))          # <class 'type'>
```

.....

自定义函数对象的类型是**function**。

内置函数对象的类型是**builtin_function_or_method**。

.....

```
def do_sth():
    pass
```

```
print(type(do_sth)) # <class 'function'>
```

```
print(type(print)) # <class 'builtin_function_or_method'>
```

"""

可以使用运算符==判断某个对象的类型是否是指定的类型。

对于基本数据类型，可以直接使用其对应的类名；如果不是基本数据类型，需要使用标准库中的模块types中定义的变量。

"""

```
print(type(18) == int)      # True
print(type('abc') == str)  # True
```

```
# print(type(do_sth) == function)
# print(type(print) == builtin_function_or_method)
```

```
import types
print(type(do_sth) == types.FunctionType)    # True
print(type(print) == types.BuiltinFunctionType) # True
```