

图解

Python

可能是迄今为止

最易懂的Python视频教程



.....

Python是动态语言，所以，在实例对象或类对象创建之后，可以对其动态地绑定实例方法。

同一个类对象的不同实例对象所绑定的实例方法是相互独立的。也就是说，给一个实例对象绑定的实例方法，对于另一个实例对象是不起作用的。

为了能让一个类对象的所有实例对象都能调用某个实例方法，可以给类对象绑定该实例方法。

.....

定义一个函数作为实例方法

```
def do_sth(self):  
    print("do_sth()被调用了")
```

导入标准库types中的类MethodType

```
from types import MethodType
```

给实例对象mc动态地绑定实例方法

```
mc.do_sth = MethodType(do_sth, mc)
```

调用绑定的实例方法

```
mc.do_sth()          # do_sth()被调用了
```

```
mc2 = MyClass()
# mc2.do_sth()      # AttributeError: 'MyClass' object has no attribute 'do_sth'

# 再定义一个函数作为实例方法
def im3(self):
    print("im3()被调用了")

# 给类对象动态地绑定实例方法
MyClass.im3 = im3

# 类对象的所有实例对象都能调用给类对象动态绑定的实例方法
mc.im3()      # im3()被调用了
mc2.im3()     # im3()被调用了
```