

图解

Python

可能是迄今为止

最易懂的Python视频教程



```
"""类对象的特殊方法之__getitem__()"""
```

```
.....
```

对于自定义类对象的实例对象，在默认情况下，是不能像列表和字典那样使用中括号语法来操作数据的。

```
.....
```

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

```
mc = MyClass()  
# print(mc[3])      # TypeError: 'MyClass' object does not support indexing
```

```
.....
```

如果想让自定义类对象的实例对象可以像列表和字典那样，使用中括号语法来操作数据，必须在自定义类对象中实现以下特殊方法：

1. `__getitem__(self, key)`
当执行操作`obj[key]`时，会自动调用该特殊方法。
2. `__setitem__(self, key, value)`
当执行操作`obj[key] = value`时，会自动调用该特殊方法。
3. `__delitem__(self, key)`
当执行操作`del obj[key]`时，会自动调用该特殊方法。

```
.....
```

```
class MyDict(object):
    def __init__(self):
        self.data = {}

    def __getitem__(self, key):
        return self.data[key]

    def __setitem__(self, key, value):
        self.data[key] = value

    def __delitem__(self, key):
        del self.data[key]
```

```
md = MyDict()
```

```
md["one"] = 18
md["two"] = 32
print(md.data)    # {'one': 18, 'two': 32}
```

```
print(md["two"])    # 32
```

```
del md["two"]
print(md.data)    # {'one': 18}
```