

图解

Python

可能是迄今为止

最易懂的Python视频教程



```
"""重写"""
```

```
"""
```

如果子类对继承自父类的某个属性或方法不满意，可以在子类中对其进行重写从而提供自定义的实现，重写的方式为：在子类中定义与父类中同名的属性或方法（包括装饰器）。

子类重写父类的属性后，通过子类或其实例对象只能访问子类中重写后的属性，而无法再访问父类中被重写的属性。

子类重写父类的方法后，通过子类或其实例对象只能调用子类中重写后的方法，而无法再调用父类中被重写的方法。

父类中被重写的名为xxx的方法，在子类重写后的方法中可以通过`super().xxx()`进行调用。

```
"""
```

```
class ParentClass(object):
    ca = "ca (父类) "

    def __init__(self):
        print("__init__()被调用了 (父类) ")

    def im(self):
        print("im()被调用了 (父类) ")

    @classmethod
    def cm(cls):
        print("cm()被调用了 (父类) ")
```

```
class ChildClass(ParentClass):
    ca = "ca (子类) "

    def __init__(self):
        super().__init__()
        print("__init__()被调用了 (子类) ")

    def im(self):
        super().im()
        print("im()被调用了 (子类) ")

    @classmethod
    def cm(cls):
        super().cm()
        print("cm()被调用了 (子类) ")
```

```
cc = ChildClass()

print(ChildClass.ca)    # ca (子类)
print(cc.ca)            # ca (子类)

cc.im()                 # im()被调用了 (子类)

ChildClass.cm()         # cm()被调用了 (子类)
cc.cm()                 # cm()被调用了 (子类)
```

