

图解

Python

可能是迄今为止

最易懂的Python视频教程



.....

访问实例属性和类属性都可以通过"实例对象.属性名"的方式。当通过"实例对象.属性名"的方式访问属性时，会先查找指定的实例对象中有没有指定名称的实例属性，如果没有，再查找对应的类对象中有没有指定名称的类属性。所以，当通过"实例对象.属性名"的方式访问属性时，如果实例属性和类属性同名，实例属性会屏蔽掉类属性。

当通过"实例对象.属性名 = 属性值"的方式绑定属性时，这里的属性只表示实例属性（因为类属性没有这样的绑定方式），所以，只会查找指定的实例对象有没有绑定指定名称的实例属性，如果没有则进行绑定，如果已经绑定了，则对属性值进行修改。

.....

```
class Person(object):  
    age = 18
```

```
p = Person()
```

```
# 在实例对象p中没有找到名为age的实例属性，然后在类对象Person中找到了名为age的类属性  
print(p.age)      # 18  
# 打印名为age的类属性  
print(Person.age) # 18
```

```
# 给实例对象p绑定实例属性
p.age = 19
# 实例属性屏蔽掉了同名的类属性
print(p.age)      # 19
# 打印名为age的类属性
print(Person.age) # 18

# 删除实例对象p绑定的实例属性
del p.age
print(p.age)      # 18

p1 = Person()
p2 = Person()

# p1.age += 2等价于: p1.age = p1.age + 2
# 等号右边是属性访问, 整个赋值语句是实例属性的绑定
p1.age += 2
print(Person.age) # 18
print(p1.age)     # 20
print(p2.age)     # 18
```

```
# Person.age += 3等价于: Person.age = Person.age + 3
# 等号右边是类属性访问, 整个赋值语句是类属性的绑定
Person.age += 3
print(Person.age)    # 21
print(p1.age)        # 20
print(p2.age)        # 21
```