

# 私有属性和方法

在实际开发中，对象的某些属性或者方法可能只希望在对象的内部别使用，而不希望在外部的被访问到，这时就可以定义私有属性和私有方法。

## 定义方法

在定义属性或方法时，在属性名或者方法名前增加两个下划线 `__`，定义的就是私有属性或方法。

```
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age
        self.__money = 2000 # 使用 __ 修饰的属性，是私有属性

    def __shopping(self, cost):
        self.__money -= cost # __money 只能在对象内部使用
        print('还剩下%d元'%self.__money)

    def test(self):
        self.__shopping(200) # __shopping 方法也只能在对象内部使用

p = Person('张三', 18)
# print(p.__money) 这里会报错，不能直接访问对象内部的私有属性
p.test()
# p.__shopping() 这里会报错，__shopping 只能在对象内部使用，外部无法访问
```

## 访问私有属性和方法

私有属性不能直接使用，私有方法不能直接调用。但是，通过一些代码，我们也可以在外部访问一个对象的私有属性和方法。

### 直接访问

使用方式:在私有属性名或方法名前添加 `__类名`

```

class Person:
    def __init__(self,name,age):
        self.name = name
        self.age = age
        self.__money = 2000

    def __shopping(self, cost):
        self.__money -= cost

p = Person('李四',20)
print(p._Person__money) # 使用对象名._类名__私有属性名 可以直接访问对象的私有属性
p._Person__shopping(100) # 使用对象名._类名__函数名 可以直接调用对象的私有方法
print(p._Person__money)

```

注意：在开发中，我们**强烈不建议使用** `对象名._类名__私有属性名` 的方式来访问对象的私有属性！

## 定义方法访问私有变量

在实际开发中，如果对象的变量使用了 `__` 来修饰，就说明它是一个私有变量，不建议外部直接使用和修改。如果非要修改这个属性，可以使用定义 `get` 和 `set` 方法这种方式来实现。

```

class Person:
    def __init__(self,name,age):
        self.name = name
        self.age = age
        self.__money = 2000 # __money 是私有变量，外部无法访问

    def get_money(self): # 定义了get_money 方法，在这个方法里获取到 __money
        return self.__money # 内部可以访问 __money 变量

    def set_money(self,money): # 定义了set_money 方法，在这个方法里，可以修改 __money
        self.__money = money

p = Person('王五',21)

# 外部通过调用 get_money 和 set_money 这两个公开方法获取和修改私有变量
print(p.get_money())
p.set_money(8000)
print(p.get_money())

```