class MyClass:

val1 = "Value 1" #类属性

def \_\_init\_\_(self):

self.val2 = "Value 2" #实例属性

@staticmethod

def staticmd():

print("静态方法，无法访问Value1和Value2")

@classmethod

def classmd(cls):

print('类方法，类：'+ str(cls)+ ",val1:"+ cls.val1+ ",无法访问val2的值") #调用类属性：cls.类属性名

>>> mc = MyClass() # 实例化

>>> mc.staticmd() # 实例调用静态方法，无法访问实例变量val1和val2

>>>

静态方法，无法访问val1和val2

>>> mc.classmd() # 实例调用类方法，注意，这里访问的是类MyClass的变量val1的值，不是实例化后mc的实例变量val1，这里容易混淆，往下看就会明白。val2一直是实例变量，所以无法访问

>>>

类方法，类：\_\_main\_\_.MyClass，val1：Value 1，无法访问val2的值

>>> MyClass.staticmd() # 类直接调用静态方法，结果同上面的实例调用，无论是类变量还是实例变量都无法访问

>>>

静态方法，无法访问val1和val2

>>> MyClass.classmd() # 类直接调用类方法，结果同上面的实例调用

>>>

类方法，类：\_\_main\_\_.MyClass，val1：Value 1，无法访问val2的值

>>> mc.val1 = 'Value changed' # 改变实例变量val1的值

>>> mc.classmd() # 实例调用类方法，注意到cls.val1的值没变，所以，这时的cls.val1是类变量val1，而非实例变量val1

>>>

类方法，类：\_\_main\_\_.MyClass，val1：Value 1，无法访问val2的值

>>> MyClass.classmd() # 类直接调用类方法，结果同上面的实例调用

>>>

类方法，类：\_\_main\_\_.MyClass，val1：Value 1，无法访问val2的值

>>> MyClass.val1 = 'Class Value changed' # 改变类变量val1的值

属性分为实例属性与类属性

方法分为普通方法，类方法，静态方法

一：属性：

　　尽量把需要用户传入的属性作为实例属性，而把同类都一样的属性作为类属性。实例属性在每创造一个实例时都会初始化一遍，不同的实例的实例属性可能不同，不同实例的类属性都相同。从而减少内存。

　　1：实例属性：

　　　　最好在\_\_init\_\_(self,...)中初始化

　　　　内部调用时都需要加上self.

　　　　外部调用时用instancename.propertyname

　　2:类属性：

　　　　在\_\_init\_\_()外初始化

　　　　在内部用classname.类属性名调用

　　　　外部既可以用classname.类属性名又可以用instancename.类属性名来调用

　　3：私有属性：

　　　　1）：单下划线\_开头：只是告诉别人这是私有属性，外部依然可以访问更改

　　　　2）：双下划线\_\_开头：外部不可通过instancename.propertyname来访问或者更改

　　　　　　实际将其转化为了\_classname\_\_propertyname

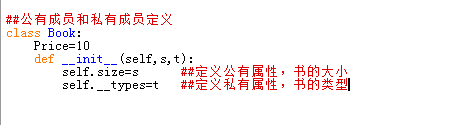
**Python中的实例属性、类属性**

**类Class中成员变量有两种属性分别是：实例属性和类属性。类属性是在类中方法（也就是类中的函数）之外但又在类之中定义的属性；而实例属性是在构造函数中定义的（\_\_init\_\_）,定义时候以self作为前缀。类属性在所有实例之间共享。在类内部和类外部都可以通过“类.类属性” 来访问。**

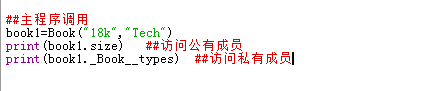
**那么，类属性和实例属性分别怎么访问呢？在主程序中，如果想要访问实例属性，因为实例属性属于对象，只能通过对象名访问；如果要访问类属性，因为类属性属于类，可以通过类名访问，同时也可以通过对象名访问。**

**Python中的私有成员与公有成员**

**类的属性在定义的时候，如果以两个短下滑线“\_\_”开头则表示是私有属性，否则就是公有属性。**



**例如，上面小程序中，书的大小是公有属性，而书的类型是私有属性。对公有属性可以公开使用，既可以在类内部访问，也可以在类外部访问。私有成员只能在类内部使用，但也可以通过特殊的方法进行访问，即“对象名.\_类名+私有成员”的方式。在主程序中对私有成员和公有成员的调用实例如下。**



**Python中的私有方法、公有方法和静态方法**

二：方法

　　1：普通类方法：

　　　　def fun\_name(self,...):

　　　　　　pass

　　　　外部用实例调用

　　2：静态方法：@staticmethod

　　　　　　不能访问实例属性！！！   参数不能传入self！！！

　　　　　　与类相关但是不依赖类与实例的方法！！

　　3:类方法：@classmethod

　　　　　　不能访问实例属性！！！   参数必须传入cls！！！

　　　　　　必须传入cls参数（即代表了此类对象-----区别------self代表实例对象），并且用此来调用类属性：cls.类属性名

　　\*静态方法与类方法都可以通过类或者实例来调用。其两个的特点都是不能够调用实例属性

**公有方法：通过对象名直接调用。**

**定义公有方法：def namemethod()。调用：对象名.公有方法**

**私有方法：不能通过对象名直接调用。只能在属于对象的方法中通过self调用或者在外部通过特殊方法调用。**

**定义私有方法：def \_\_namemethod()。调用：self.\_namemethod。**

**静态方法：可以通过类名和对象名调用，但不能直接访问属于对象的成员，只能访问属于类的成员。**

**定义静态方法:**

**@staticmethod**

**def methodname()。调用：类名.方法名**