Python类变量和实例变量（类属性和实例属性）

无论是类属性还是类方法，都无法像普通变量或者函数那样，在类的外部直接使用它们。我们可以将类看做一个独立的空间，则类属性其实就是在类体中定义的变量，类方法是在类体中定义的函数。

前面章节提到过，在类体中，根据变量定义的位置不同，以及定义的方式不同，类属性又可细分为以下 3 种类型：

1. 类体中、所有函数之外：此范围定义的变量，称为类属性或类变量；
2. 类体中，所有函数内部：以“self.变量名”的方式定义的变量，称为实例属性或实例变量；
3. 类体中，所有函数内部：以“变量名=变量值”的方式定义的变量，称为局部变量。

不仅如此，类方法也可细分为实例方法、静态方法和类方法，后续章节会做详细介绍。

那么，类变量、实例变量以及局部变量之间有哪些不同呢？接下来就围绕此问题做详细地讲解。

**类变量（类属性）**

类变量指的是在类中，但在各个类方法外定义的变量。举个例子：

**class** CLanguage :

# 下面定义了2个类变量

name = "C语言中文网"

add = "http://c.biancheng.net"

# 下面定义了一个say实例方法

**def** say(self, content):

**print**(content)

上面程序中，name 和 add 就属于类变量。  
  
类变量的特点是，所有类的实例化对象都同时共享类变量，也就是说，类变量在所有实例化对象中是作为公用资源存在的。类方法的调用方式有 2 种，既可以使用类名直接调用，也可以使用类的实例化对象调用。  
  
比如，在 CLanguage 类的外部，添加如下代码：

#使用类名直接调用

**print**(CLanguage.name)

**print**(CLanguage.add)

#修改类变量的值

CLanguage.name = "[Python](http://c.biancheng.net/python/)教程"

CLanguage.add = "http://c.biancheng.net/python"

**print**(CLanguage.name)

**print**(CLanguage.add)

程序运行结果为：

C语言中文网  
http://c.biancheng.net  
Python教程  
http://c.biancheng.net/python

可以看到，通过类名不仅可以调用类变量，也可以修改它的值。  
  
当然，也可以使用类对象来调用所属类中的类变量（此方式不推荐使用，原因后续会讲）。例如，在 CLanguage 类的外部，添加如下代码：

clang = CLanguage()

**print**(clang.name)

**print**(clang.add)

运行程序，结果为：

C语言中文网  
http://c.biancheng.net

注意，因为类变量为所有实例化对象共有，通过类名修改类变量的值，会影响所有的实例化对象。例如，在 CLanguage 类体外部，添加如下代码：

**print**("修改前，各类对象中类变量的值：")

clang1 = CLanguage()

**print**(clang1.name)

**print**(clang1.add)

clang2 = CLanguage()

**print**(clang2.name)

**print**(clang2.add)

**print**("修改后，各类对象中类变量的值：")

CLanguage.name = "Python教程"

CLanguage.add = "http://c.biancheng.net/python"

**print**(clang1.name)

**print**(clang1.add)

**print**(clang2.name)

**print**(clang2.add)

程序运行结果为：

修改前，各类对象中类变量的值：  
C语言中文网  
http://c.biancheng.net  
C语言中文网  
http://c.biancheng.net  
修改后，各类对象中类变量的值：  
Python教程  
http://c.biancheng.net/python  
Python教程  
http://c.biancheng.net/python

显然，通过类名修改类变量，会作用到所有的实例化对象（例如这里的 clang1 和 clang2）。

注意，通过类对象是无法修改类变量的。通过类对象对类变量赋值，其本质将不再是修改类变量的值，而是在给该对象定义新的实例变量（在讲实例变量时会进行详细介绍）。

值得一提的是，除了可以通过类名访问类变量之外，还可以动态地为类和对象添加类变量。例如，在 CLanguage 类的基础上，添加以下代码：

clang = CLanguage()

CLanguage.catalog = 13

**print**(clang.catalog)

运行结果为：

13

**实例变量（实例属性）**

实例变量指的是在任意类方法内部，以“self.变量名”的方式定义的变量，其特点是只作用于调用方法的对象。另外，实例变量只能通过对象名访问，无法通过类名访问。  
  
举个例子：

**class** CLanguage :

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

self.name = "C语言中文网"

self.add = "http://c.biancheng.net"

# 下面定义了一个say实例方法

**def** say(self):

self.catalog = 13

此 CLanguage 类中，name、add 以及 catalog 都是实例变量。其中，由于 \_\_init\_\_() 函数在创建类对象时会自动调用，而 say() 方法需要类对象手动调用。因此，CLanguage 类的类对象都会包含 name 和 add 实例变量，而只有调用了 say() 方法的类对象，才包含 catalog 实例变量。  
  
例如，在上面代码的基础上，添加如下语句：

clang = CLanguage()

**print**(clang.name)

**print**(clang.add)

#由于 clang 对象未调用 say() 方法，因此其没有 catalog 变量，下面这行代码会报错

#print(clang.catalog)

clang2 = CLanguage()

**print**(clang2.name)

**print**(clang2.add)

#只有调用 say()，才会拥有 catalog 实例变量

clang2.say()

**print**(clang2.catalog)

运行结果为：

C语言中文网  
http://c.biancheng.net  
C语言中文网  
http://c.biancheng.net  
13

前面讲过，通过类对象可以访问类变量，但无法修改类变量的值。这是因为，通过类对象修改类变量的值，不是在给“类变量赋值”，而是定义新的实例变量。例如，在 CLanguage 类体外，添加如下程序：

clang = CLanguage()

#clang访问类变量

**print**(clang.name)

**print**(clang.add)

clang.name = "Python教程"

clang.add = "http://c.biancheng.net/python"

#clang实例变量的值

**print**(clang.name)

**print**(clang.add)

#类变量的值

**print**(CLanguage.name)

**print**(CLanguage.add)

程序运行结果为：

C语言中文网  
http://c.biancheng.net  
Python教程  
http://c.biancheng.net/python  
C语言中文网  
http://c.biancheng.net

显然，通过类对象是无法修改类变量的值的，本质其实是给 clang 对象新添加 name 和 add 这 2 个实例变量。

类中，实例变量和类变量可以同名，但这种情况下使用类对象将无法调用类变量，它会首选实例变量，这也是不推荐“类变量使用对象名调用”的原因。

另外，和类变量不同，通过某个对象修改实例变量的值，不会影响类的其它实例化对象，更不会影响同名的类变量。例如：

**class** CLanguage :

name = "xxx" #类变量

add = "http://" #类变量

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

self.name = "C语言中文网" #实例变量

self.add = "http://c.biancheng.net" #实例变量

# 下面定义了一个say实例方法

**def** say(self):

self.catalog = 13 #实例变量

clang = CLanguage()

#修改 clang 对象的实例变量

clang.name = "python教程"

clang.add = "http://c.biancheng.net/python"

**print**(clang.name)

**print**(clang.add)

clang2 = CLanguage()

**print**(clang2.name)

**print**(clang2.add)

#输出类变量的值

**print**(CLanguage.name)

**print**(CLanguage.add)

程序运行结果为：

python教程  
http://c.biancheng.net/python  
C语言中文网  
http://c.biancheng.net  
xxx  
http://

不仅如此，Python 只支持为特定的对象添加实例变量。例如，在之前代码的基础上，为 clang 对象添加 money 实例变量，实现代码为：

clang.money = 30

**print**(clang.money)

**局部变量**

除了实例变量，类方法中还可以定义局部变量。和前者不同，局部变量直接以“变量名=值”的方式进行定义，例如：

**class** CLanguage :

# 下面定义了一个say实例方法

**def** count(self,money):

sale = 0.8\*money

**print**("优惠后的价格为：",sale)

clang = CLanguage()

clang.count(100)

通常情况下，定义局部变量是为了所在类方法功能的实现。需要注意的一点是，局部变量只能用于所在函数中，函数执行完成后，局部变量也会被销毁。