　　学完类后，感觉类和函数区别并不大，都是避免重复代码，方便维护、调用，那么为什么还要有类的存在，用函数不就好了么，答案是否定的，存在即是合理，那这个理在哪里？

　　个人总结在以下几点：

　　1.模块化设计：定义class，定义好域和方法，具体对象直接调用域和方法即可；

　　2.函数只能被运行，返回或者不返回值，类可以有自己的变量和方法。

　　综上：需要把功能和数据归在一起，用类更方便。

#假设有类B，其中有变量var，方法fun1，类方法fun2。

#调用方法是：

shili = B()

aa=shili.var #引用变量

bb=shili.fun1() #引用函数

cc=shili.fun2() #引用类方法

　　以下是基本用法总结：

[复制代码](javascript:void(0);)

1 class Teacher(object):

2 school='GTA 学院'#类变量

3 def \_\_init\_\_(self,name,sex,age):#构造函数，在创建类时会初始化

4 self.name=name#实例变量

5 self.sex=sex

6 self.age=age

7 self.\_\_money=123#私密变量，只能在内部调用，外部调用失败

8 self.describe()#调用方法

9 self.\_\_simi()#调用私密方法，只能在这里(实例内部)调用，外部调用会报错

10 def \_\_del\_\_(self):#析构函数，在结束实例运行时会调用,用于关闭数据库连接等，在实例运行完会自动运行，不需要调用

11 print('end')

12 def describe(self):#实例方法

13 print('%s老师性别是%s，年龄是%s'%(self.name,self.sex,self.age))

14 def \_\_simi(self):#私密方法，外部不能调用，只能实例内调用

15 print('不能说的秘密')

16 print(self.\_\_money)#私密变量，在内部调用，这里可以调用，外部调用失败

17

18 class Student(Teacher):#继承Teacher，Teacher里有的所有方法、变量都继承过来

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

20 print(self.school)

21 A=Teacher(1,2,3)

22 B=Student()

23

24 #类方法

25 #通过类名可以直接调用，不需要实例化

26 @classmethod

27 def fuc(cls):

28 pass

29 #静态方法

30 #就是写在类里面的一个普通函数而已。

31 @staticmethod

32 def myFunc():

33 pass

34 #实例方法

35 #必须实例化之后才可以调用的方法

36 #类里面的函数，如果第一个参数是self，那么就是一个实例方法

37

38 #属性方法

39 #把一个方法变成一个变量

40 @property

41 def add(self):

42 return 18

43 #属性方法也是一个实例方法，需要实例化才能使用。

44 #属性方法同样可以通过实例方法实现

45 #如果你使用属性方法的话，那么必须得有返回值，而且不能有入参。