类的方法必须有一个二外的第一个参数名称叫self

创建类：

class employee:

empCount=0

def \_\_init\_\_(self,name,salary):

self.name=name

self.salary=salary

employee.empCount+=1

def displaycount(self):

print employee.empCount

def displayemployee(self):

print name, salary

创建实例对象：

emp1=employee(‘annie’ 2000)  
emp2=employee(‘janne’,3000)

访问属性：使用.访问对象的属性，使用类的名称访问类变量

emp1.displayemployee()

emp2.displayemployee()

print employee.empCount

访问属性：

getattr(obj, name[, default]) : 访问对象的属性。 getattr(emp1, ‘ age’)

hasattr(obj,name) : 检查是否存在一个属性。 hasattr(emp1, ‘age’)

setattr(obj,name,value) : 设置一个属性。如果属性不存在，会创建一个新属性。 Setattr(emp1, ‘age’,8)

delattr(obj, name) : 删除属性。 Delattr(emp1, ‘age’)

内置类属性：

\_\_dict\_\_ : 类的属性（包含一个字典，由类的数据属性组成）

\_\_doc\_\_ :类的文档字符串

\_\_name\_\_: 类名

\_\_module\_\_: 类定义所在的模块（类的全名是'\_\_main\_\_.className'，如果类位于一个导入模块mymod中，那么className.\_\_module\_\_ 等于 mymod）

\_\_bases\_\_ : 类的所有父类构成元素（包含了一个由所有父类组成的元组）

Python 对象销毁：

编译器再合适的时间销毁对象，如果class里面有定义\_\_del\_\_()则执行里面的语句。

类的继承：

Class child(parent)

1：在继承中基类的构造（\_\_init\_\_()方法）不会被自动调用，它需要在其派生类的构造中亲自专门调用。

2：在调用基类的方法时，需要加上基类的类名前缀，且需要带上self参数变量。区别于在类中调用普通函数时并不需要带上self参数

3：Python总是首先查找对应类型的方法，如果它不能在派生类中找到对应的方法，它才开始到基类中逐个查找。（先在本类中查找调用的方法，找不到才去基类中找）。

4.子类不能子类的方法访问父类的私有属性