python 基础 08：面向对象的基本概念

Python 使用类(class)和对象(object)，进行面向对象（object-oriented programming，简称 OOP）的编程。

面向对象的最主要目的是提高程序的重复使用性。我们这么早切入面向对象编程的原因是，Python 的整个概念是基于对象的。了解 OOP 是进一步学习 Python 的关键。

下面是对面向对象的一种理解，基于分类。

1、相近对象，归为类

在人类认知中，会根据属性相近把东西归类，并且给类别命名。比如说，鸟类的共同属性是有羽毛，通过产卵生育后代。任何一只特别的鸟都在鸟类的原型基础上的。

面向对象就是模拟了以上人类认知过程。在 Python 语言，为了听起来酷，我们把上面说的“东西”称为对象（object）。

先定义鸟类

class Bird(object): have_feather = True way_of_reproduction = 'egg'

我们定义了一个类别（class），就是鸟（Bird）。在隶属于这个类比的语句块中，我们定义了两个变量，一个是有羽毛（have_feather），一个是生殖方式（way_of_reproduction）,这两个变量对应我们刚才说的属性（attribute）。我们暂时先不说明括号以及其中的内容，记为问题 1。

假设我养了一只小鸡，叫 summer。它是个对象，且属于鸟类。使用前面定义的类：

summer = Bird()print summer.way_of_reproduction

通过第一句创建对象，并说明 summer 是类别鸟中的一个对象，summer 就有了鸟的类属性，对属性的引用是通过 对象.属性（object.attribute） 的形式实现的。

可怜的 summer，你就是个有毛产的蛋货，好不精致。

2、动作

日常认知中，我们在通过属性识别类别的时候，有时根据这个东西能做什么事情来区分类别。比如说，鸟会移动。这样，鸟就和房屋的类别区分开了。这些动作会带来一定的结果，比如移动导致位置的变化。

这样的一些“行为”属性为方法（method）。Python 中通过在类的内部定义函数，来说明方法。

```
class Bird(object):
have_feather = True
way_of_reproduction = 'egg'
def move(self, dx, dy):            position = [0,0]
    position[0] = position[0] + dx          position[1] = position[1] + dy
    return position
```

summer = Bird()print 'after move:',summer.move(5,8)

我们重新定义了鸟这个类别。鸟新增一个方法属性，就是表示移动的方法 move。（我承认这个方法很傻，你可以在看过下一讲之后定义个有趣些的方法）

（它的参数中有一个 self，它是为了方便我们引用对象自身。方法的第一个参数必须是 self，无论是否用到。有关 self 的内容会在下一讲展开）

另外两个参数，dx, dy 表示在 x、y 两个方向移动的距离。move 方法会最终返回运算过的 position。

在最后调用 move 方法的时候，我们只传递了 dx 和 dy 两个参数，不需要传递 self 参数（因为 self 只是为了内部使用）。

我的 summer 可以跑了。

3、子类

类别本身还可以进一步细分成子类

比如说，鸟类可以进一步分成鸡，大雁，黄鹂。

在 OOP 中，我们通过继承(inheritance)来表达上述概念。

class Chicken(Bird): way_of_move = 'walk' possible_in_KFC = True class Oriole(Bird): way_of_move = 'fly' possible_in_K

FC = False summer = Chicken()print summer.have_featherprint summ er.move(5,8)

新定义的鸡（Chicken）类的，增加了两个属性：移动方式（way_of_move），可能在 KFC 找到（possible_in_KFC）

在类定义时，括号里为了 Bird。这说明，Chicken 是属于鸟类（Bird）的一个子类，即 Chicken 继承自 Bird。自然而然，Bird 就是 Chicken 的父类。Chicken 将享有 Bird 的所有属性。尽管我只声明了 summer 是鸡类，它通过继承享有了父类的属性（无论是变量属性 have_feather 还是方法属性 move）

新定义的黄鹂(Oriole)类，同样继承自鸟类。在创建一个黄鹂对象时，该对象自动拥有鸟类的属性。

通过继承制度，我们可以减少程序中的重复信息和重复语句。如果我们分别定义两个类，而不继承自鸟类，就必须把鸟类的属性分别输入到鸡类和黄鹂类的定义中。整个过程会变得繁琐，因此，面向对象提高了程序的可重复使用性。

（回到问题 1, 括号中的 object，当括号中为 object 时，说明这个类没有父类（到头了））

将各种各样的东西分类，从而了解世界，从人类祖先开始，我们就在练习了这个认知过程，面向对象是符合人类思维习惯的。所谓面向过程，也就是

执行完一个语句再执行下一个，更多的是机器思维。通过面向对象的编程，我们可以更方便的表达思维中的复杂想法。

4、总结

将东西根据属性归类（将 object 归为 class）

方法是一种属性，表示动作

用继承来说明父类-子类关系。子类自动具有父类的所有属性。

self 代表了根据类定义而创建的对象。

建立对一个对象： 对象名 = 类名()

引用对象的属性： object.attribute