Python基础入门升级版

火热开团

从零开始学 Python

(四)、面向对象编程

七月在线 David

2017年12月3日

复习

- 函数的意义与创建
- 变量的作用域与生命周期
- BIFs内的三个高阶函数理解及应用
- 偏函数与回调函数，递归函数，闭包问题
- 装饰器

本节课程目标

- 理解面向对象的基本思想
- 理解类与对象关系
- 了解类的结构与成员
- 熟悉Python中的魔法函数
- 能按功能要求完成类的设计与实例化对象等操作

一切皆对象

- **关于类与对象操作的BIFs**
  - type() 返回对象类型
  - id(), 查看对象id
  - dir(), 查看对象下变量及函数
  - issubclass(),isinstance(), super(),类,实例,调父类
  - hasattr(),getattr(),setattr(),delattr()类属性操作
  - globals(),locals(), 全局与局部的名称空间
  - import(),reload(), 模块的导入与重载

面向对象OOP

- 基本概念:
  - 面向过程与面向对象

- OOP的核心思想:
  - 类：从近似的对象中抽象出类
  - 对象：然后再从类出实例化出对象

类的创建与实例化

- 创建类
  - class关键字
  - 指定继承
  - 定义类的成员
    - 数据成员
      - 类变量
      - 实例变量
    - 方法成员
      - 类方法
      - 实例方法
- 实例化类

类的内部结构

- **数据成员：** [用于处理类及实例对象的相关数据]
  - 类变量：在类中且在函数体外,实例之间共享
  - 实例变量：定义在方法中，作用于当前实例的类

- **方法成员**（在类中定义的函数叫方法）：
  - 类方法
    - 定义时需要使用@classmethod装饰器，第一个参数为cls
  - 实例方法
    - 绑定到实例的方法，第一个参数为self,
  - 静态方法[普通方法]
    - 定义的时候使用@staticmethod装饰器。
    - 静态方法没有参数限制，不需要实例参数self和类参数cls
    - 静态法可以通过类名访问，也可以通过实例访问。

类的继承与多态

- **继承**
  - 创建一个类时可以从新开始，也可以从已经有的类继承下来
  - super()子调用父类的方法

- **多态**
  - 因为类具有继承关系，子类可以向上转型被看做是父类的类型，比如无论是战士还是快递员，都是人类。
  - 有了继承关系，子类可以继承父类的所有方法和属性，当然也可以重载父类的成员函数及属性。
  - 例如，当子类（直升机）和父类（飞机）都存在相同的fly()方法时，子类的fly()覆盖了父类的fly()，在运行时就总是会调用子类的fly()。
  - 这就是继承带来的多态。

访问控制

- **访问控制**
- Python没有像其它语言有访问控制的关键字，例如private、protected等等。Python通过命名约定来实现的访问控制
  - 对模块级的控制，通过在标识符前加单下划线_实现。
  - 对类内部的属性及方法，通过在在标识符前加双下划线__来实现的私有化
    - 类中两个双下划线包裹的属性或方法为特殊方法

魔法方法Magic Method

- **魔术方法:**
  - 魔法方法就是可以给你的类增加魔力的特殊方法，如果你的对象实现（重载）了这些方法中的某一个，那么这个方法就会在特殊的情况下被 Python 所调用，你可以定义自己想要的行为，这些会自动发生。
  - 它们经常是两个下划线包围来命名的

模块module

- 模块定义:
  - 一个.py文件
  - 包含了对象定义与语句
- 模块作用:
  - 用来从逻辑上组织代码
- 模块使用:
  - 搜索路径（标准模块，自定义与第三方模块)路径
    - 搜索路径设置(修改sys.path(sys.path.append()), 设置环境变量)
  - 导入方法
    - import test#作为模块空间导入
    - from ** import * #指定模块下具体的类，对象导入,并入当前空间
    - from *** import ***#将模块下所有对象导入，并入当前空间

包package

- **包定义:**
  - 含有__init__.py文件夹被称为包，__init__.py文件用于标识当前文件夹是一个包。（该文件可以为空,但必须有）
  - 包用于组织模块，通常把功能相近的模块进行再次封装成为包。
- **包的目录结构:**
  - 模块
  - 子包
    - 子包下的子包
- **包的安装，导入与访问**
  - 包的安装(pip,conda)
  - 不同的导入方式(假设包名为test)
    - import test#导入__init__.py这个moduel。
    - from test import *#导入__init__.py这个moduel下所有对象导入到当前空间。
    - from test.test_level1.test_level2 import test_level2#导入的层次目录下的模块。还是模块

练习1 ：完成简单类的设计

- **创建一个名为phone的类**
  - 1）类实例成员变量有screeen_size、price、brand
  - 2）给成员变量创建 访问及设置 方法
  - 3）定义play方法，功能为打印：play game
  - 4）定义sendMessage方法，功能为打印：text message
  - 5）定义powerOff方法，功能为打印：power off
  - 6）创建get_info方法，打印对象的相关信息
  - 7）实例化类phone()，对象名为phone1, 屏幕尺寸为5.5,价格为6288, 品牌为apple
  - 8）调用3，4，5，6方法，查看结果
  - 9）实例化类phone()，对象名为phone2, 屏幕尺寸为5,价格为1999,品牌为mi.
  - 10）调用3，4，5，6方法，查看结果

练习2 ： 完成简单类的设计

- **设计一个公司类，完成以下要求，并实例化不同对象进行验证**

- **类变量**
- - 类下公司的总个数，类下实例的名称列表

- **类方法**
- - 返回公司类共有多少个公司实例
- - 返回公司类的公司实例有名称列表

- **实例变量**
- - 公司名，简介，利润，销售额，总成本，雇员姓名，雇员列表

- **实例方法：**
- - 招聘人才（每招一个人会有成本产生，影响该实例雇员列表，人数，总成本）
- - 解雇人员（每解雇一个人会有成本产生，影响该实例雇员列表，人数，总成本）
- - 公司广告推广(影响该实例总成本)
- - 交社保(按公司雇员总人数计算，影响该实例总成本)
- - 交税(按公司雇员总人数计算，影响该实例总成本)
- - 销售（按销售件数*价格计算销售额，利润按销售额*利润率进行计算利润。）
- - 获取公司雇员列表
- - 获取公司净利润

练习3 ：设计类及其继承类

- **设计一个叫cinema的电影院的类，包含：**
  - 类方法：
    - 初始化方法，并初始相应的类变量实例变量
    - getSales（获取全部电影院实际销售的方法）的电影院
  - 实例方法：
    - saleTickets方法，要求更新实例的销售总额及类的销售总额
  - 类变量：
    - 类下所有实例电影院的数量，销售额总和
  - 实例变量：
    - 电影院名称，位置，销售额

- **创建一个miniCinema迷你电影院的类，继承自cinema类**
  - 重写卖票方法，大于100元的票价加入打9折扣的功能。
  - 对miniCinema实例进行调用体现多态性

练习4 ：导入包完成遍历目录

- 导入os包
- 创建递归循环完成一个目录的遍历
  - 如果下还有一个目录，依次进行遍历

练习5 ：练习使用PIL包

- 导入PIL包
  - 如果没有该包，请自行安装后导入

- 将一张图片进行尺寸缩放,颜色调整后保存。
  - （dir,help函数查看库）

作业1 ：新建类并修改其实例的切片访问

- Python中对序列的切片访问，是左闭右开。
- 现请您
  - 创建一个可被迭代的类
  - 改变默认的切片访问方式为左闭右闭

  - 举例:c1=youclassname([0,1,2,3,4,5,6,7,8,9]
  - c1[2:7] 返回2,3,4,5,6,7