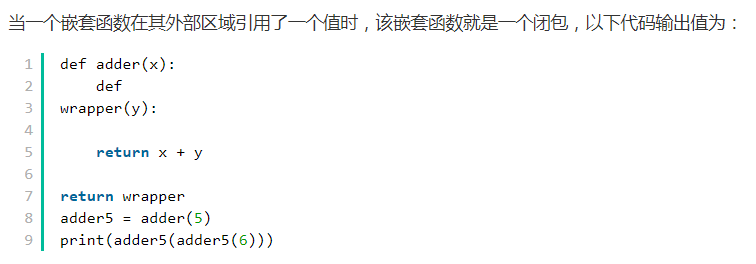
# python题目总结

## 嵌套函数



1、如果在一个函数的内部定义了另一个函数，外部的我们叫他外函数，内部的我们叫他内

函数。

2. 在一个外函数中定义了一个内函数，内函数里运用了外函数的临时变量，并且外函数的返回值是内函数的引用。这样就构成了一个闭包。

3. 一般情况下，如果一个函数结束，函数的内部所有东西都会释放掉，还给内存，局部变量都会消失。但是闭包是一种特殊情况，如果外函数在结束的时候发现有自己的临时变量将来会在内部函数中用到，就把这个临时变量绑定给了内部函数，然后自己再结束。

具体就是5+6+5=16

## 文件读取

read:读取整个文件，内容为字符串

readline：返回下一行，内容为字符串；

readlines:返回list,保存有每行的数据。

## python中复数

1 虚数不能单独存在，它们总是和一个值为 0.0 的实数部分一起来构成一个复数。

2复数由实数部分和虚数部分构成

3表示虚数的语法： real+imagj

4 实数部分和虚数部分都是浮点数

5 虚数部分必须有后缀 j 或 J

6 Python2 与 Python3 均不支持复数比较大小

## 向下取整

print math.floor(5.5) 向下取整

math.ceil(2.3) 向上取整

Python2 支持数字与字符串之间的比较，而 Python3 则不支持，ASCII值得比较

## python scket操作

1. **sk.recv(bufsize[,flag])**:接受套接字的数据。数据以字符串形式返回，bufsize指定最多可以接收的数量。flag提供有关消息的其他信息，通常可以忽略。
2. **recvfrom(bufsize[.flag]):**与recv()类似，但返回值是（data,address）。其中data是包含接收数据的字符串，address是发送数据的套接字地址

3、**getsockname():**返回套接字自己的地址。通常是一个元组(ipaddr,port)

4、**connect(address):**连接到address处的套接字。一般，address的格式为元组（hostname,port）,如果连接出错，返回socket.error错误

5、**listen(backlog):**开始监听传入连接。backlog指定在拒绝连接之前，可以挂起的最大连接数量

## 字典的相关操作

字典的键值必须是不可变类型，如数字，字符串，元组，而列表是可变类型。

字典本身是可变数据类型，字典的键记住两点：唯一：同一个字典中的键必须唯一，如果出现多个相同的键，则最新的键会被记住；不可变：键是不可变数据类型，使用可以是int,string,float,tuple

## isinstance函数，继承

abc isinstance（object，classinfo），用于判断object是否是classinfo的一个实例，或者object是否是classinfo类的子类的一个实例，如果是返回True. issubclass（class，classinfo），用于判断class是否是classinfo类的子类，如果是返回True.

python中继承的相关概念：<https://www.cnblogs.com/bigberg/p/7182741.html>

1. 类的概述

面向对象编程 (OOP) 语言的一个主要功能就是“继承”。继承是指这样一种能力：它可以使用现有类的所有功能，并在无需重新编写原来的类的情况下对这些功能进行扩展。

通过继承创建的新类称为“子类”或“派生类”，被继承的类称为“基类”、“父类”或“超类”，继承的过程，就是从一般到特殊的过程。在某些 OOP 语言中，一个子类可以继承多个基类。但是一般情况下，一个子类只能有一个基类，要实现多重继承，可以通过多级继承来实现。

　　继承概念的实现方式主要有2类：实现继承、接口继承。

实现继承是指使用基类的属性和方法而无需额外编码的能力。

接口继承是指仅使用属性和方法的名称、但是子类必须提供实现的能力(子类重构爹类方法)。

　　在考虑使用继承时，有一点需要注意，那就是两个类之间的关系应该是“属于”关系。例如，Employee 是一个人，Manager 也是一个人，因此这两个类都可以继承 Person 类。但是 Leg 类却不能继承 Person 类，因为腿并不是一个人。

　　OO开发范式大致为：划分对象→抽象类→将类组织成为层次化结构(继承和合成) →用类与实例进行设计和实现几个阶段。

1. 类的继承
2. 继承的定义



1. 构造函数的继承

如果我们要给实例 c 传参，我们就要使用到构造函数，那么构造函数该如何继承，同时子类中又如何定义自己的属性？

继承类的构造方法：

        1.经典类的写法： 父类名称.\_\_init\_\_(self,参数1，参数2，...)

        2. 新式类的写法：super(子类，self).\_\_init\_\_(参数1，参数2，....)



如果我们只是简单的在子类Chinese中定义一个构造函数，其实就是在重构。这样子类就不能继承父类的属性了。所以我们在定义子类的构造函数时，要先继承再构造，这样我们也能获取父类的属性了。

      子类构造函数基础父类构造函数过程如下：

      实例化对象c ----> c 调用子类\_\_init\_\_()  ---- > 子类\_\_init\_\_()继承父类\_\_init\_\_()  ----- > 调用父类 \_\_init\_\_()

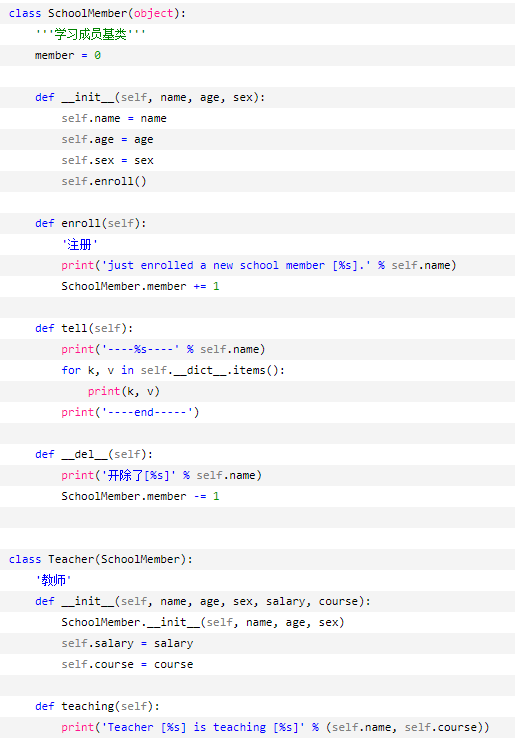


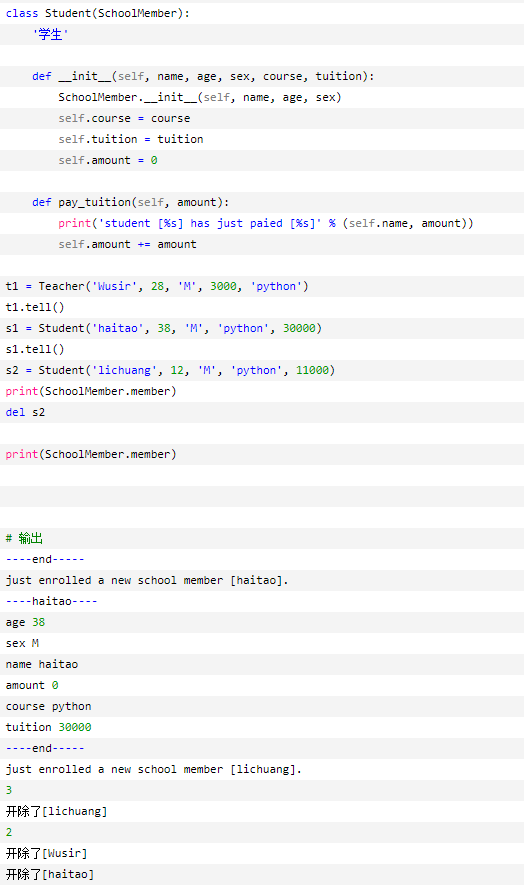
1. 子类对父类

如果我们对基类/父类的方法需要修改，可以在子类中重构该方法。如下的talk()方法



1. 类继承的事例





解释self的作用，self是将类中的数据和方法实例化，能够时内部变量或者方法互相调用<https://blog.csdn.net/xrinosvip/article/details/89647884>

## python中for循环

for I in range(len(nums))

假设len（nums）为3，则i的值为从0开始到2

## pandas的使用

pandas的数据类型常见的有：pandas.core.frame.DataFrame

pandas.core.series.Series

获取DataFrame类型数据的第几行，data.iloc[行数]，eg：data.iloc[0]

data.ix[行数]，eg：data.ix[0]

获取DataFrame类型数据的第几列，data[列名]，eg(知道列名的)：

data['w'] #选择表格中的'w'列，使用类字典属性,返回的是Series类型

data.w #选择表格中的'w'列，使用点属性,返回的是Series类型

data[['w']] #选择表格中的'w'列，返回的是DataFrame类型

不知道列名的时候用 data.ix[:,[1]] 去所有行的第二列

详情见连接：<https://blog.csdn.net/xiaodongxiexie/article/details/53108959>



可以看得出，nuinque()是查看该序列(axis=0/1对应着列或行)的不同值的数量。用这个函数可以查看数据有多少个不同值。

pandas聚合和分组运算之groupby

<https://blog.csdn.net/Leonis_v/article/details/51832916>

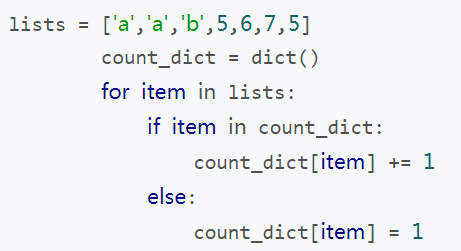
pandas - groupby, agg分组统计

<https://blog.csdn.net/ZK_J1994/article/details/77461480>

## 统计字符串里每个字符出现的次数的三种方法

1) 使用字典dict()

循环遍历出一个可迭代对象中的元素,如果字典没有该元素,那么就让该元素作为字典的键,并将该键赋值为1,如果存在就将该元素对应的值加1.



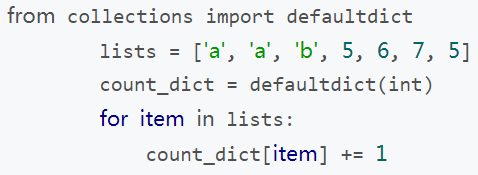
2)使用defaultdict（）

defaultdict(parameter)可以接受一个类型参数,如str,int等,但传递进来的类型参数，不是用来约束值的类型，更不是约束键的类型，而是当键不存在的话,实现一种值的初始化

defaultdict(int)：初始化为 0

defaultdict(float)：初始化为 0.0

defaultdict(str)：初始化为 ”



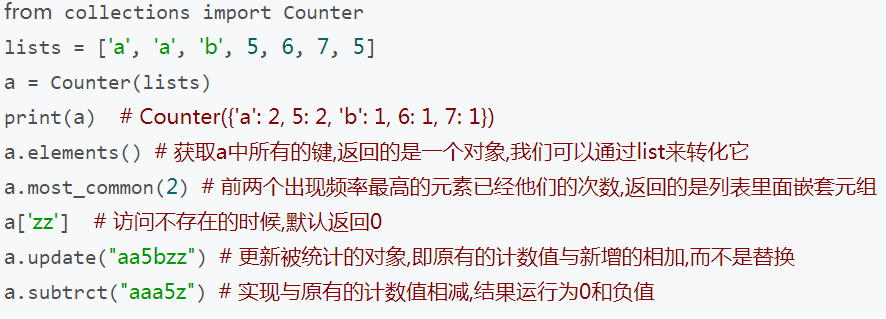
3）使用counter

Counter是一个容器对象，主要的作用是用来统计散列对象，可以使用三种方式来初始化

1、参数里面参数可迭代对象 Counter("success")

2、传入关键字参数Counter((s=3,c=2,e=1,u=1))

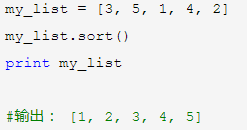
3、传入字典 Counter({"s":3,"c"=2,"e"=1,"u"=1})



## sort（）与sorted（）

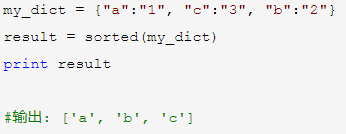
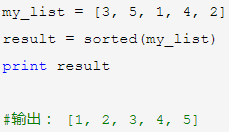
### sort（）

使用sort()方法对list排序会修改list本身,不会返回新list，通常此方法不如sorted()方便，但是如果你不需要保留原来的list，此方法将更有效sort()；sort()不能对dict字典进行排序



### sorted（）

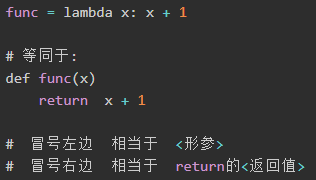
sorted（）排序，不会修改原list



对dict排序默认会按照dict的key值进行排序，最后返回的结果是一个对key值排序好的list

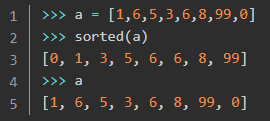
指定sorted的排序

**lambda 匿名函数**

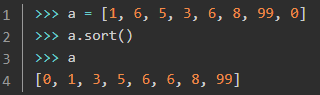


**注意, 既然我们称 lambda 为匿名函数, 所以官方并 不建议 用 func 去接收, 而是 直接使用**

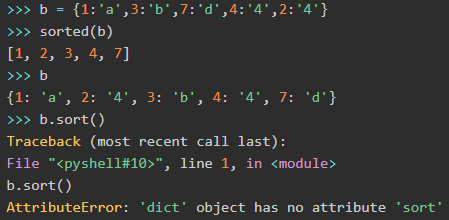
sorted()函数不会改变数组本身



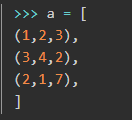
而 list.sort()会对list进行修改：



list.sort()函数只适用于数组，而sorted()还能用于其他数据结构：



下面用三个元组, 组成列表 通过 **使用key参数, 传入lambda匿名函数** 决定按哪个参数排序：



以 元组 的 第二个 元素 排序:



以 元组 的 第三个 元素 排序:



以 元组 的 第一个 元素 排序:



1. **sorted** 负责对 **列表a** 进行 迭代, 循环遍历每一个 对象, 这里就是 遍历这三个元组
2. **a** 就是那个列表

**3、x** 指代 元组里的 每一个对象, 即 每一个元组

4、**x[1]** 指代 元组的 第二个元素

字典排序



直接用key排序



下方代码流程:

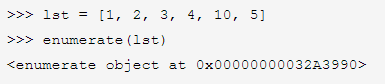
 sorted(b.items()... -----> 遍历 b.items()一个元组列表[(key，value)…]

 lambda a:a[1]-------------> 把每一个 遍历的对象 赋值给 a, 返回值 a[1]即每个字典对象的value值

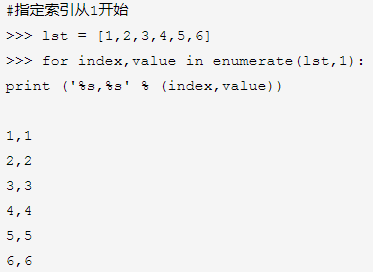
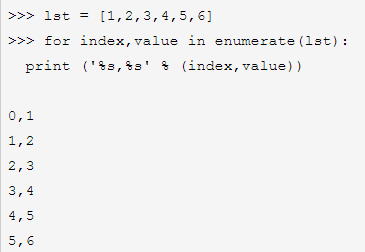
 key = ...------------------> 接收 lambda 的返回值 a[0]  


## enumerate（）函数

enumerate()是python的内置函数、适用于python2.x和python3.x  
enumerate在字典上是枚举、列举的意思  
enumerate参数为可遍历/可迭代的对象(如列表、字符串)  
enumerate多用于在for循环中得到计数，利用它可以同时获得索引和值，即需要index和value值的时候可以使用enumerate  
enumerate()返回的是一个enumerate对象



查看enumerate（）函数里的内容，得需要for



## lambda表达式

lambda表达式，通常是在**需要一个函数，但是又不想费神去命名一个函数**的场合下使用，也就是指**匿名函数**。

lambda和普通的函数相比，就是省去了函数名称而已，同时这样的匿名函数，又不能共享在别的地方调用。

其实说的没错，lambda在Python这种动态的语言中确实没有起到什么惊天动地的作用，因为有很多别的方法能够代替lambda。

1. 使用Python写一些执行脚本时，使用lambda可以省去定义函数的过程，让代码更加精简。

2. 对于一些抽象的，不会别的地方再复用的函数，有时候给函数起个名字也是个难题，使用lambda不需要考虑命名的问题。

3. 使用lambda在某些时候让代码更容易理解。

lambda基础

lambda语句中，冒号前是参数，可以有多个，用逗号隔开，冒号右边的返回值。lambda语句构建的其实是一个函数对象

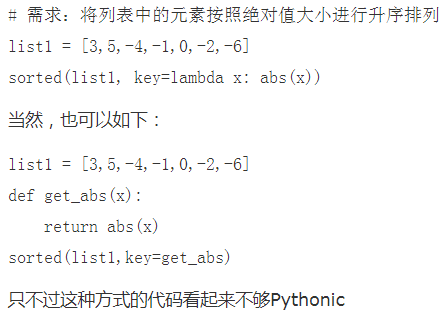
lambda所表示的匿名函数的内容应该是很简单的，如果复杂的话，干脆就重新定义一个函数了，使用lambda就有点过于执拗了。

lambda就是用来定义一个匿名函数的，如果还要给他绑定一个名字的话，就会显得有点画蛇添足，通常是直接使用lambda函数。如下所示：



### 应用在函数式编程中

Python提供了很多函数式编程的特性，如：map、reduce、filter、sorted等这些函数都支持函数作为参数，lambda函数就可以应用在函数式编程中。如下：



### 应用在闭包中

