python学习笔记

Python的发展历程

1、Python是一种简单易学，功能强大的编程语言。它有高效率的高层数据结构，能够简单、有效地实现面向对象编程

1. Python语法简单，支持动态输入，是解释型语言。

3、运用领域

（1）桌面GUI软件开发(wxPython, PyQT…)

（2）网络应用开发(内置模块, Twistd, Stackless…)

（3）2/3D图形处理,游戏开发 (PIL, pyGame…)

（4）文档处理,科学计算(moinmoin, numpy…)

（5）Web应用开发(Flask, Django,Z,OPEQuixote…)

（6）移动设备应用开发(PyS60…)

（7）数据库开发(支持SQL和NOSQL, ZODB…)

（8）嵌入其它应用(嵌入C/C++, delphi, 调用DLL…)

二、Python2，Python3不同

1.版本差异

2.不等于

3.除法不同

4类型差异

Py2long类型。Py3去除了，py3新增了bytes类型

一.基本注释

#标注注释

“””标注多行注释

二.变量和数据类型

(1).Python中的变量用来存储数据。

(2).变量的类型

1.数字类型2.布尔类型3.字符串类型4.列表类型5.元组类型6.字典类型

(3).关键字

help() #进入帮助系统

help>keywords #查看所有的关键字列表

help>return #查看return这个关键字的说明

help>quit #退出帮助系统

（4）简单数值类型

1.整型

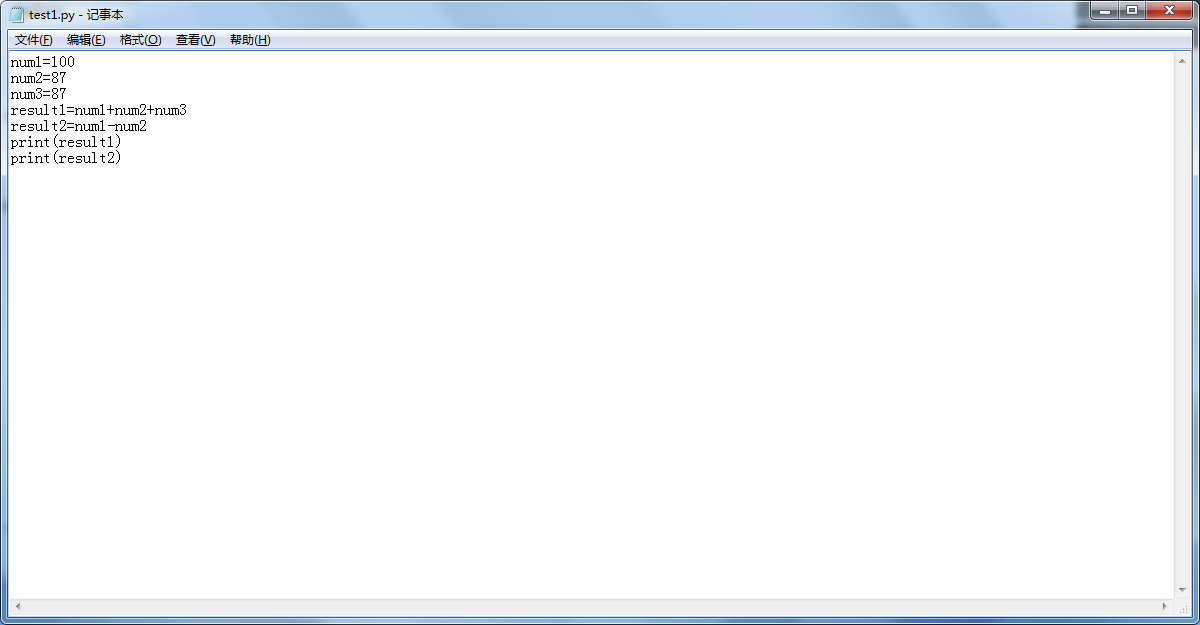
2.浮点型

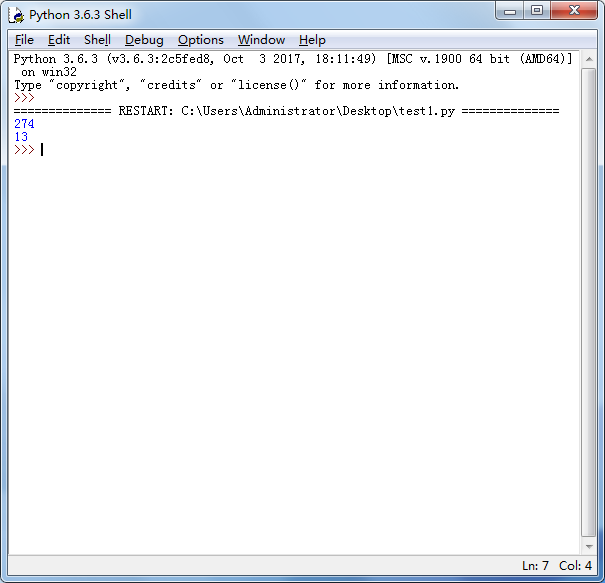
浮点型表示实数

浮点型字面值可以用十进制或科学计数法表示

浮点型遵循的是IEEE754双精度表示

布尔型其实是整型的





（5）Python标识符

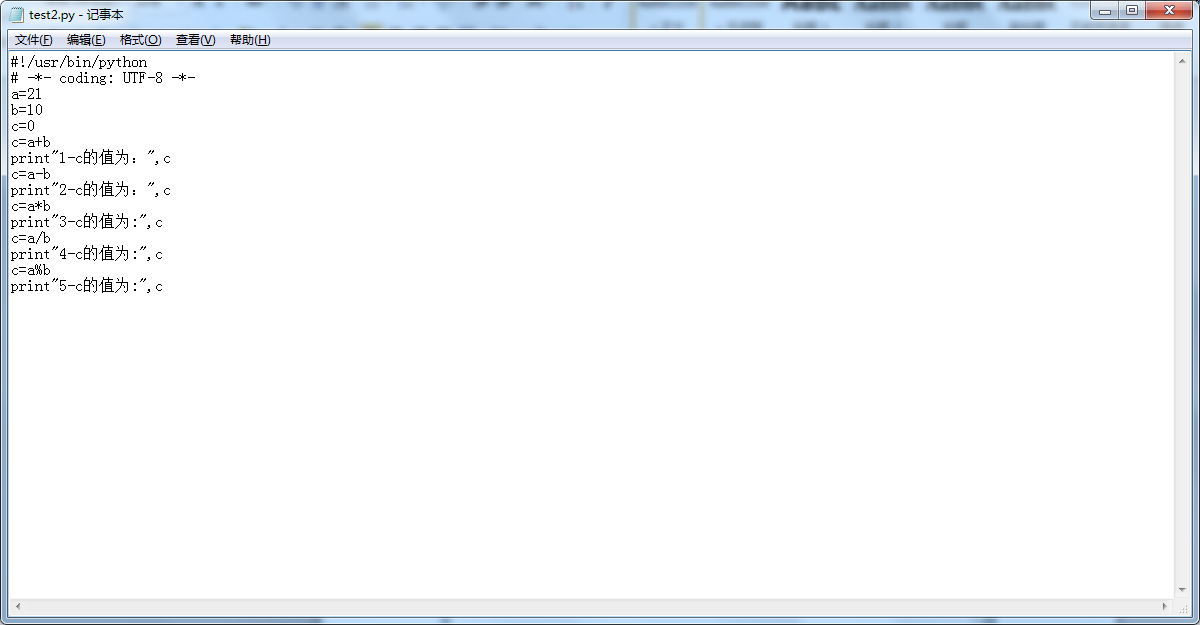
1.标识符的命名规则

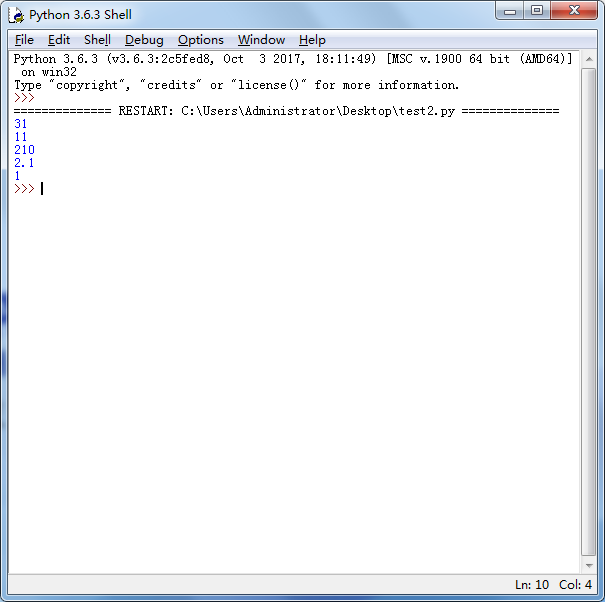
标识符由字母，数字和下划线组成，且数字不能开头。

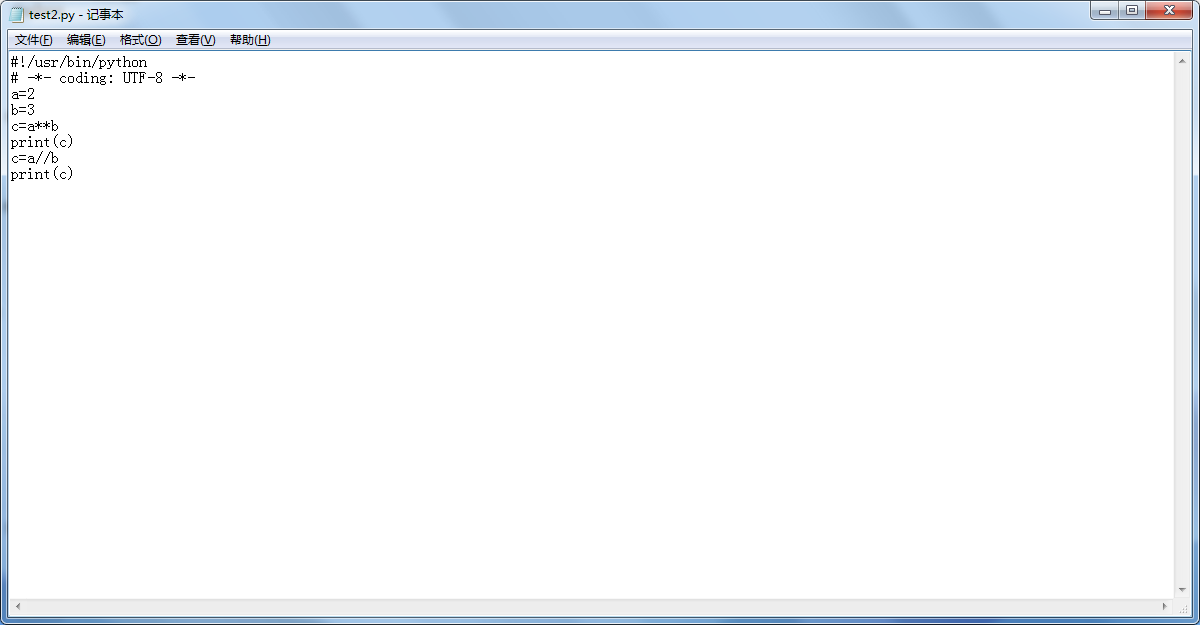
Python中的标识符是区分大小写的。

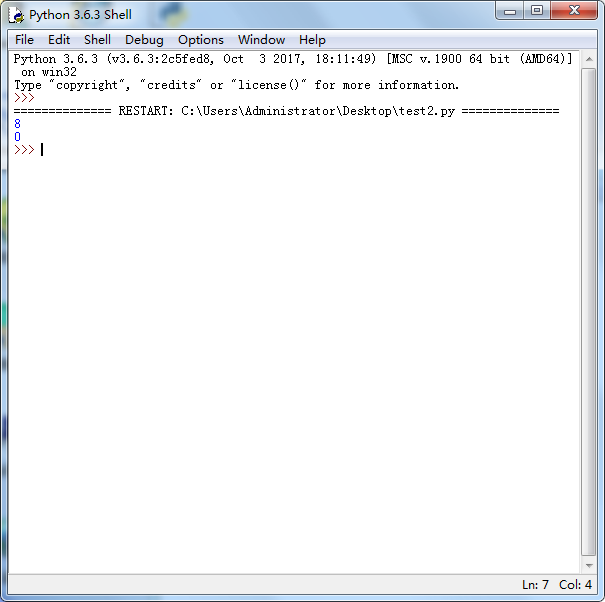
Python中的标识符不可以使用关键字。

2关键字指的是具有特殊功能的标识符









一、基本语法

1注释

#标注单行注释

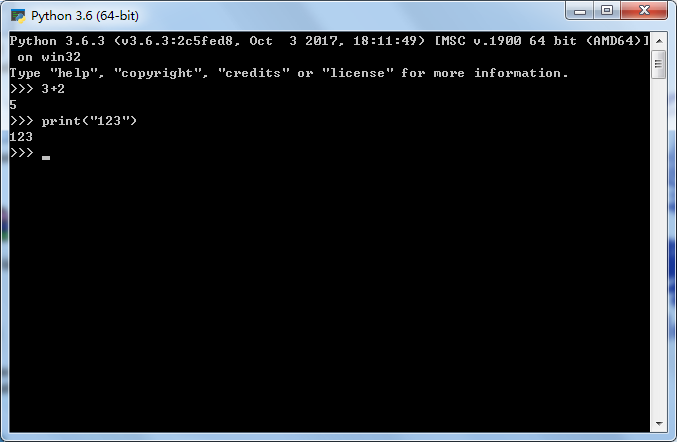
“””标注多行注释

Python用缩进来表示代码块

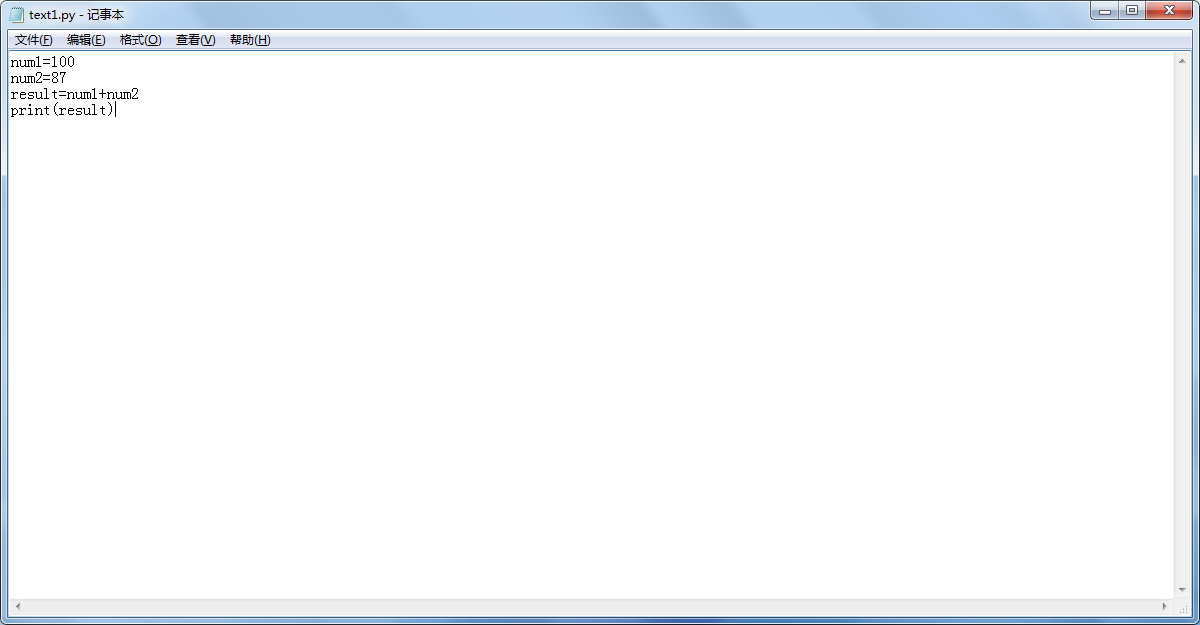
Python用圆括号进行换行

Python中的变量用来存储数据

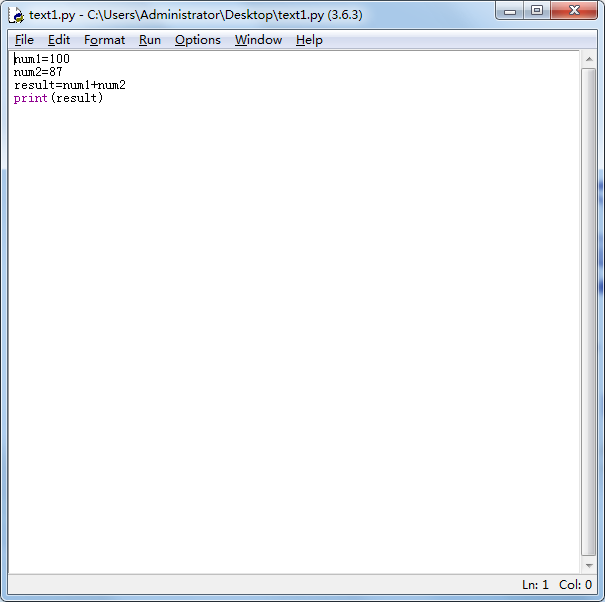
输入：



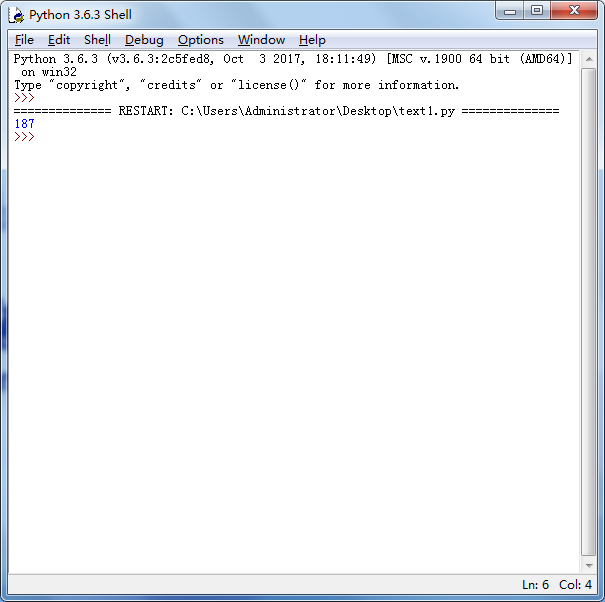
输入：



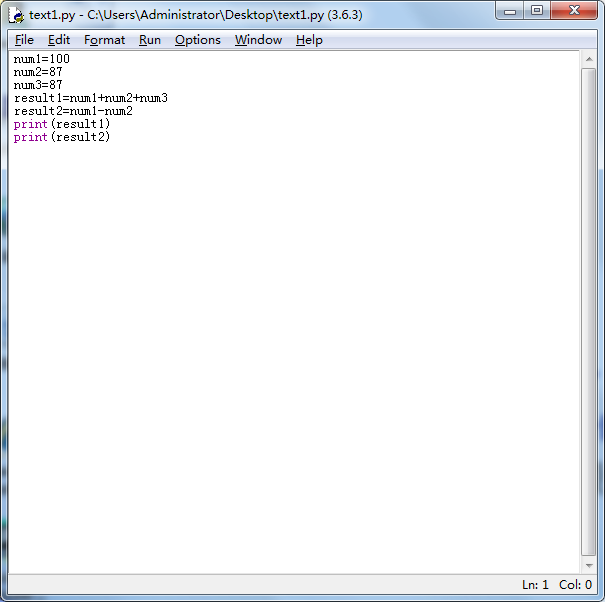
输入：



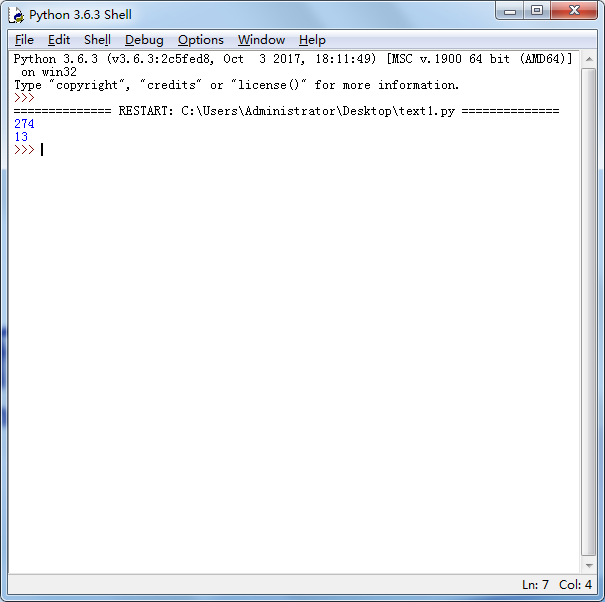
输出：



输入：



输出：



二、变量和数据类型

Python变量类型

1数字类型 2布尔类型（1/0） 3字符串类型 4列表类型 5元组类型 6字典类型

三、标识符和关键字

关键字是具有特殊功能的标识符

Python标识符

命名规则：

1标示符由字母、下划线和数字组成，且数字不能开头

2Python中的标示符是区分大小写的

3Python中的标识符不能使用关键字

>>>help()#帮助进入系统

help>keywords

help>return

help>quit

python整型数值范围2的-231~231-1

浮点型用于表示实数，用十进制或科学计数法表示

布尔类型其实是整型的子类型，只有两个取值True和False

复数类型用于表示数学中的复数

数字类型转换

Int（x[,base]） 将x转换为一个整数

Float(x) 将x转换为一个浮点数

Complex(real [,imag]) 创建一个复数

四、运算符

算术运算符

\*\*表示求幂

//表示取整除

赋值运算符，即=

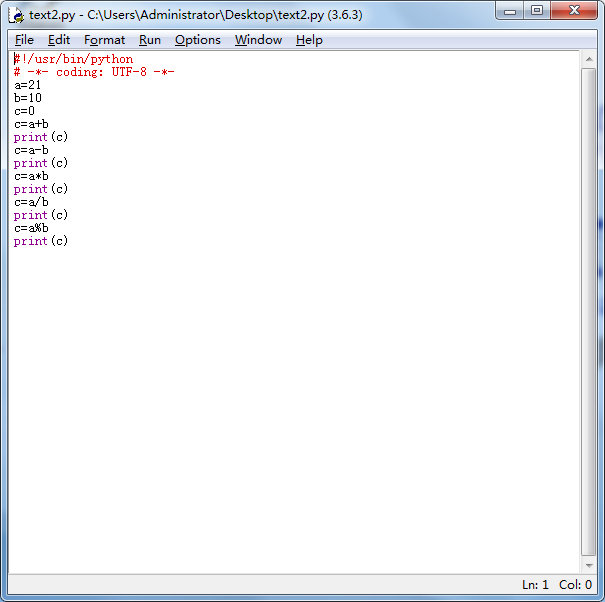
复合赋值运算符

比较运算符

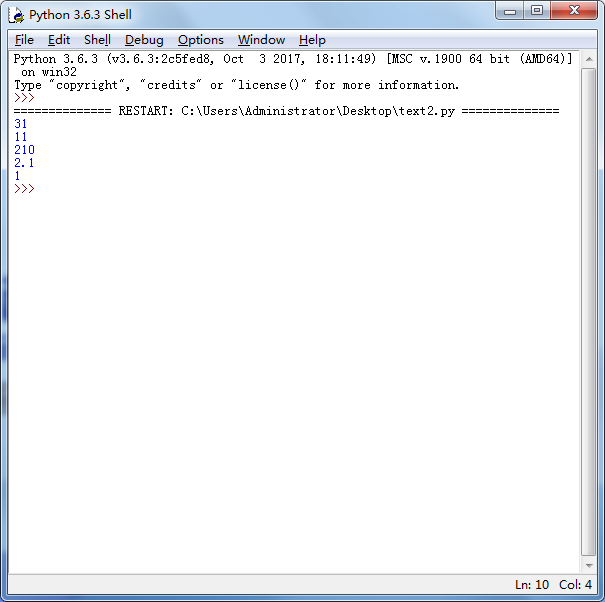
逻辑运算符and,or,not

成员运算符in，not in

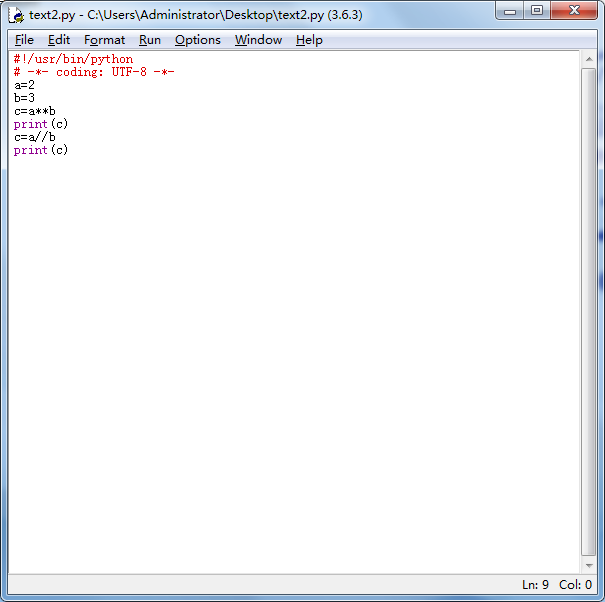
输入：



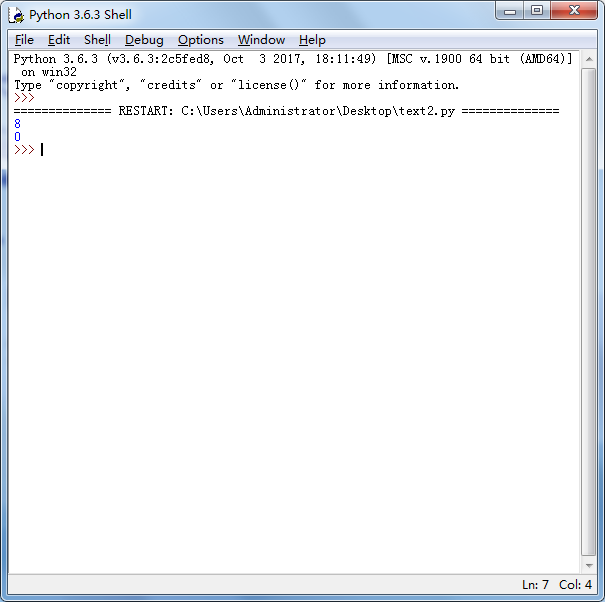
输出：



输入：



输出：



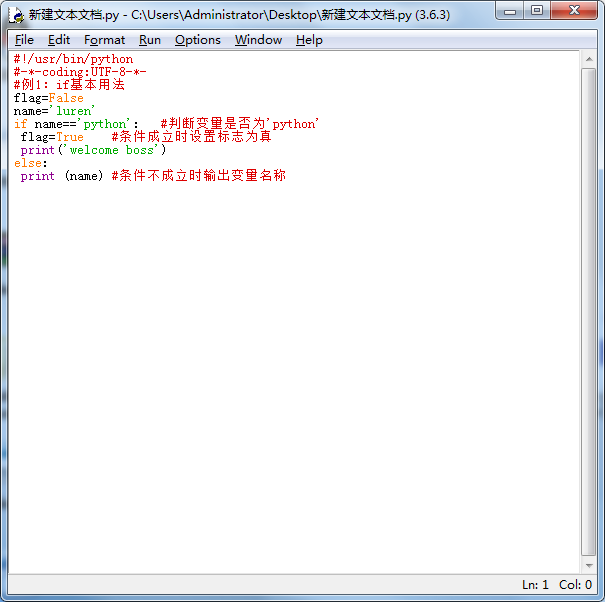
五、Python判断语句

if是最简单的条件判断语句，它可以控制程序的执行流程

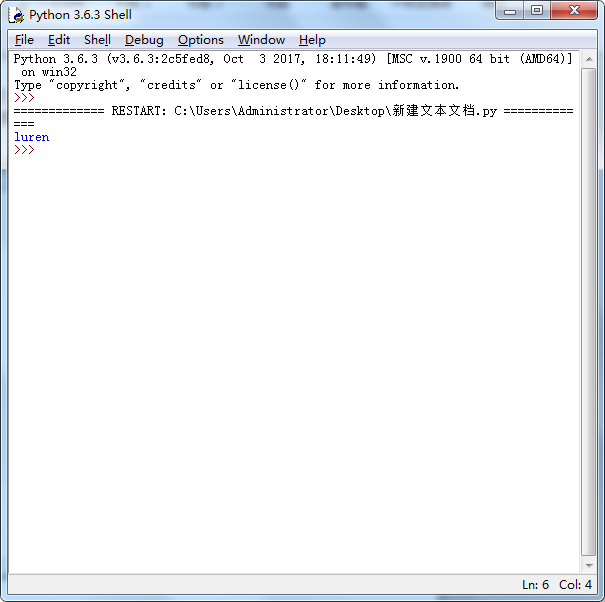
if-elif判断语句，该语句可以判断多种情况

if嵌套

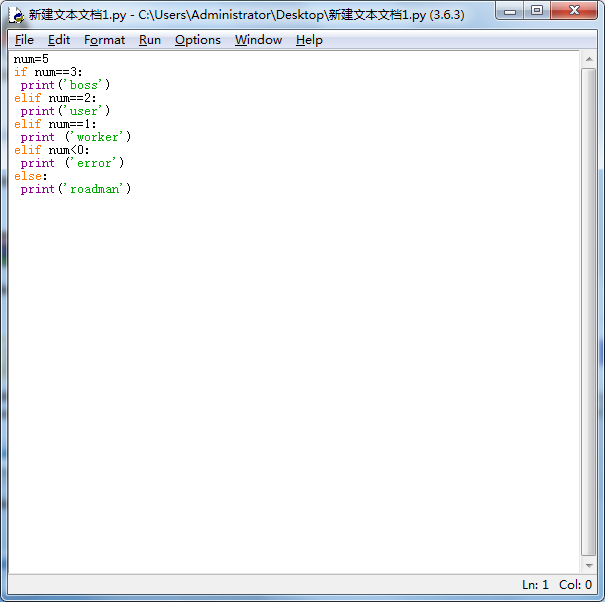
输入：



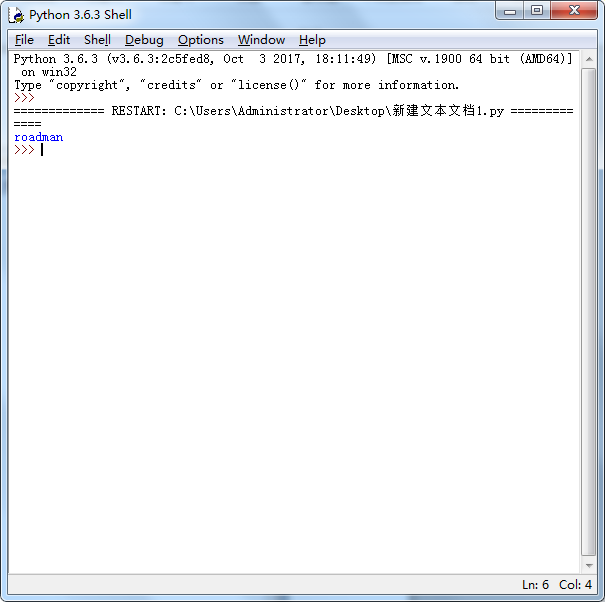
输出：



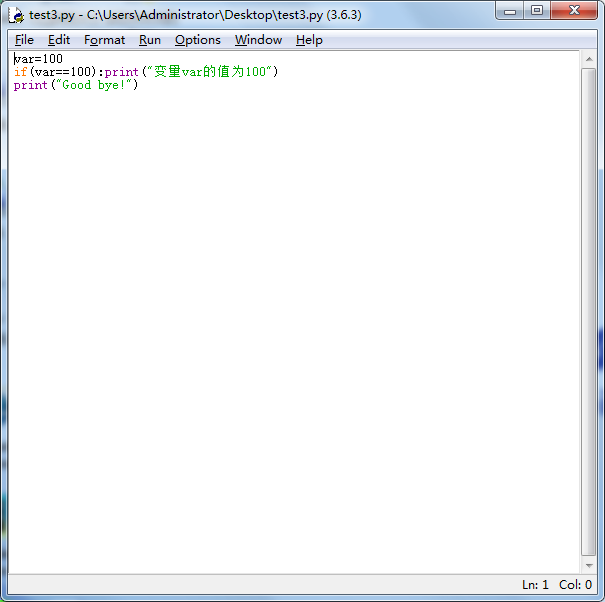
输入：



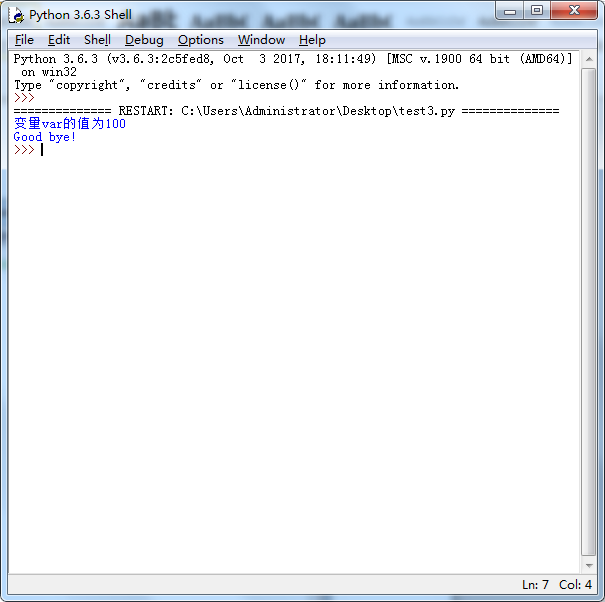
输出：



输入;



输出：



六、循环语句

while循环语句

while条件表达式：条件满足，执行循环语句

for循环可以遍历任何序列的项目

for i in [0,1,2]:

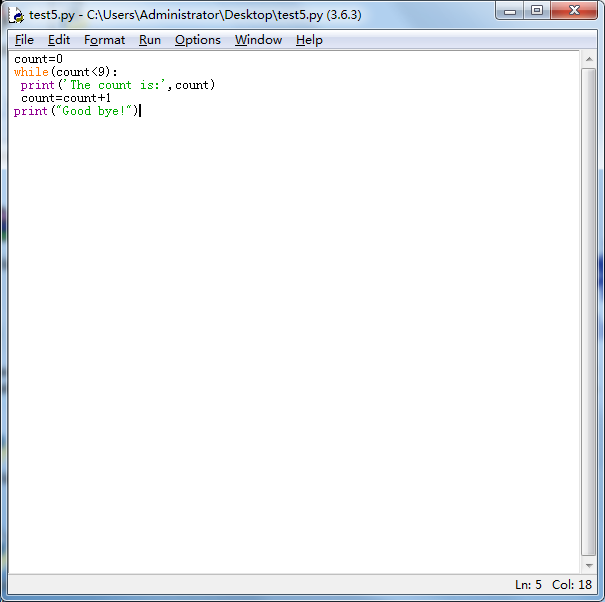
print(i)

range函数，它可以生成一个数字序列

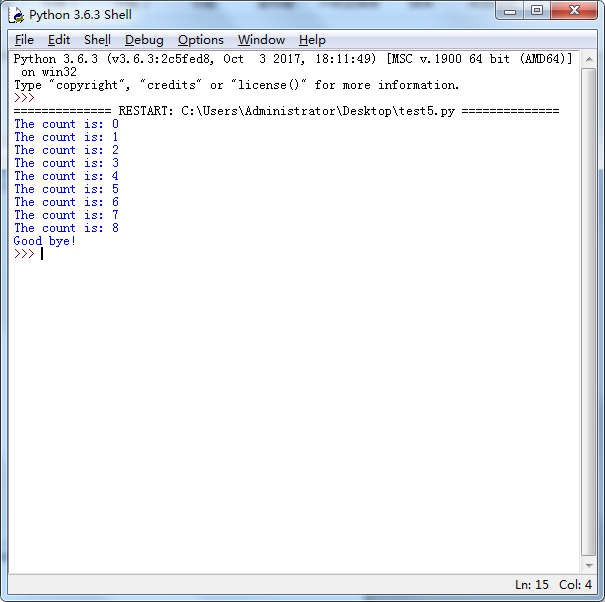
for i in range(start,end):

while嵌套指的是while循环里面还包含while循环

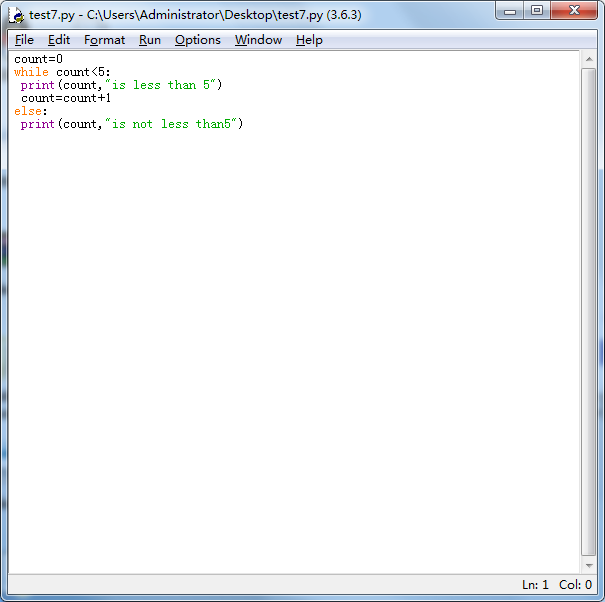
输入：



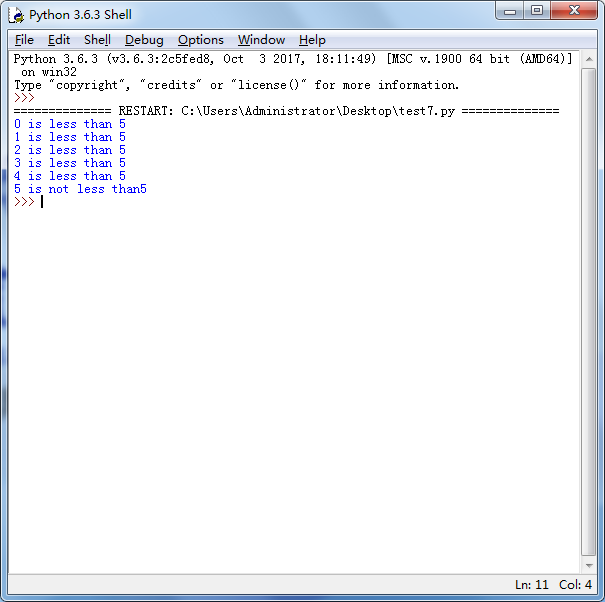
输出：



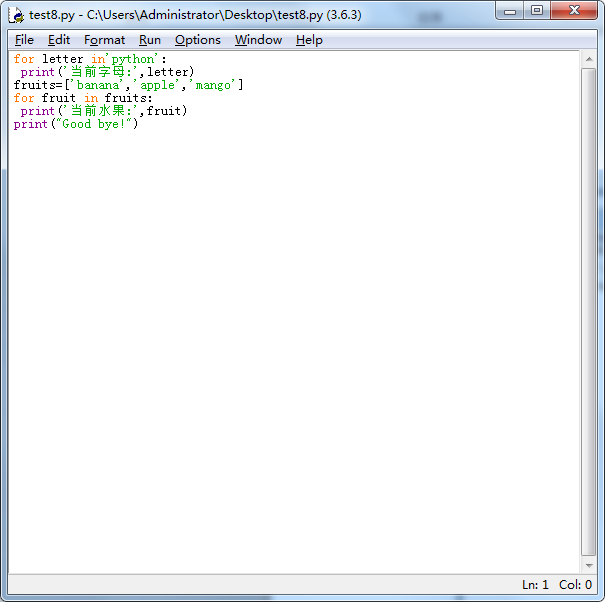
输入：



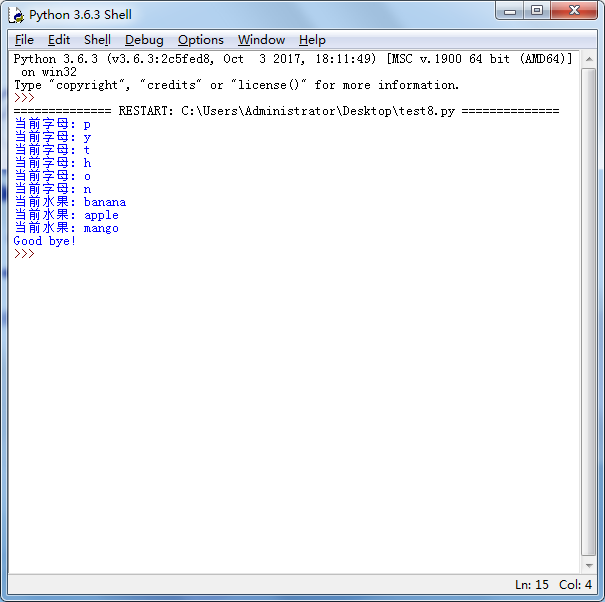
输出：



输入：

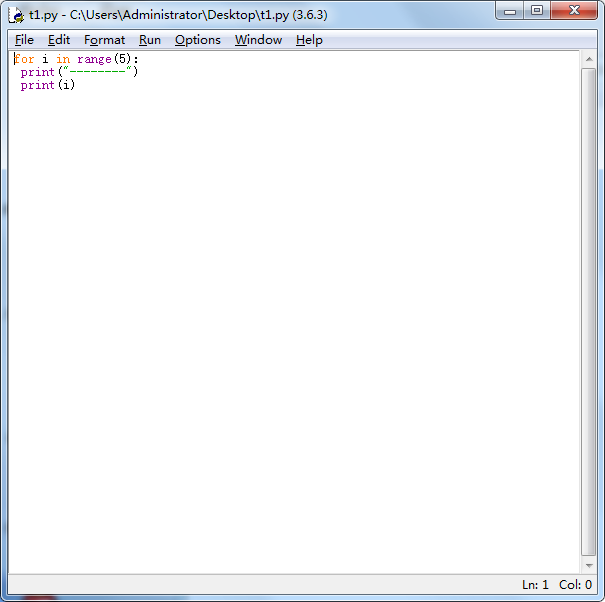


输出：

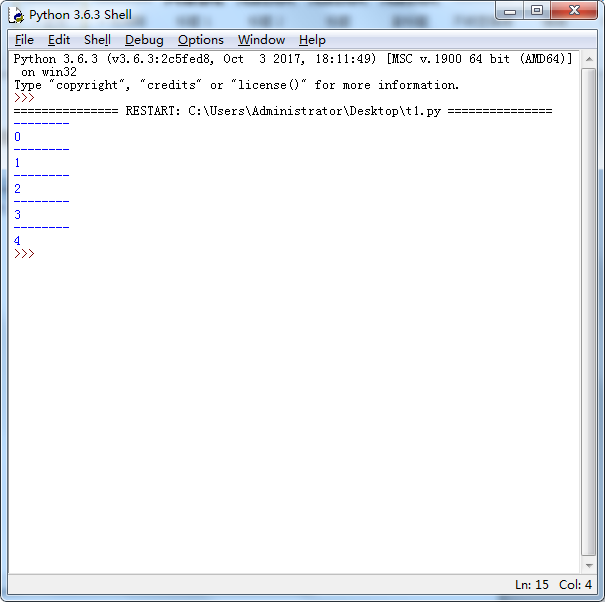


break语句用于结束整个循环

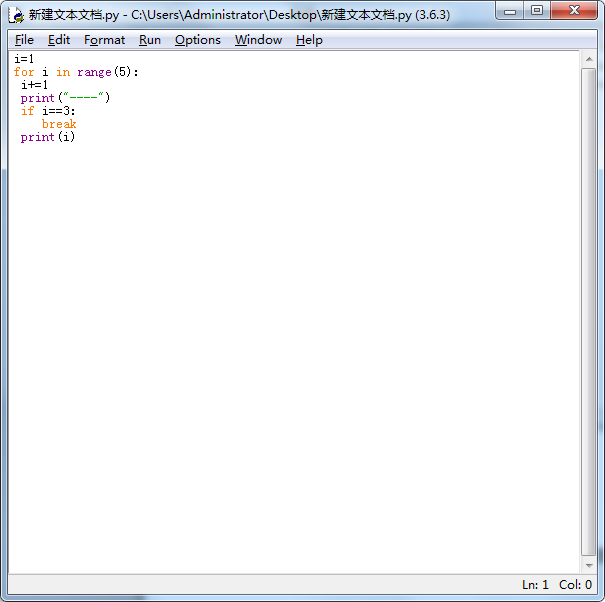
输入：



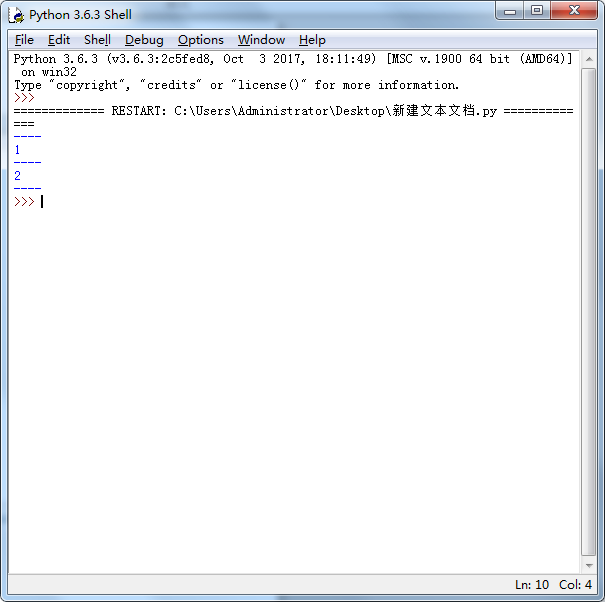
输出：



输入：



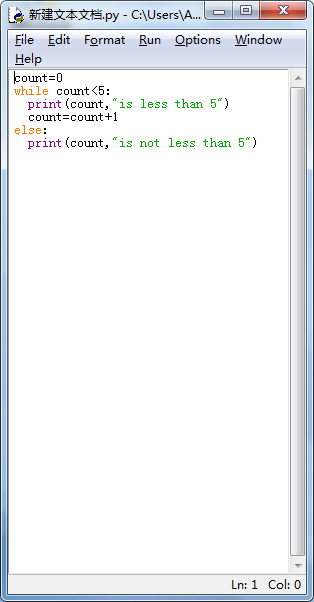
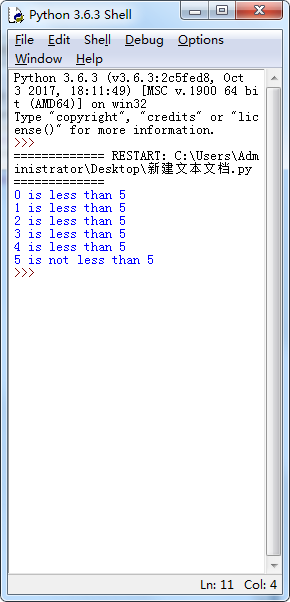
输出：



Continue结束本次循环，接着下面的循环

Pass是空语句，为了保持程序结构完整性

else语句可以和循环语句结合使用，并且else语句旨在循环完成后执行

第四章

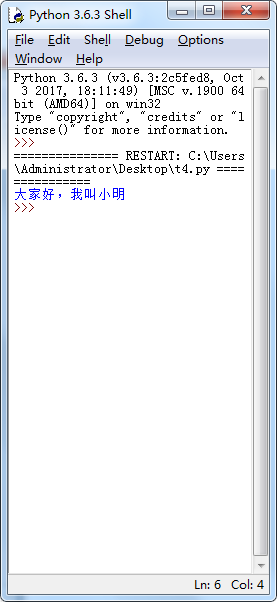
一、字符串

字符串是一种表示文本数据的类型

转义字符let\’s \”双引号 \n换行 \b退格

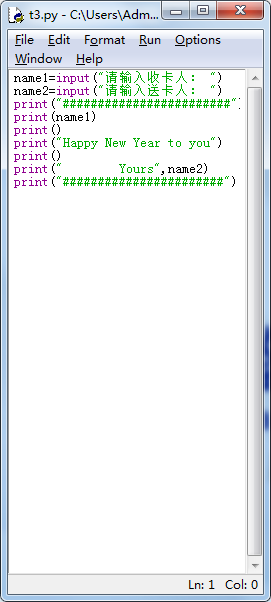
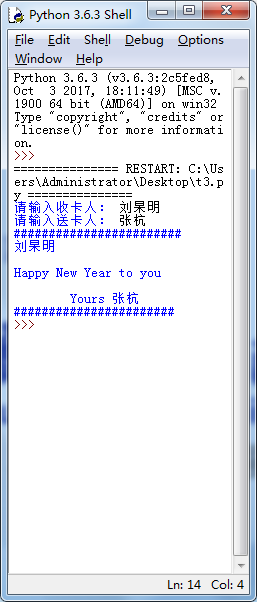
字符串输出

字符串格式化再输出

字符串输入

输入： 输出：

字符串存储方式

字符串中的每个字符都对应一个下标，下标编号是从0开始的

使用切片截取字符串

字符串内建函数

字符串转换

find函数：检测字符串是否包含子字符串

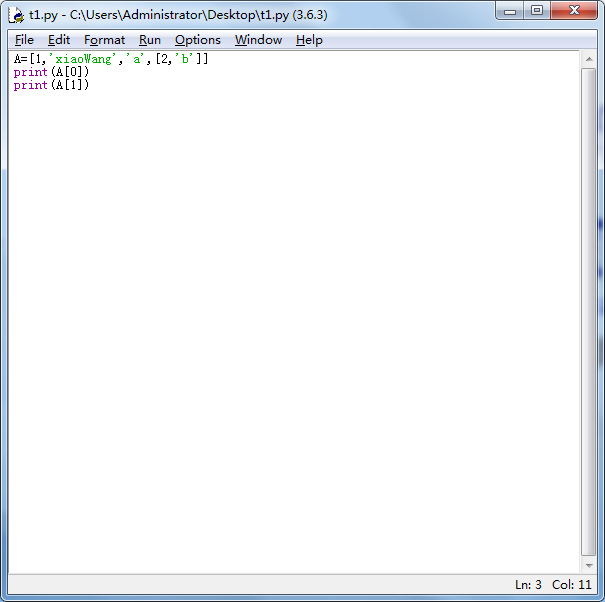
len、find、str、

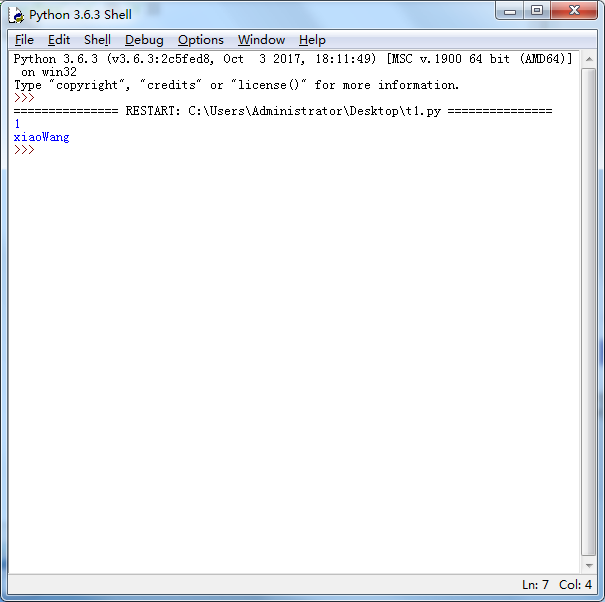
第五章 列表、元组和字典

1、列表的概述：

列表是Python中的一种数据结构，他可以存储不同类型的数据。

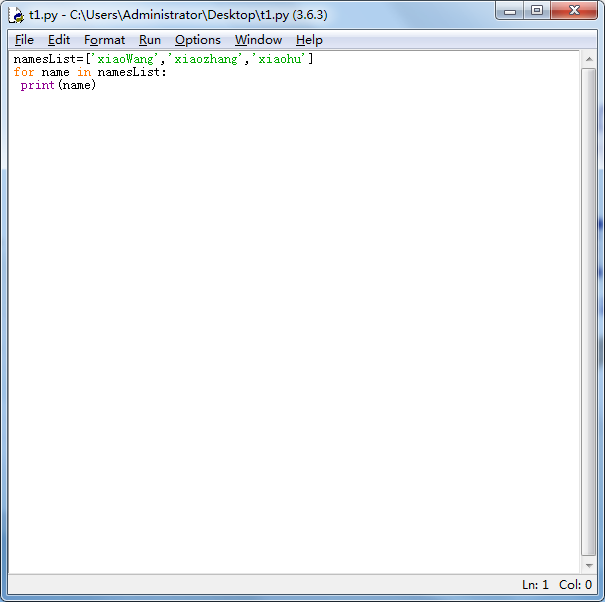
列表索引是从0开始的，我们可以通过下标索引的方式来访问列表中的值。

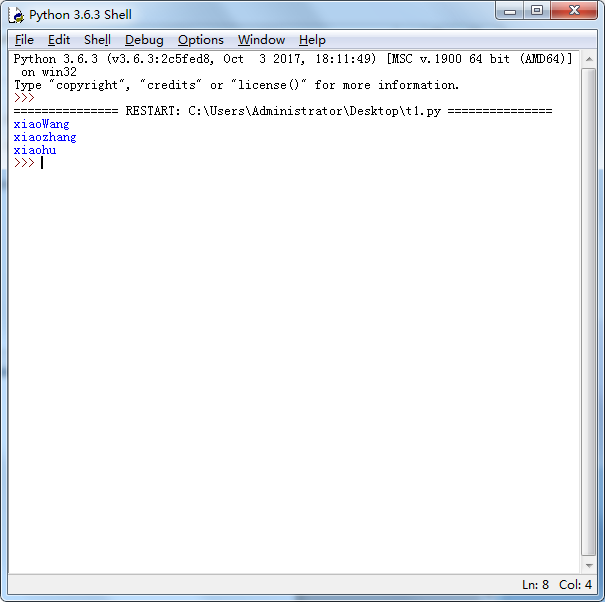




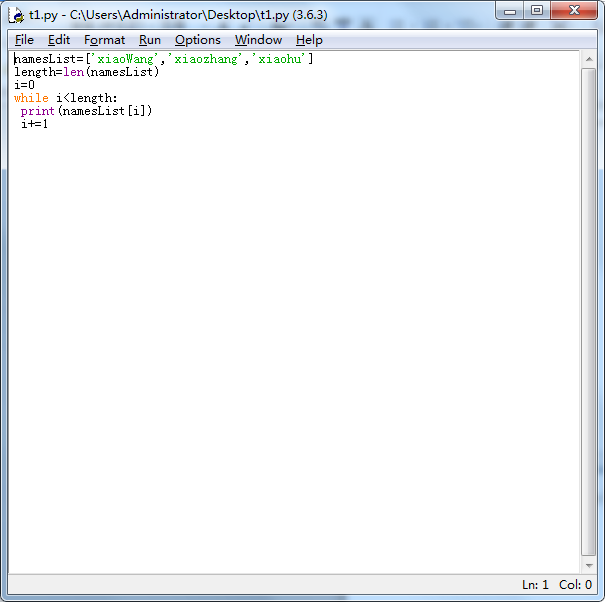
列表的循环遍历

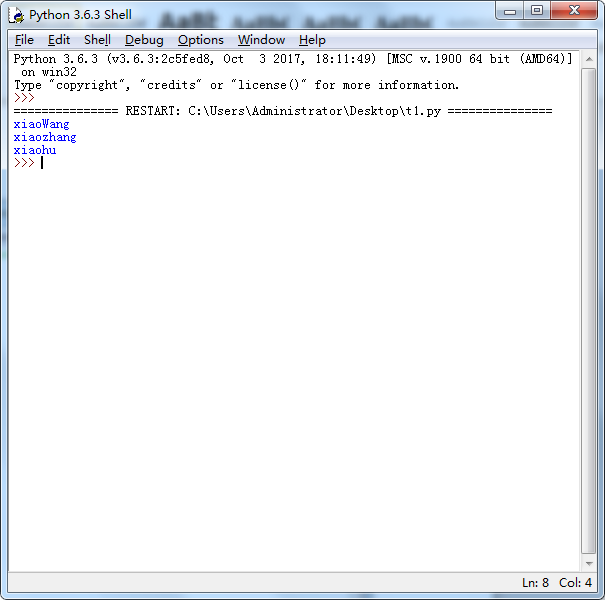
1. 使用for循环遍历列表



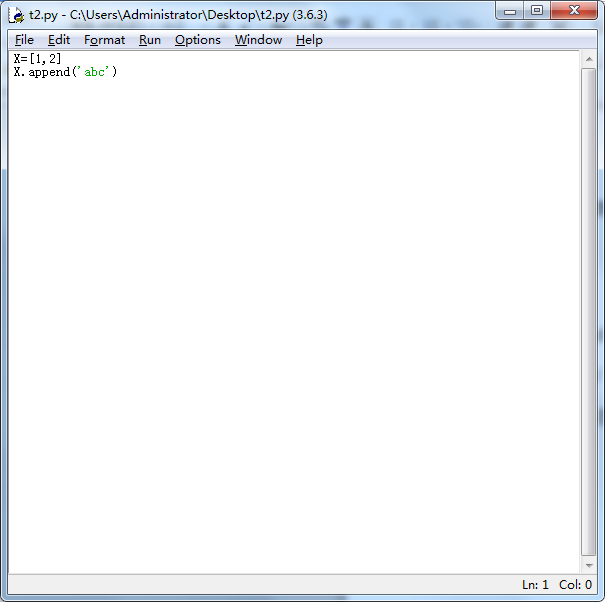


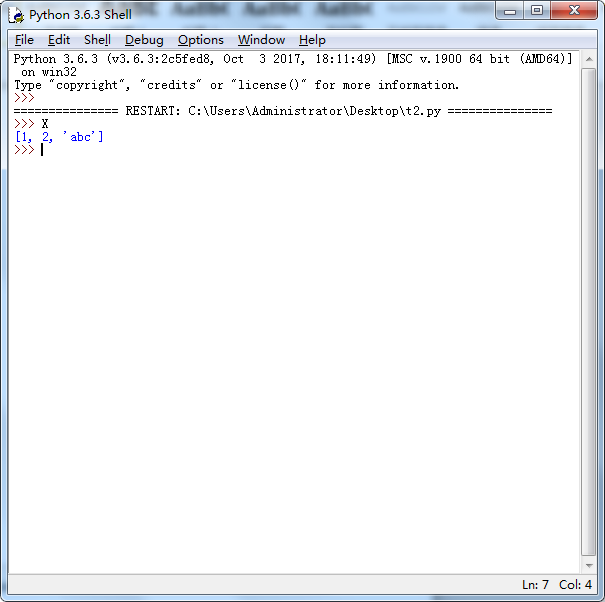
2、使用while循环遍历列表



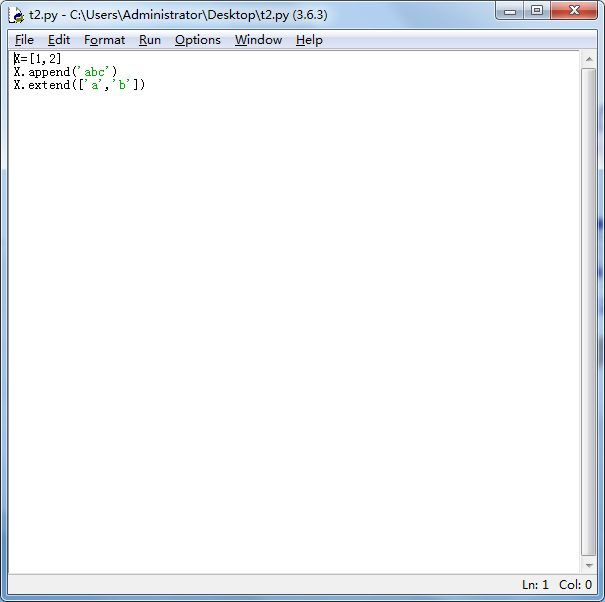


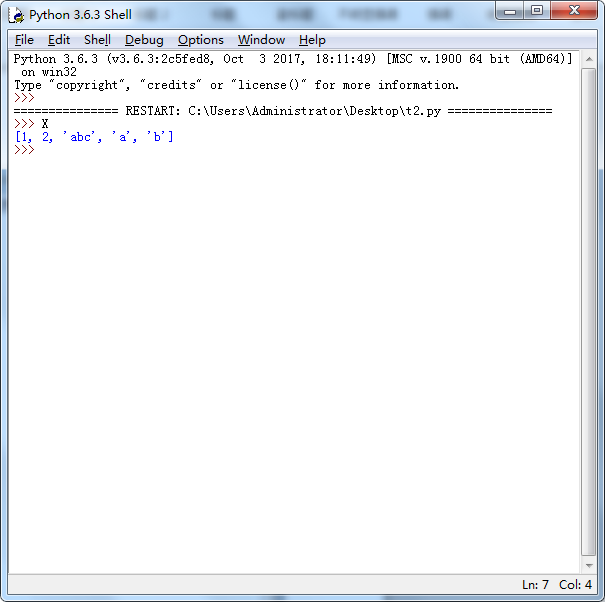
3、通过append可以向列表添加元素



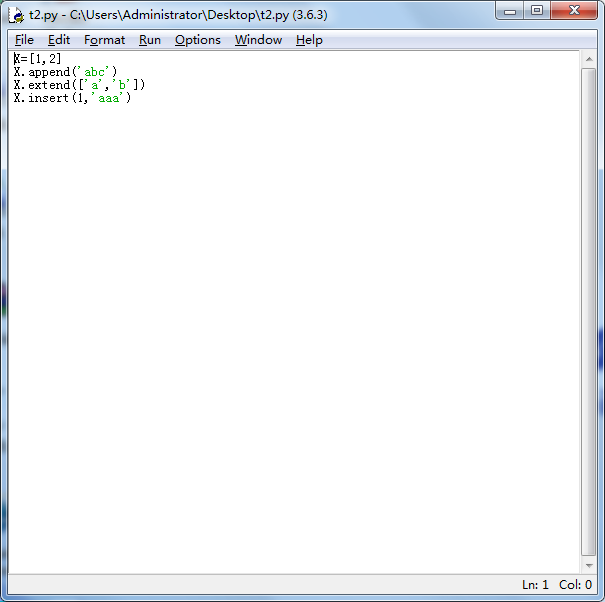


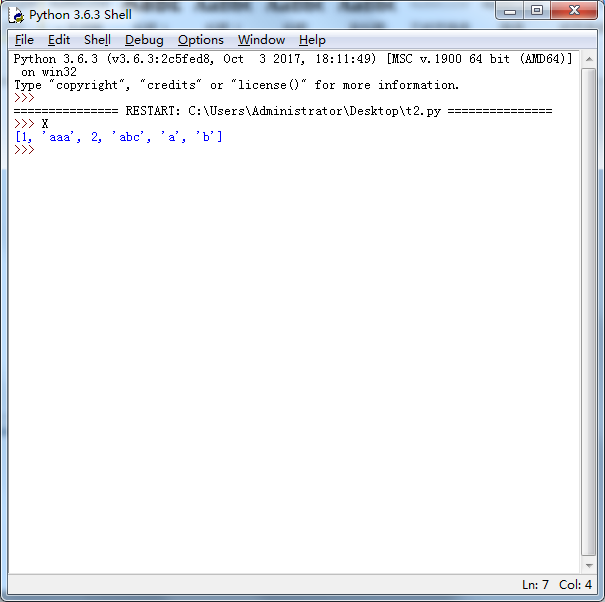
通过extend可以将另一个列表的元素添加到列表中。



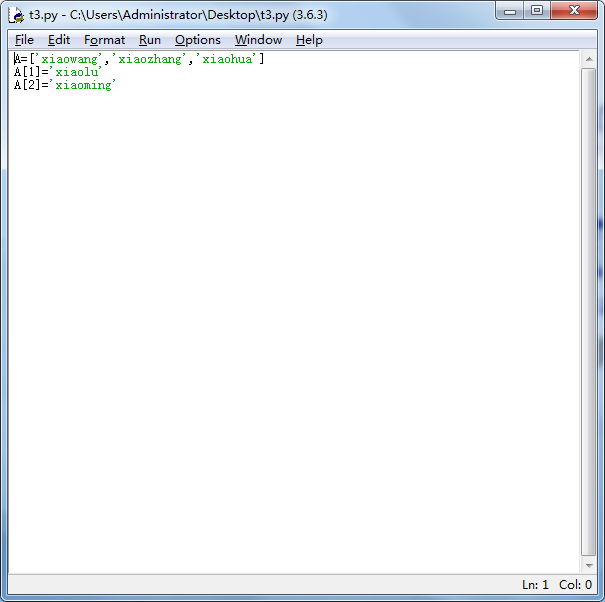


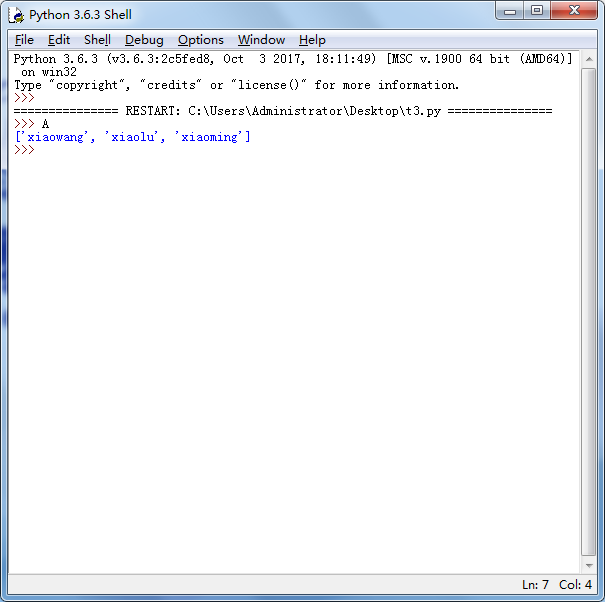
通过insert在指定位置index前添加object。



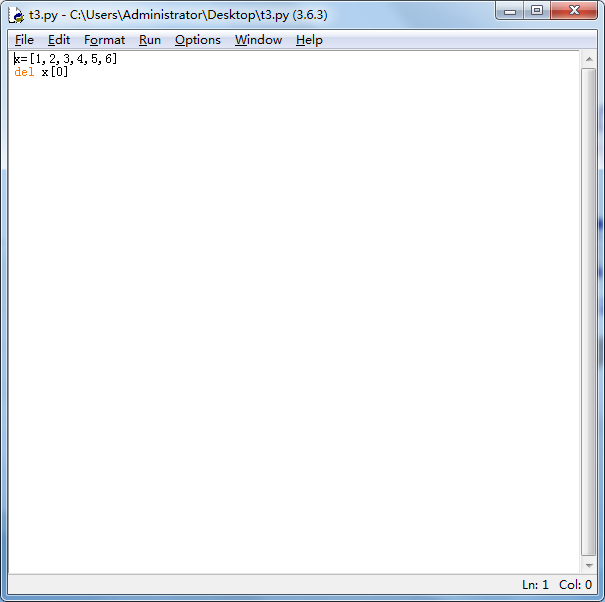


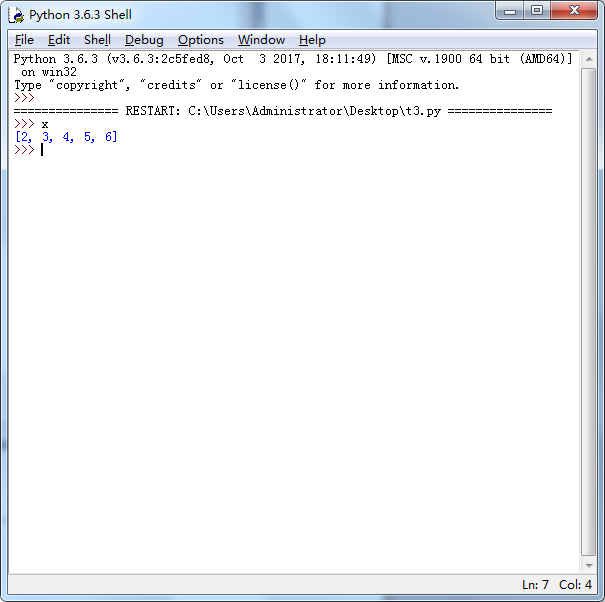
列表元素的修改，也是通过下标来实现的。



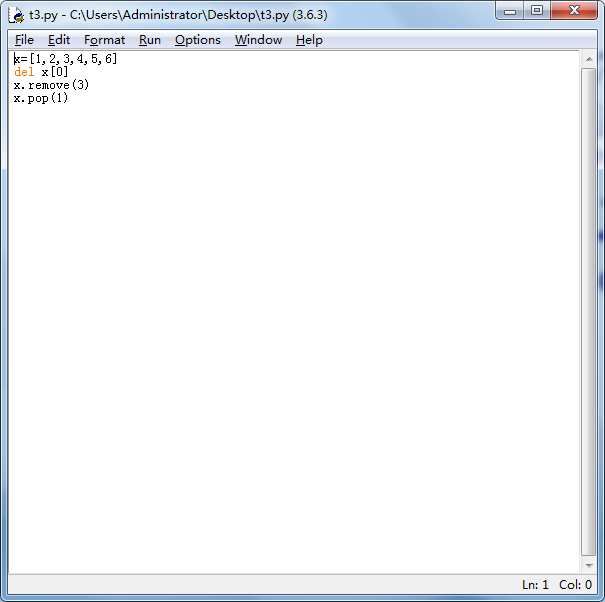


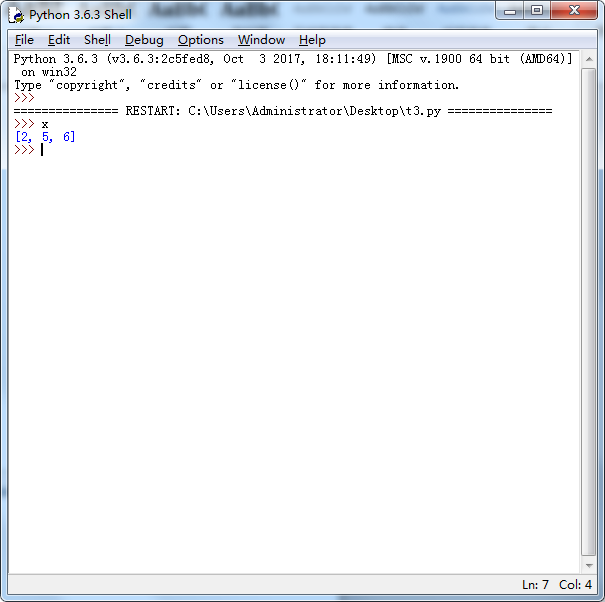
del:根据下标进行删除

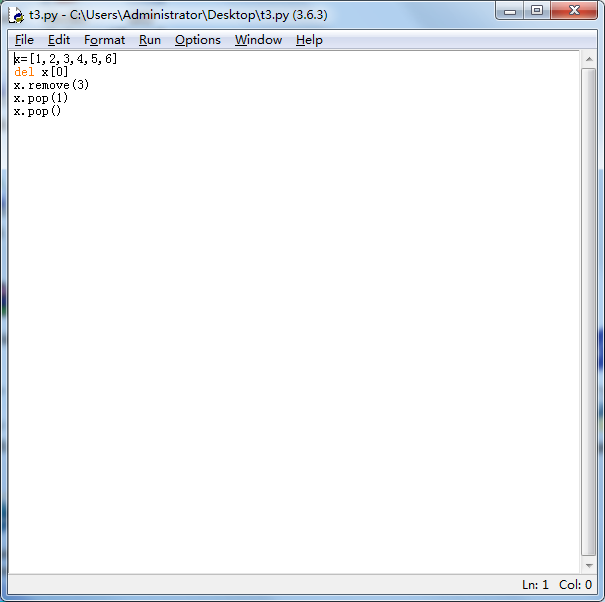


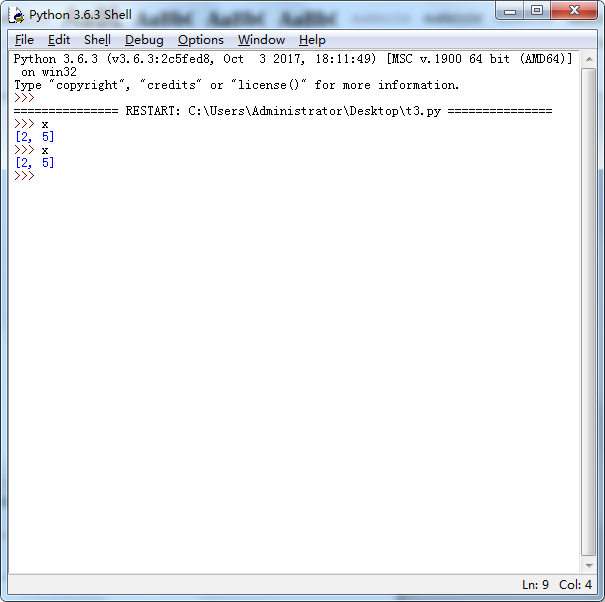


pop:删除最后一个元素

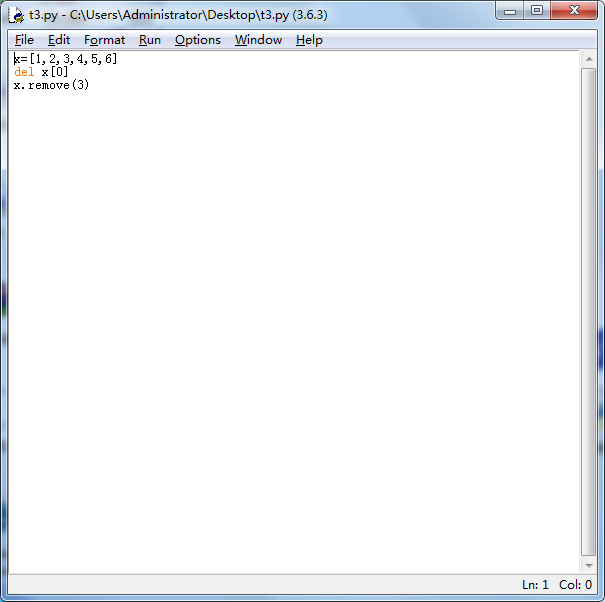


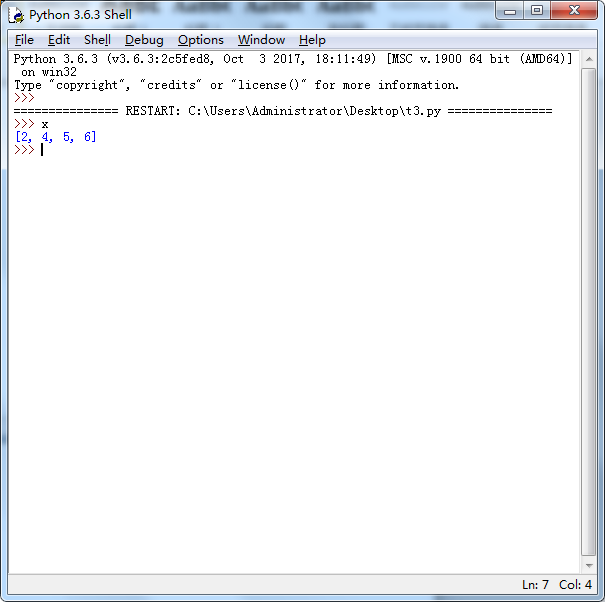






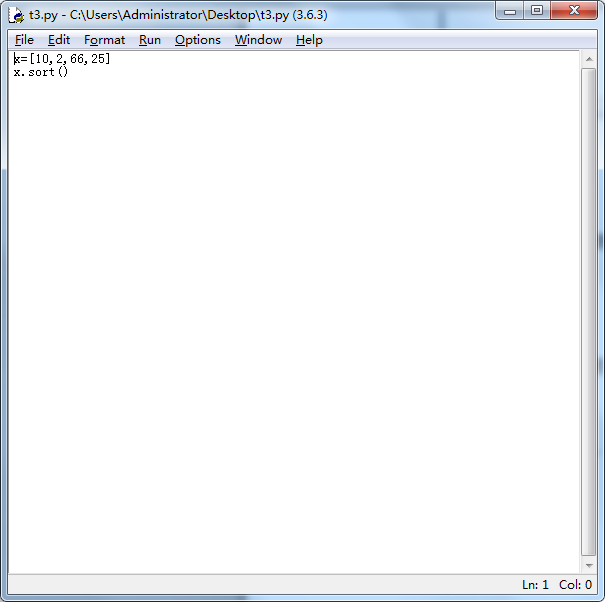
remove：根据元素的值进行删除

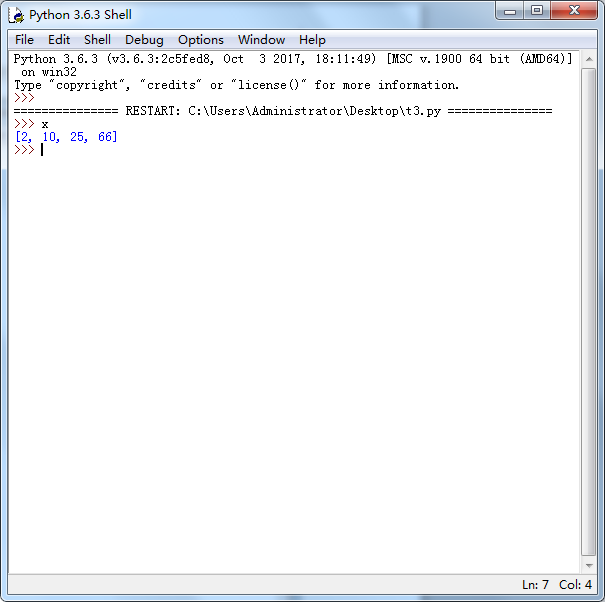


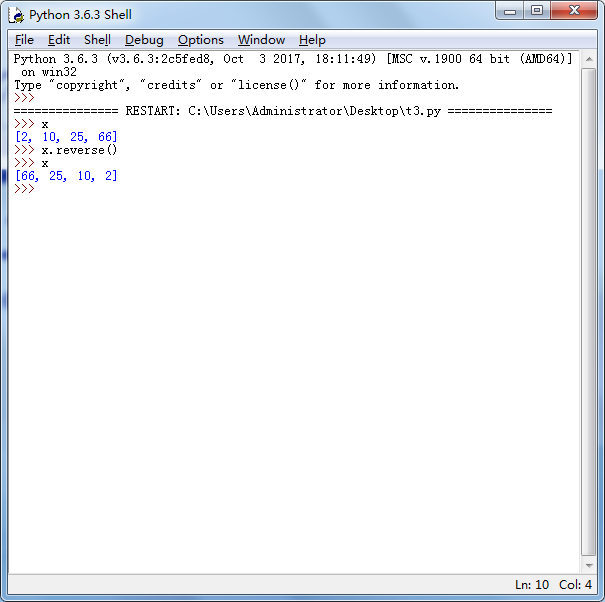


sort方法：列表的元素按照特定顺序排列

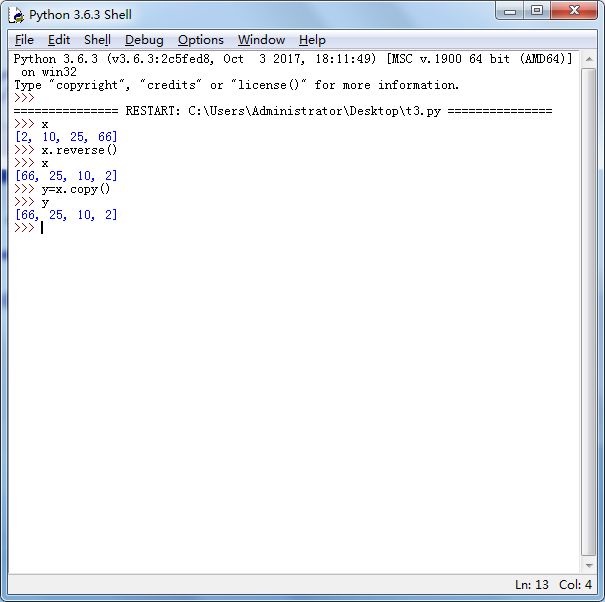
reverse方法：将列表逆置



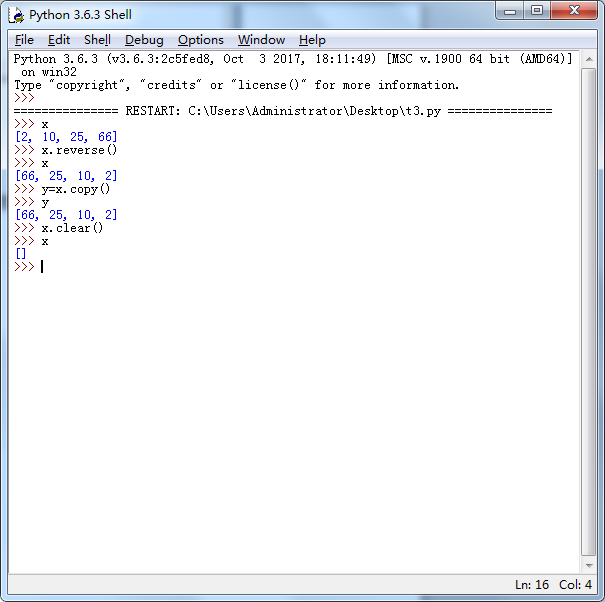




列表的复制：

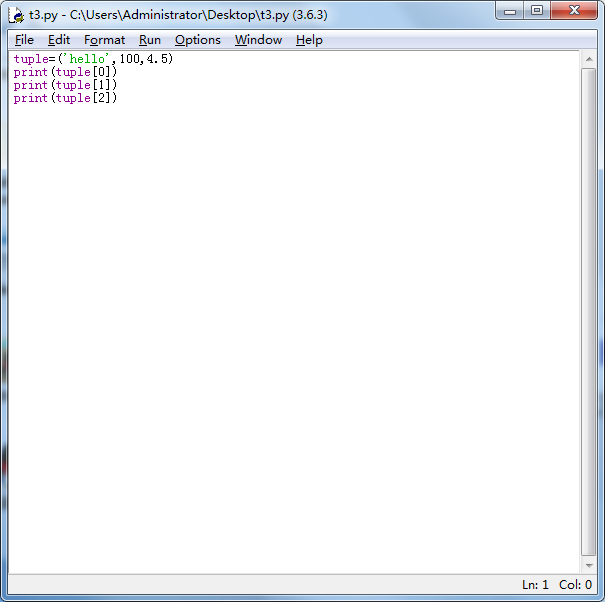


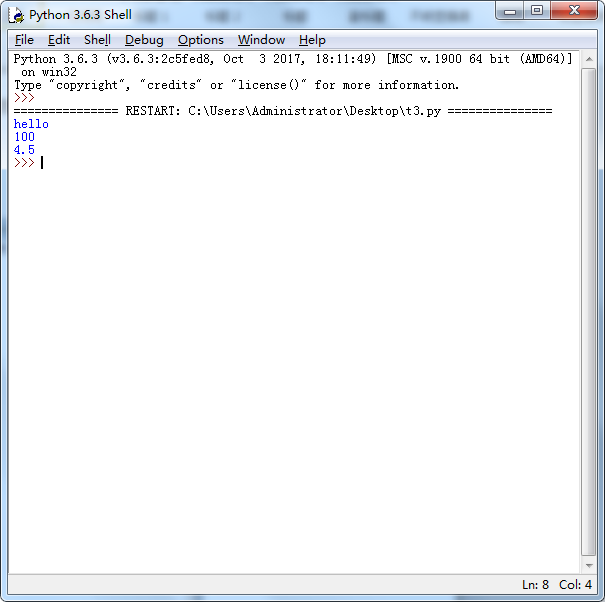
列表的清除：



列表的嵌套指的是一个列表的元素又是另一个列表。

Python的元组与列表相似，不同之处在于元组的元素不能修改。元组使用小括号，列表使用方括号。





元组的内置函数

len（tuple） 计算元组元素个数

max(tuple) 返回元组中元素最大值

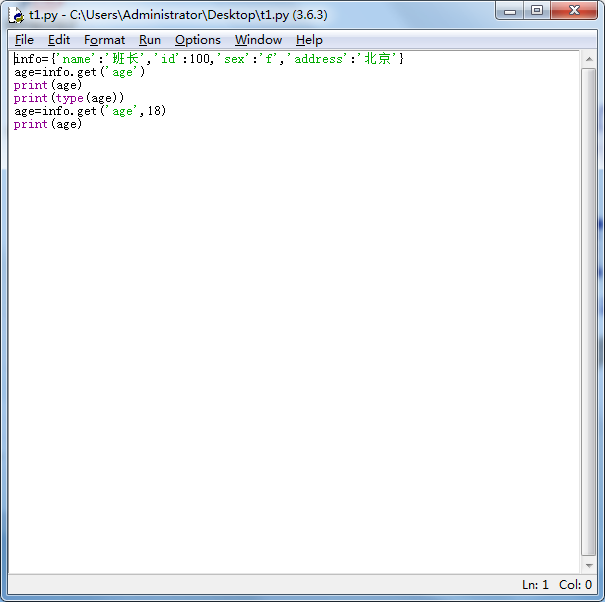
min(tuple) 返回元组中元素最小值

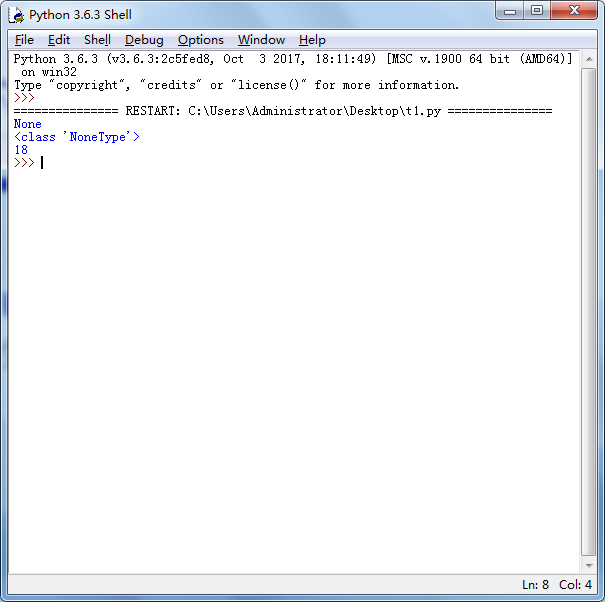
tuple(seq) 将列表转为元组

字典是一种存储数据的容器,它和列表一样，都可以存储多个数据。

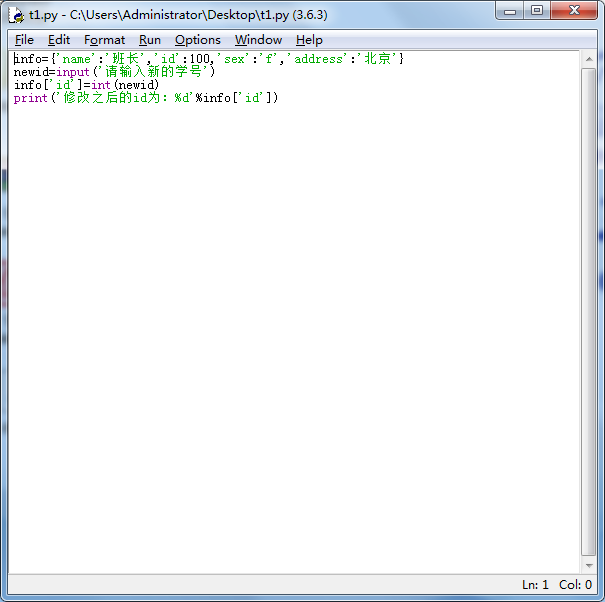
根据键访问值。

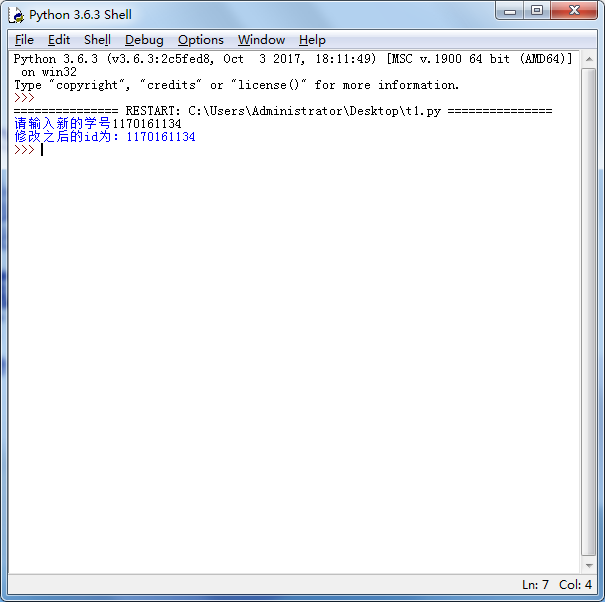
如果我们不确定字典中是否存在某个键而又想获得其值时，可以使用get方法，还可以设置默认值。



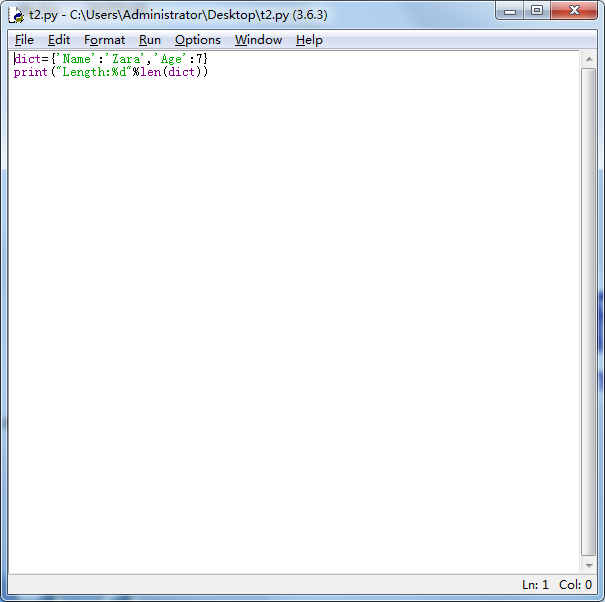


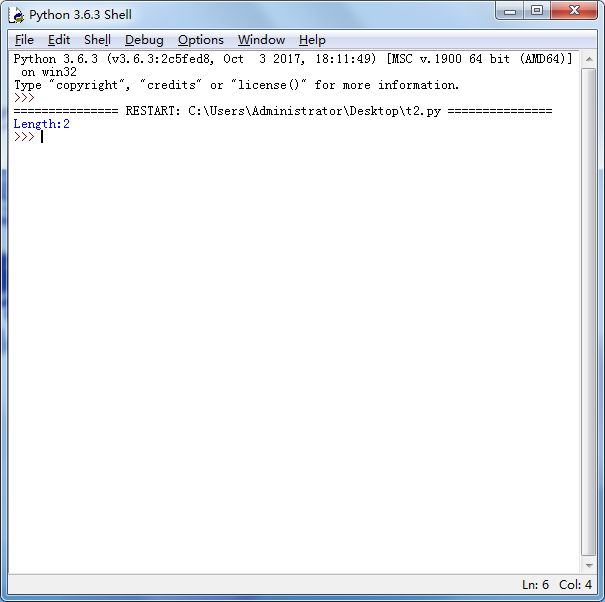
修改字典的元素



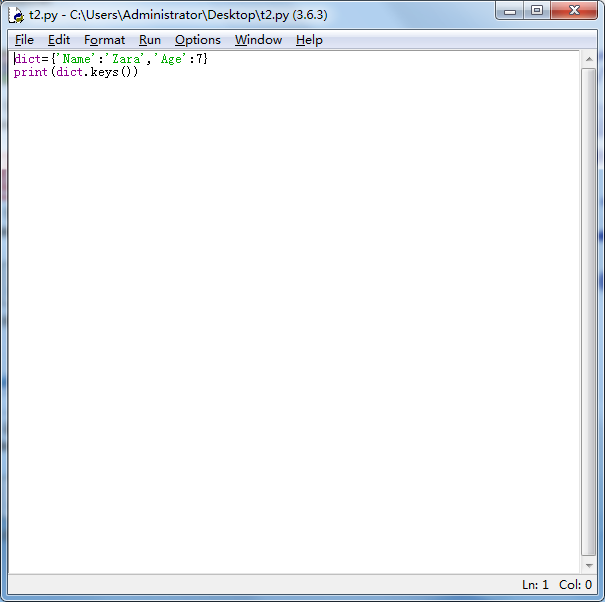


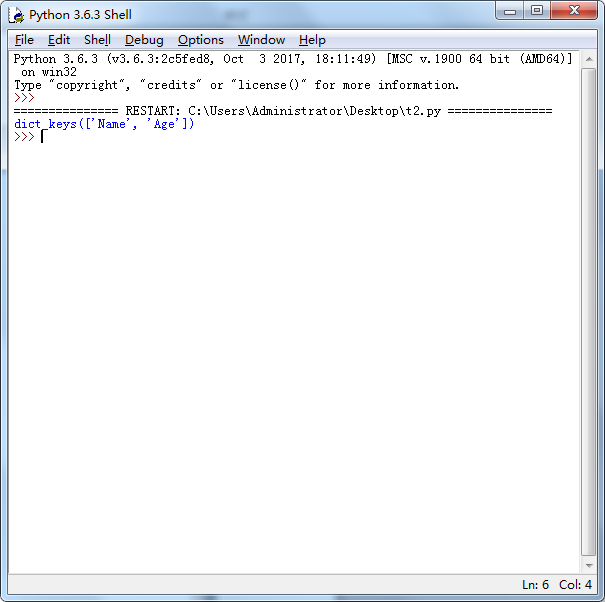
1.添加字典的元素2.删除字典的元素3.计算字典中键值对的个数



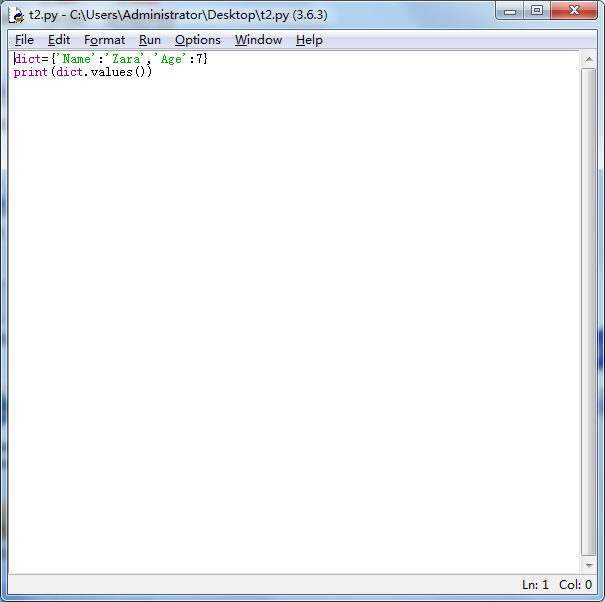


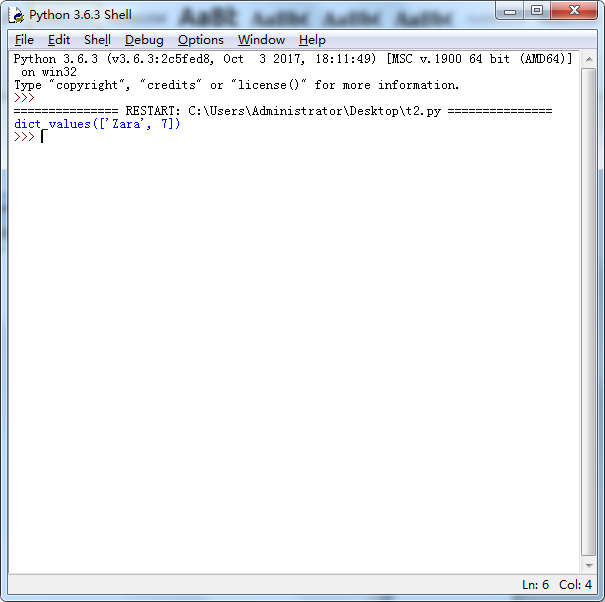
获取字典中键的列表

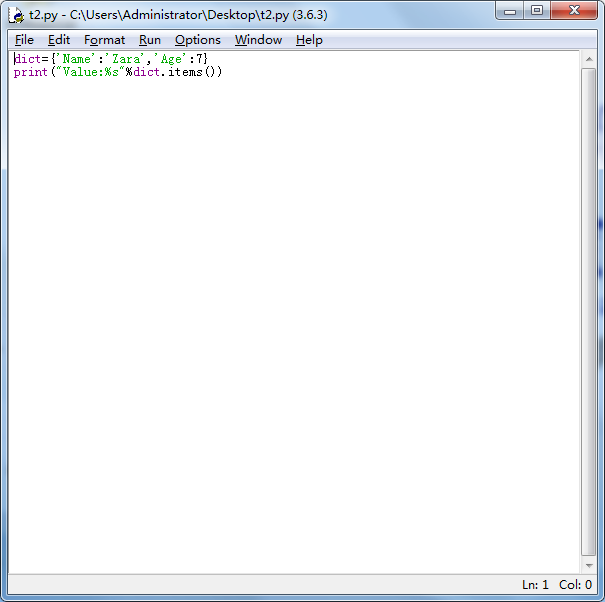


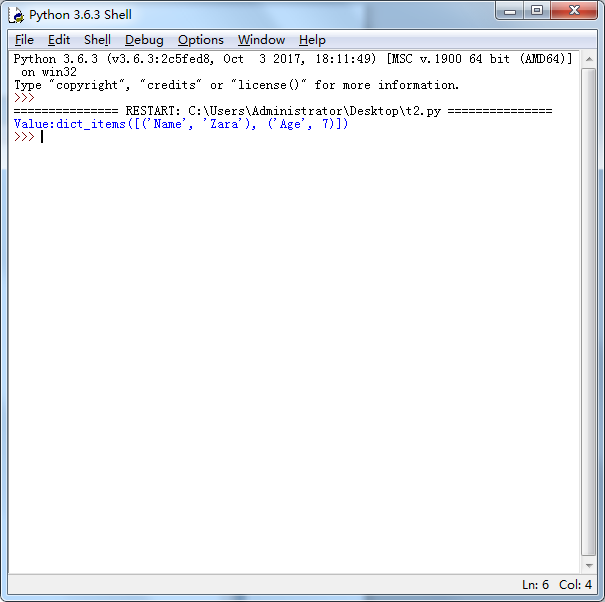


获取列表中字典的值

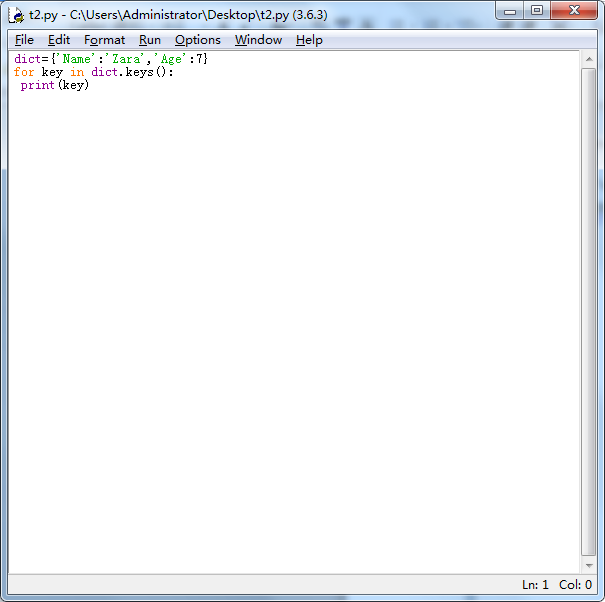


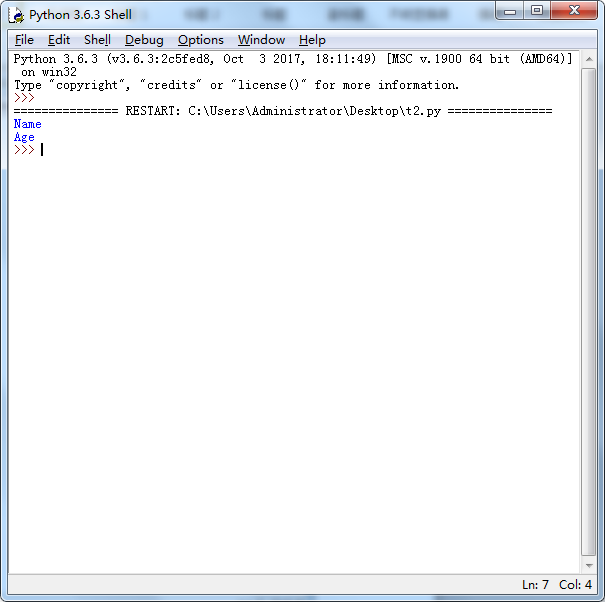






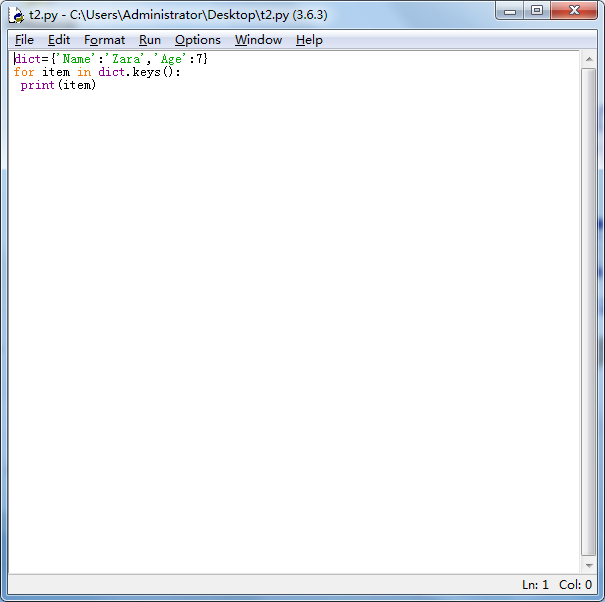
遍历字典的键key

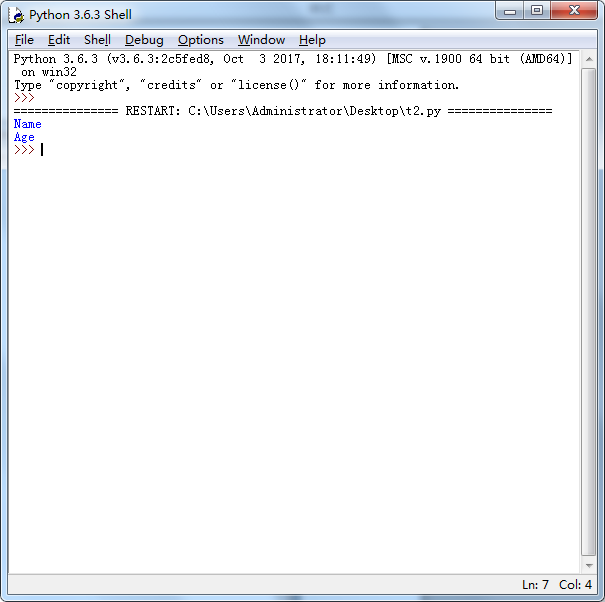




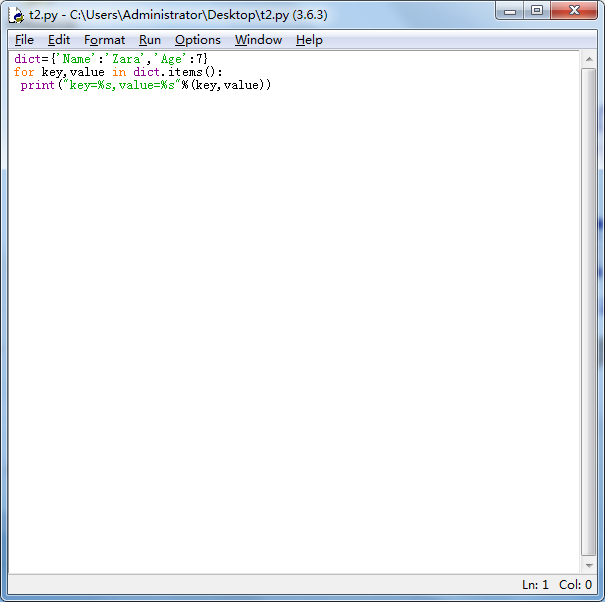
遍历字典的键value

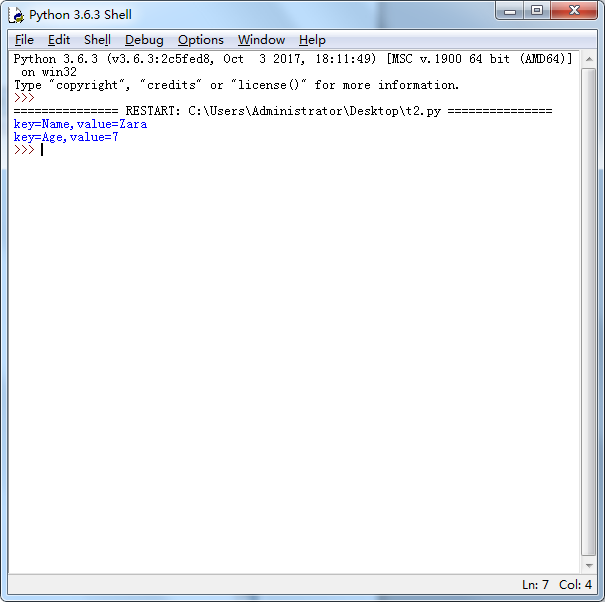
遍历字典的元素

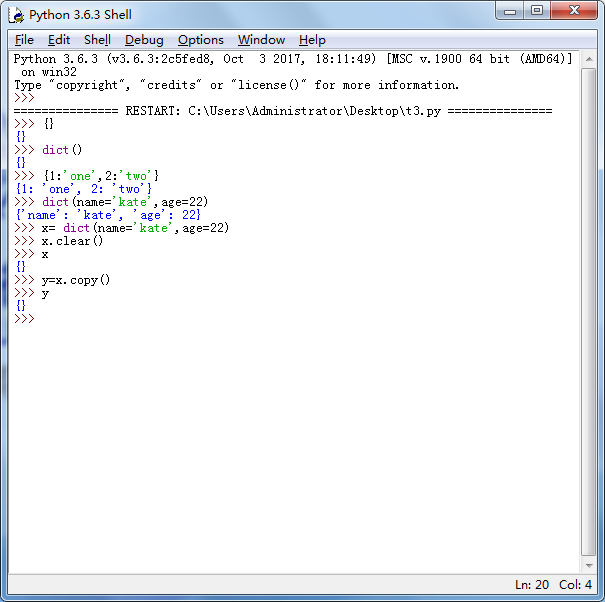




遍历字典的键值对







第六章函数

函数是组织好的，可重复使用的，用来实现单一或相关联功能的代码段，他能够提高应用的模块性和代码的重复利用率。

Python定义函数使用def关键字

def函数名（参数列表）

函数体

定义了函数之后，要想让这些代码能够执行，需要调用函数。通过“函数名（）”

Printinfo()

函数的返回值：如果希望定义的函数，可以计算任何两个数的和，我们在定义函数的时候，让函数接收数据，这就是函数的参数。

函数调用时，如果没有传递参数，则会使用默认参数。

函数的返回值是使用return语句来完成的

根据函数的参数和返回值，函数可以分为四种类型：

无参数，无返回值的函数

无参数，有返回值的函数

有参数，无返回值的函数

有参数，有返回值的函数

函数的嵌套调用

作用域：为了避免变量的名称发生冲突，所以Python引入了命名空间的概念。

命名空间是指名称到对象的映射

局部变量：在函数内部定义的变量。

局部变量的作用域是函数内部，意味着他只在定义它的函数中有效，一旦函数结束就会消失。

全局变量

递归函数：一个函数内部可以调用其他函数。但是，如果一个函数在内部不调用其他函数，而是自己本身的话，这个函数就是递归函数。

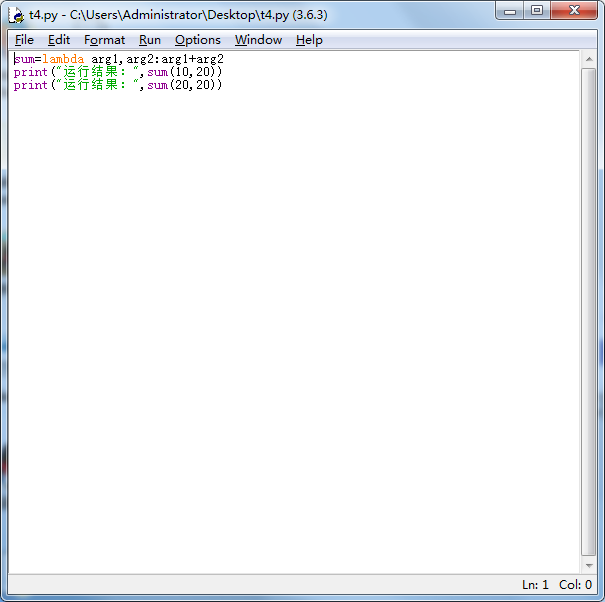
匿名函数：即使没有名称的函数也就是不再使用def语句定义的函数。

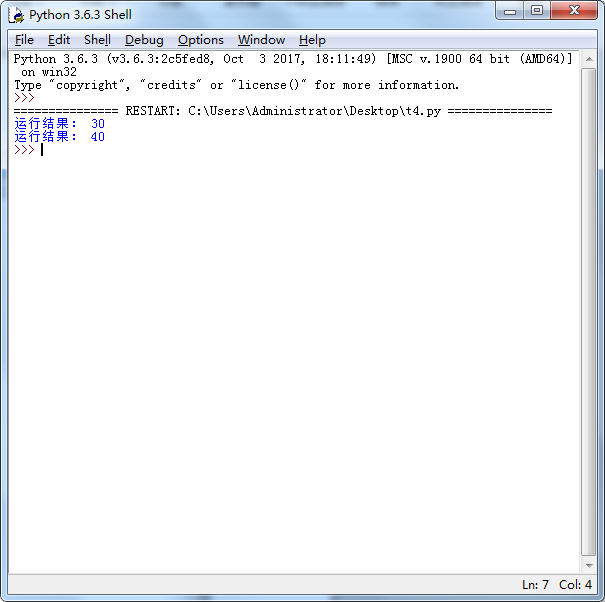
声明匿名函数：lambda Lambda[arg1,[,arg2…]]

参数传递

传不可变参数

传不可变参数





函数的调用

日期时间函数

在python中，通常有如下几种方式表示时间：

(1)时间戳;

时间戳表示的是从1970年1月1日00；00；00开始按秒计算的偏移量。

用time模块的strftime方法来格式化日期

日历函数

Calendar.firstweekday()

返回当前每周起始日期的设置

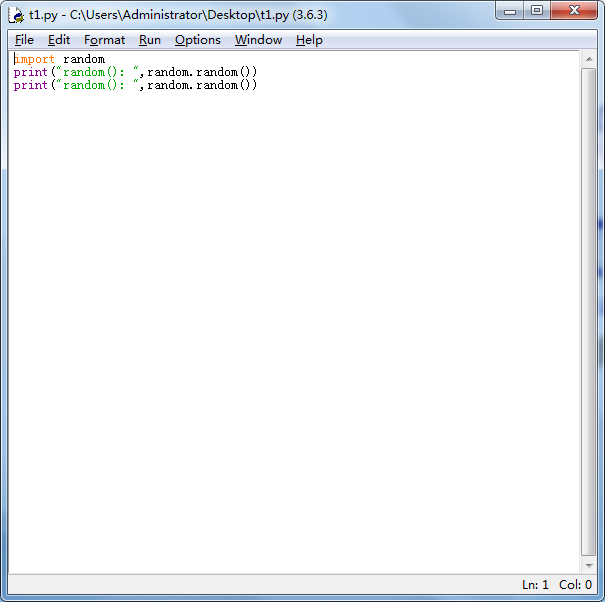
Calendar.isleap(year)

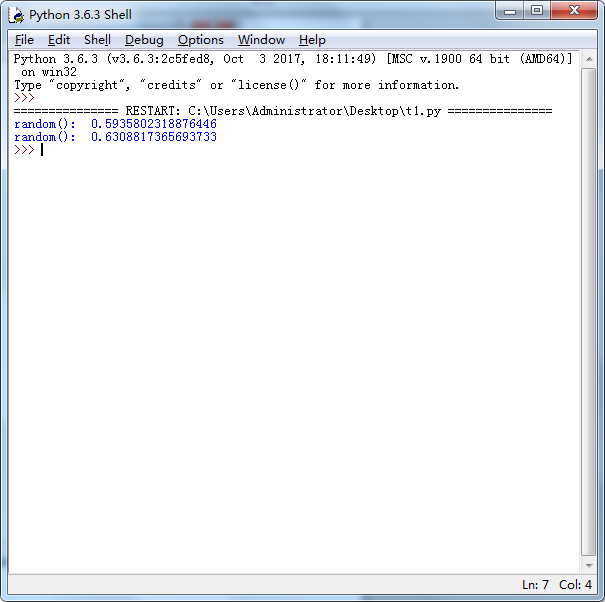
如果是闰年就返回true，否则为false

随机数函数

random.random()

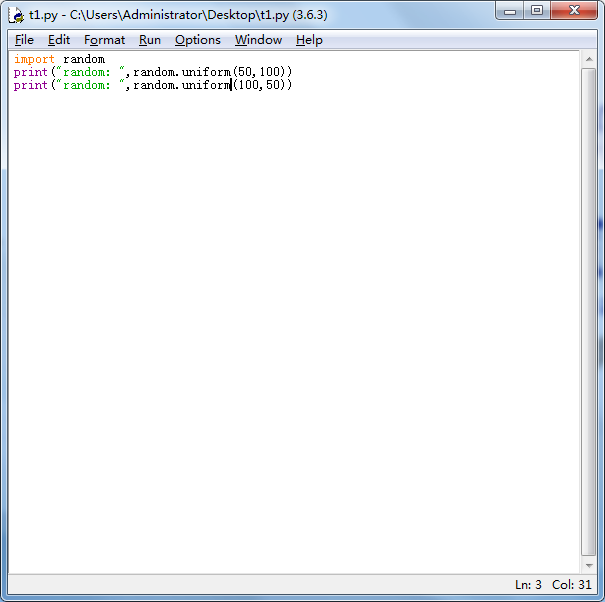
用于生成一个0到1的随机浮点数：0<=n<1.0

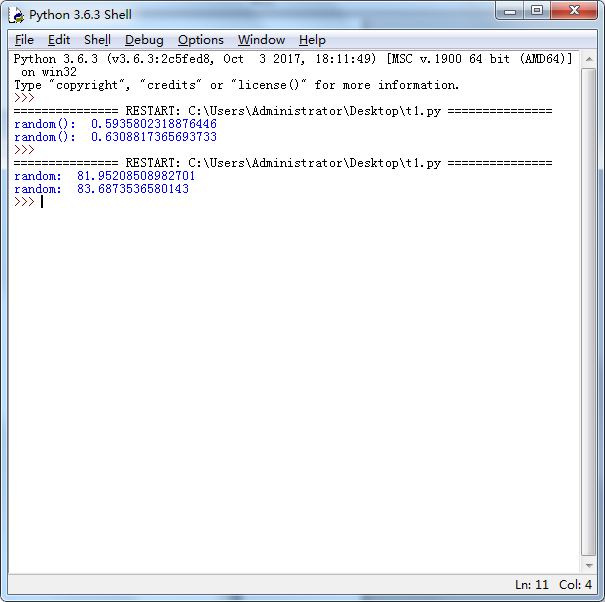




random.uniform(a,b)

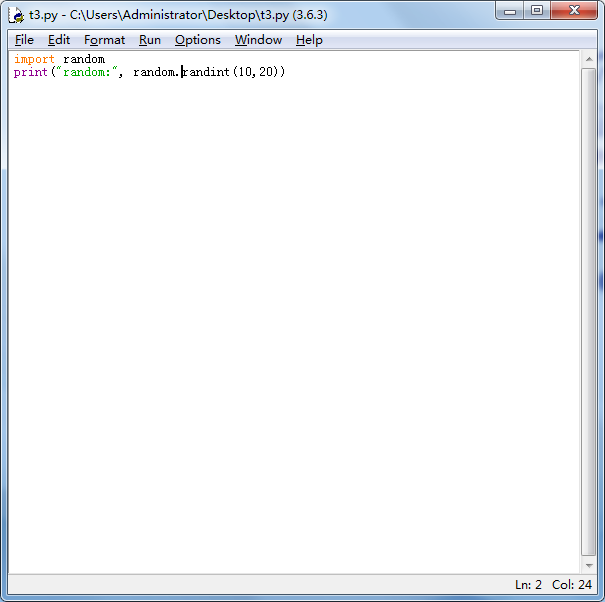
返回a,b之间的随机浮点数，范围[a,b]或[b,a]取决于四舍五入，a，不一定比b小。

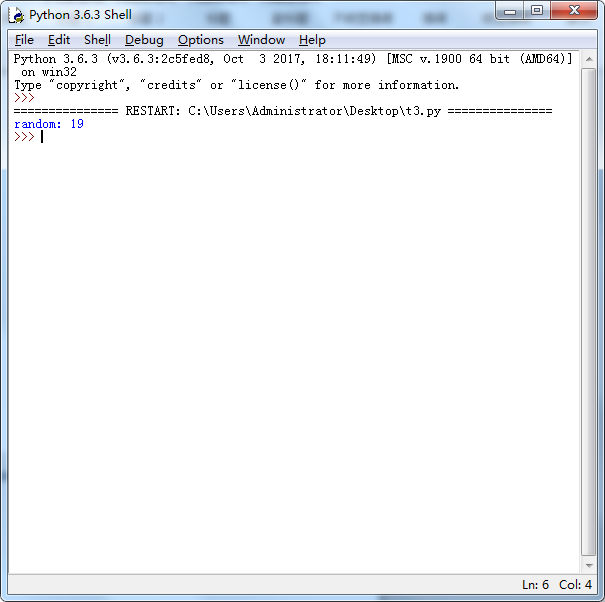


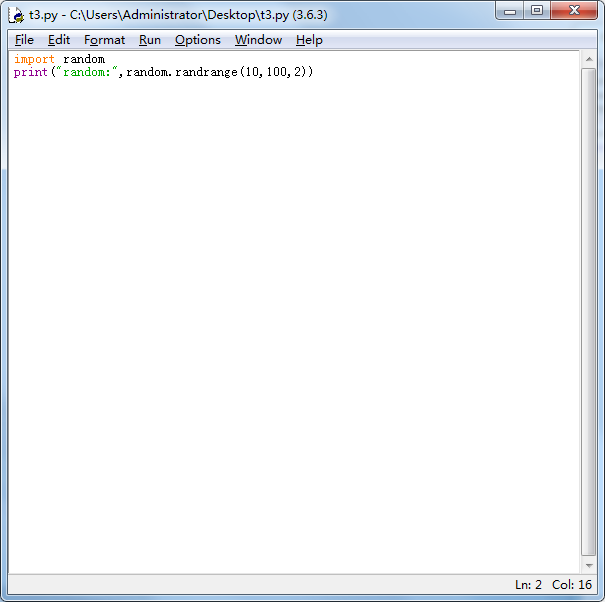


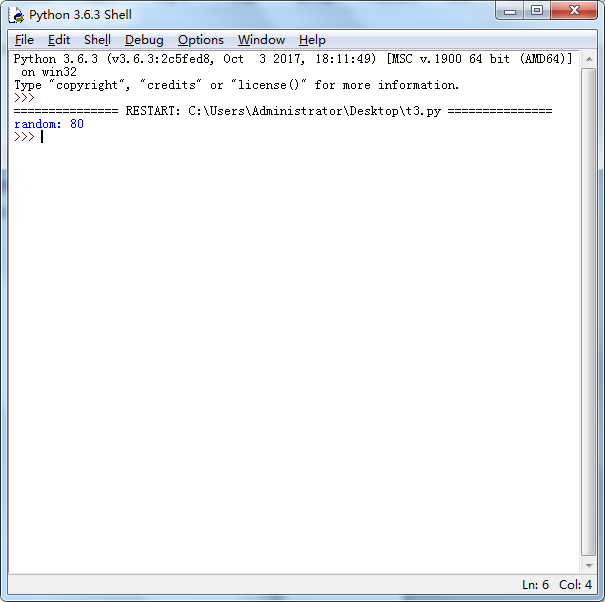
random.randint(a,b)

返回a,b之间的整数，范围[a,b]，注意：传入参数必须是整数，a一定比b小。



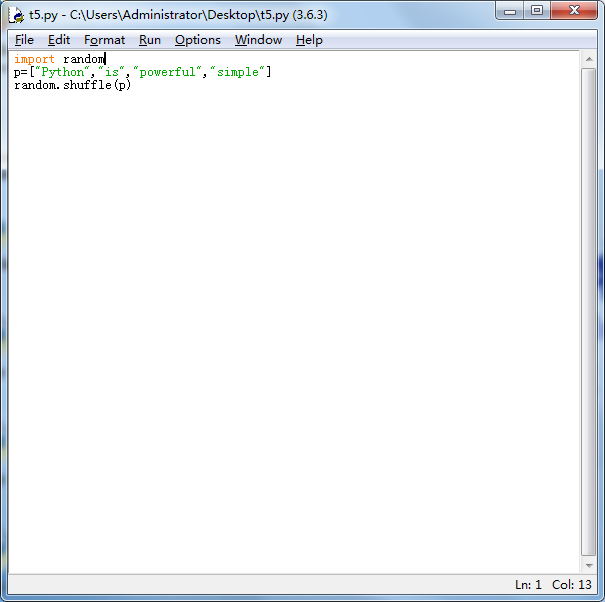


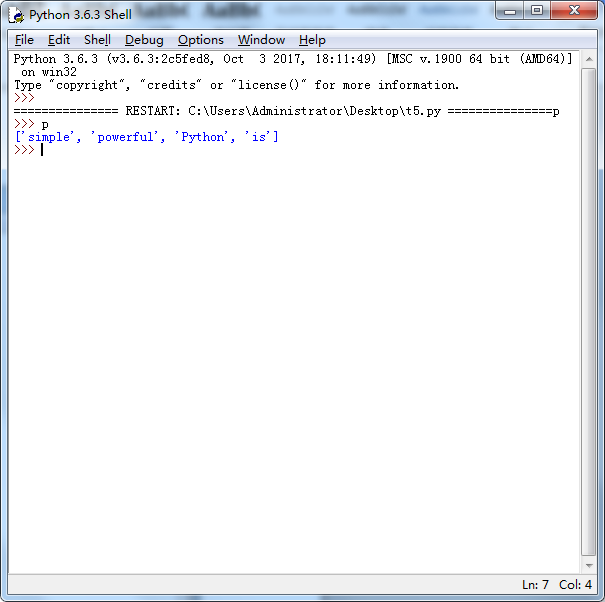




random.shuffle(x[,random])

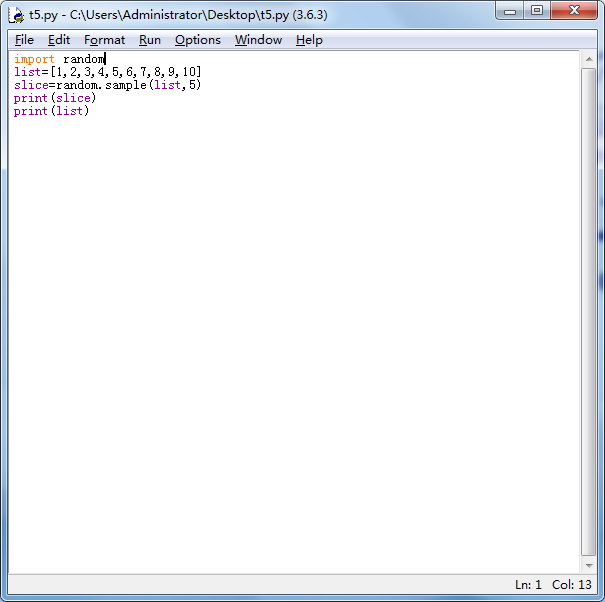
将列表中的元素打乱顺序，俗称洗牌

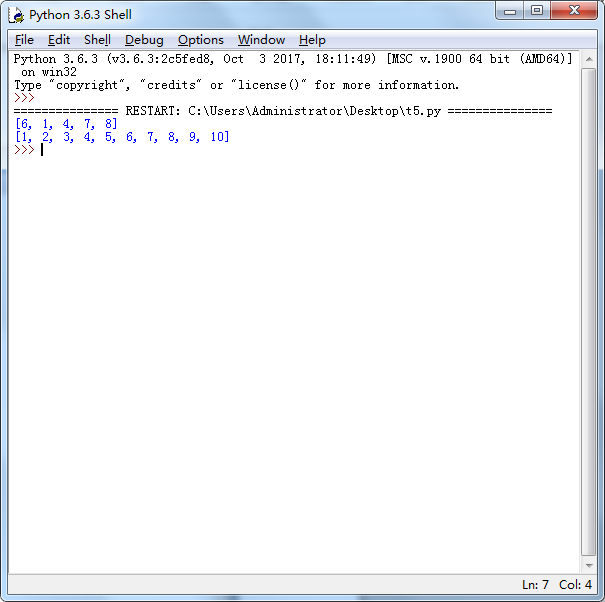




random.sample(sequence,k)

从指定序列中随机获取k个元素作为一个片段返回，sample函数不会修改原有原则。





第八章 文件操作

1．文件的打开

open（文件名，访问模式）

访问模式

r 默认模式，以只读的方式打开文件

w 打开一个文件只用于写入

a 打开一个文件用于追加

rb 以二进制格式打开一个文件只用于读

wb 以二进制方式打开一个文件只用于写入

ab 以二进制方式打开一个文件只用于追加

2.文件的关闭

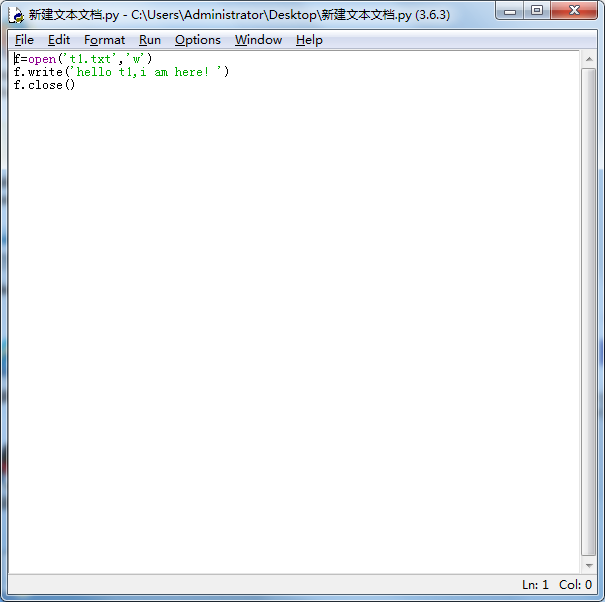
#新建一个文件，文件名为：test.txt

f=open(‘itheima.txt’,’w’)

#关闭这个文件

f.close()

写文件

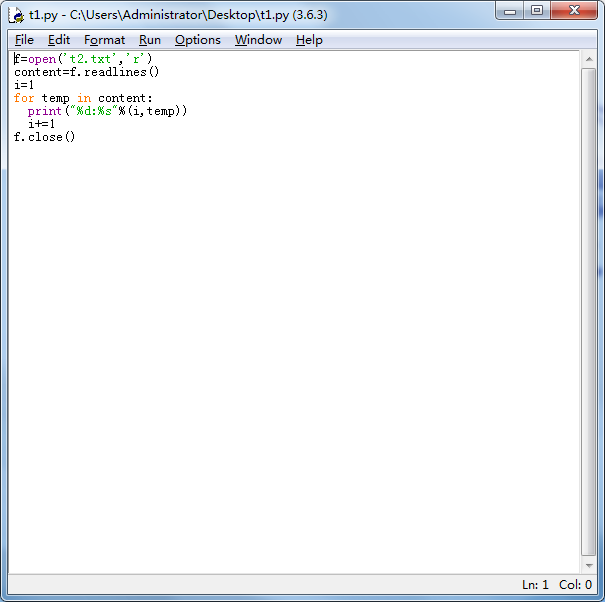


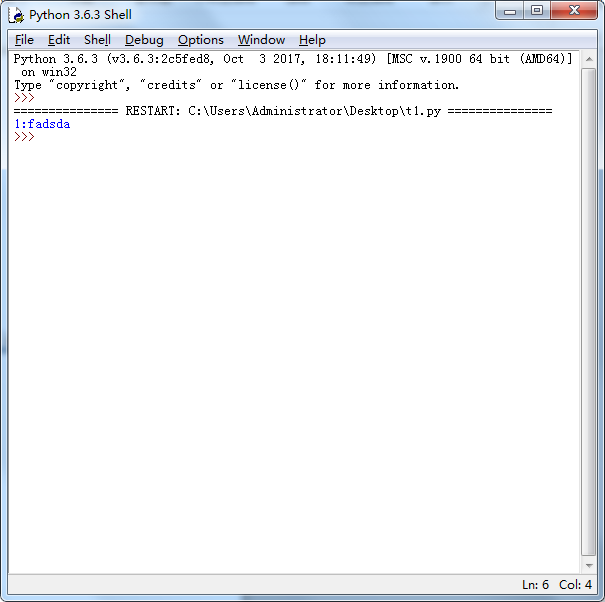
读文件

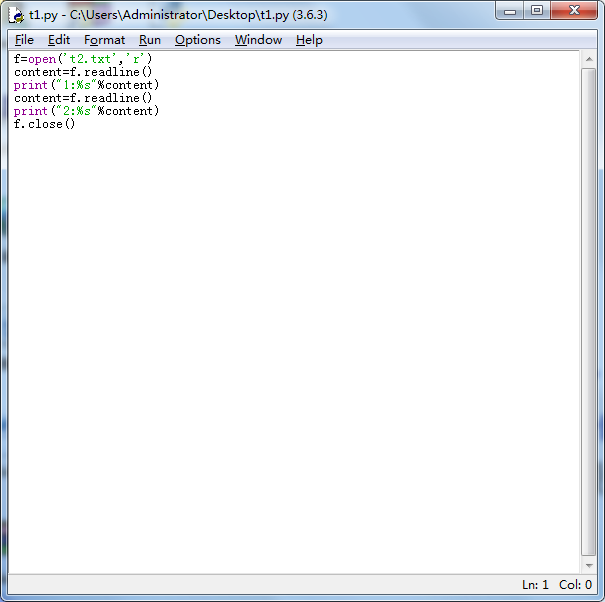
方法一：使用read方法读取文件

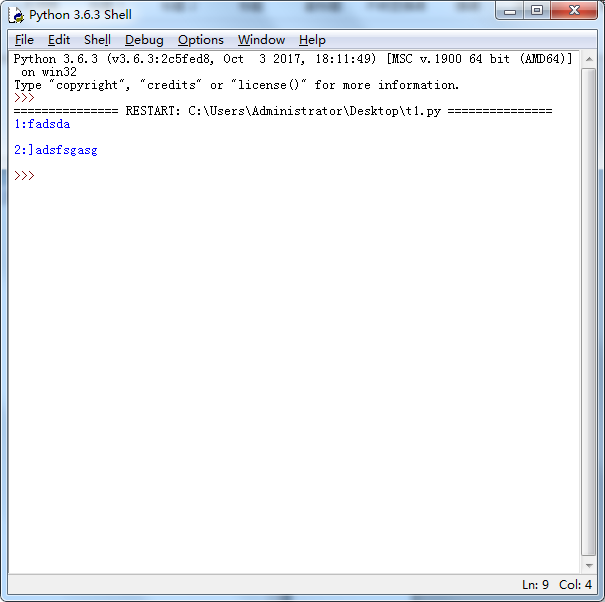
方法二：使用readlines方法读取文件

方法三：使用readline方法一行一行读数据



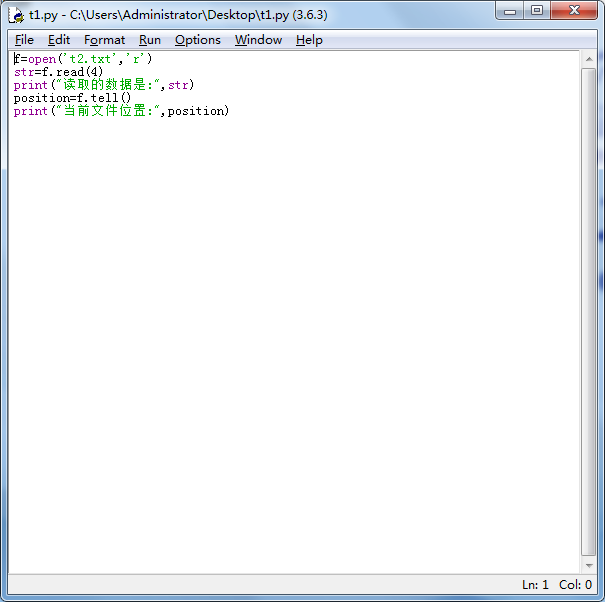


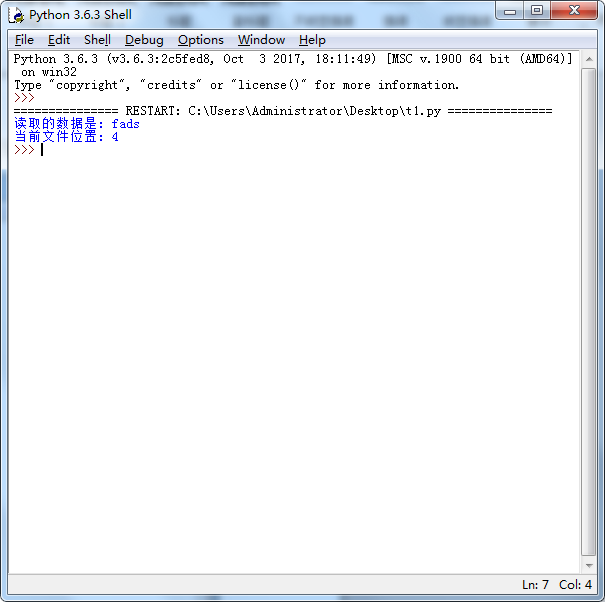




文件的定位读写

方式一：使用tell方法来获取文件当前读写位置





方式二：使用seek方法来获取文件当前位置的读写位置

seek(offset,from)方法包含两个参数：

from：表示方向，可以指定从哪个位置开始偏移

0：表示文件开头（默认值）

1：表示当前位置

2：表示文件末尾

文件的重命名和删除

文件的重命名

os模块中的rename（）方法可以完成文件的重命名。

rename(需要修改的文件名，新的文件名)

文件的删除

os模块中的remove（）方法可以完成文件的删除操作。

remove(待删除的文件名)

文件的相关操作

创建文件夹

1. os模块的mkdir的方法来创建文件夹。
2. 获取当前目录

import os

os.getcwd()

1. 改变默认目录

import os

os.chadir(“../”)

4.获取目录列表

import os

os.listdir(“../”)

5.删除文件夹

Import os

os.rmdir(“张三”)

面向对象编程

具有相似特征和行为的事物的集合统称为类

类的名称：类名，首字母必须大写，比如Person

类的属性：一组数据，比如性别

类的方法：允许进行操作的方法，比如说话

使用class关键字来声明一个类

class类名：

类的属性

类的方法

根据类创建对象的语法格式如下：

对象名=类名（）

要想给对象添加属性，可以通过

对象名.新的属性名=值

构造方法指的是\_init\_方法

当创建类的实例的时候，系统会自动调用构造方法，从而实现对类进行初始化的操作。

当删除一个对象来释放所占用的资源的时候，Python解释器默认会调用另一个方法，这个方法就是\_del\_()方法。

\_del\_方法被称为析构方法

self的使用

在方法的列表中，第一个参数永远是self。

当某个对象调用方法的时候，Python解释器会把这个对象作为第一个参数传给self，开发者只需要传递后面的参数就可以了。