Python笔记（四）：异常处理机制与 open（）

（一）  异常处理机制概述

就像日常生活中会遇到各种意外情况一样（例如：你可能考虑过如果中了500w该怎么做），代码运行过程中也会遇到这种意外情况，python提供了这么一种机制，处理意外情况（就像如果中了5百万你希望代码怎么做一样）。

注：不处理这种意外情况，代码就会崩溃，后面全部代码都将停止运行。

实际应用中，主要分为3步：

（1）   圈出你认为可能出异常的代码。

（2）   设置你认为可以忽略的异常类型。

（3）   异常出现时，你希望的处理方式。

异常类型和格式可查看：http://www.runoob.com/python/python-exceptions.html

（二）  open

the\_file = open(‘测试.txt’)

参数及方法可参考：http://www.runoob.com/python/python-func-open.html

（三）  例子

（1）   新建一个文件 测试.txt，输入以下内容

张三：今天天气真好.

李四：是的，真的太好了。

张三：阳光明媚。

李四：鸟语花香。

王五：装逼现场：请带好安全帽

张三：难得好天气，今天就不偷懒了。

李四：能把乞讨说的这么高大上的也就只有你了。

分隔符

张三：难得好天气，今天就不偷懒了。

李四：能把乞讨说的这么高大上的也就只有你了。

（2）   使用open读取上面的文件，结合异常处理机制进行说明

try:  
 the\_file = open(r'C:\Users\123456\Desktop\测试.txt',encoding='UTF-8')  
 for each\_line  
in the\_file:  
      try:  
          (role,line\_spoken) =  
each\_line.split("：",1)  
          #以：分割数据，参数1表示只分解为2部分。

#例如：分割第一行数据，就是 role = ‘张三’ line\_spoken = 今天天气真好.  
          # 不设置这个参数就是尽可能分割，那么读取 王五：装逼现场：请带好安全帽  
数据时会出错  
          #分割“分隔符”这行数据时会出现ValueError异常（因为没有 ：），不忽略这个错误后面全部数据都会不能读取  
          print(role,end="")  
          print(" said"+":  
",end="")  
          print(line\_spoken,end="")  
      except ValueError:  
          # 出现ValueError时，直接输出 each\_line的值  
          print(each\_line,end="")  
 the\_file.close()  
except IOError:   
    #找不到文件时提示文件不存在  
    print("文件不存在！")

（四）  事先处理可能出现的错误 和 错误出现后再处理2种方式

（1）    
实际上我们也可以事先处理可能出现的错误，通过下面的代码实现。

（2）   但是存在的问题是，你可能需要写大量的逻辑代码去处理这种情况，这样功能代码可能只占极小的一部分，导致代码要实现的功能不清晰。通过异常处理机制就不存在这个问题，代码要实现的功能一目了然。

if not each\_line.find('：') == -1:  
    # find() 找不到目标字符串时会返回 -1 ，not 取反  
      else:  
       print(each\_line,end="")