

文件与异常

上海财经大学 信息管理与工程学院

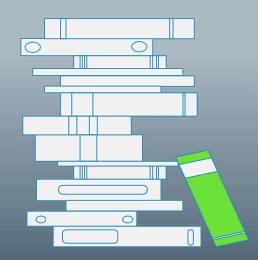


1 2 2 文件及文件基本操作

3 --- 异常和异常处理



文件及文件基本操作



文件简介

文件,是一组相关数据的集合。组成文件的数据可以是ASCII编码,也可以是二进制编码。

文件名都包含两个部分:主文件名和扩展名,两者之间用"." 隔开。

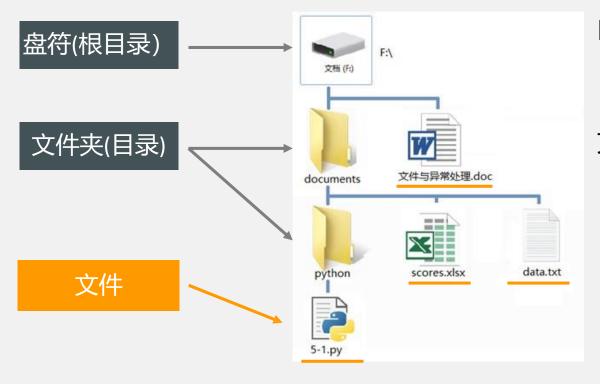
主文件名.扩展名

由用户自行命名, 用来与其他文件加以区别 文件类型对应缩写, 用来指定打开和操作该文件的应用程序

文件——用来组织和管理一组相关数据的;

目录——是用来组织和管理一组相关文件的。

目录,又可称为文件夹,可以包含文件,也可以包含其他目录。



F盘根目录下包含:

文件夹 "documents" 和 文件 "文件与异常处理.doc

文件夹 "documents" 下包含:

文件夹 "python" 、文件 "scores.xlsx" 和文件 "data.txt"

文件夹 "python" 下包含:

文件 "5-1.py"

文件保存的位置被称为路径。

绝对路径——从文件所在驱动器名字(又称为"盘符")开始描述文件的保存位置。

分隔符

F:\documents\python\5-1.py

路径字符串

"F:\\documents\\python\\5-1.py"

简便的 路径字符串

r "F:\documents\python\5-1.py"



文件保存的位置被称为路径。

相对路径——从当前工作目录开始描述文件的保存位置。

当前工作目录, cwd, 默认为安装目录。

可通过os库的方法查看和重设。

>>> import os

>>> os.getcwd()

默认的当前工作目录

'C:\\Programs\\Python\\Python37-32'

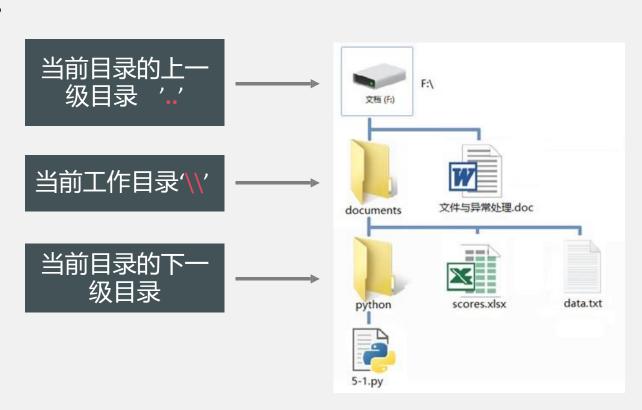
#修改当前工作目录为F盘根目录下documents

>>> os.chdir('F:\\documents')

>>> os.getcwd()

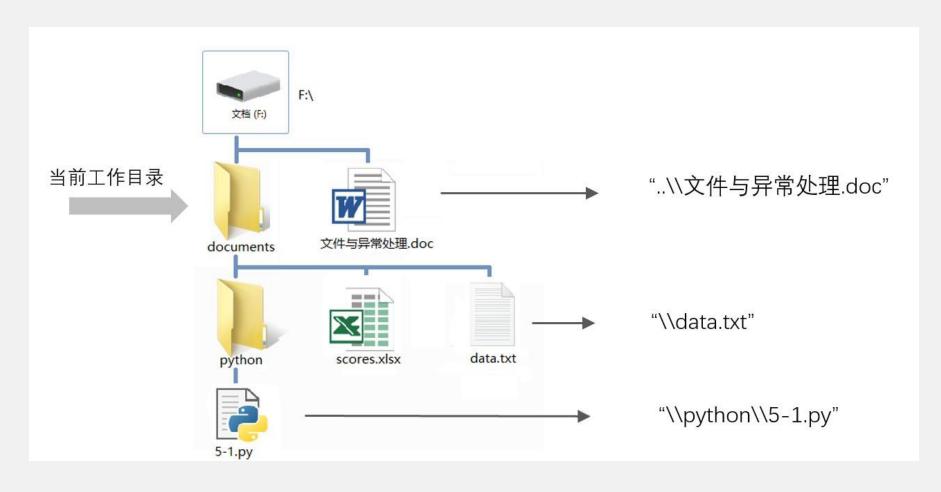
重新设置后的当前目录

'F:\\documents'



文件保存的位置被称为路径。

相对路径——从当前工作目录开始描述文件的保存位置。





文件的操作一般包含三个步骤:

打开文件 → 读文件或者写文件 → 关闭文件

将文件从外部存储器调入内存的过程

指定打开文件的类型和操作文件的方式

File对象名 = open(文件路径字符串,模式字符)

可以采用绝对路径,也可以采用相对路径



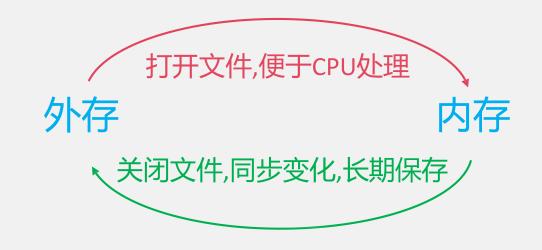
File对象名 = open(文件路径字符串,模式字符)

打开模式	文件类型	操作方式	文件不存在时	是否覆盖写
'r'	文本文件	只可读文件	报错	-
'r+'		可读可写	报错	是
'w'		只可写文件	新建文件	是
'w+'		可读可写	新建文件	是
'a'		只可写文件	新建文件	否,从EOF处开始追加写
'a+'		可读可写	新建文件	否,从EOF处开始追加写
'rb'		只可读文件	报错	-
'rb+'	二进制文件	可读可写	报错	是
'wb'		只可写文件	新建文件	是
'wb+'		可读可写	新建文件	是
'ab'		只可写文件	新建文件	否,从EOF处开始追加写
'ab+'		可读可写	新建文件	否,从EOF处开始追加写



关闭文件,是将文件从从内存保存回外存的过程。

File对象名 .close()

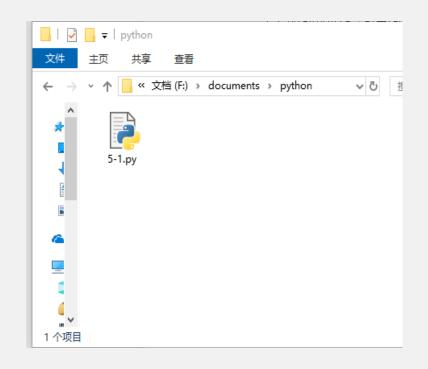


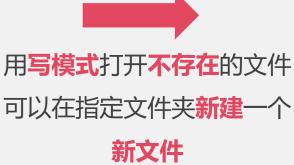
长期保存文件

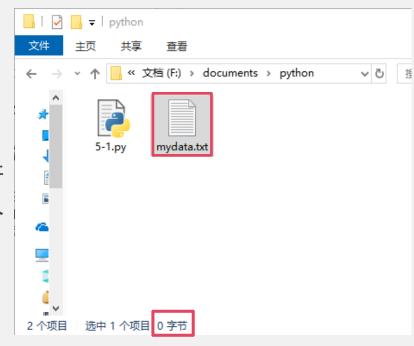
临时存放处理的中间结果



文件操作一写文件



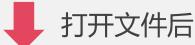






>>> os.chdir("F:\\documents\\python")

>>> file=open("mydata.txt",'w')



文件与文件对象file关联,

后续文件操作通过文件对象实现

文件操作—写文件

- 用文件对象的write方法写文件。
 - 具体的格式: 文件对象.write(写入字符串)
 - 将指定的字符串写入文件当前插入点位置

```
>>> file=open("mydata.txt",'w')
```

>>> file.write("飞雪连天射白鹿")

7 #回显写入的字符数

>>> file.write("笑书神侠倚碧鸳")

7

>>> file.close()



文件操作—写文件

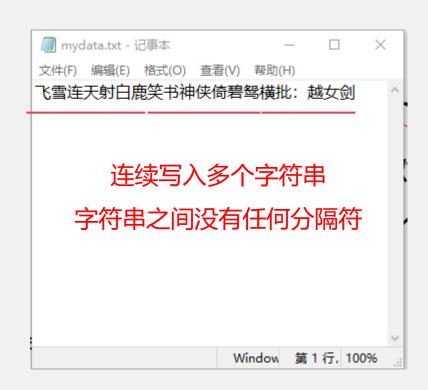
- 用文件对象的writelines方法写文件。
 - 具体的格式: 文件对象.write(字符串序列)
 - 以序列的形式接受多个字符串作为参数,一次性写入多个串

```
>>> file=open("mydata.txt",'w')
```

>>> ls=["飞雪连天射白鹿","笑书神侠倚碧

鸳"]

- >>> file.writelines(ls)
- >>> file.writelines(["横批: 越女剑"])
- >>> file.close()



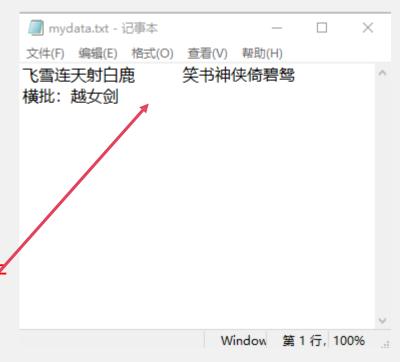
文件操作—写文件

- 用文件对象的writelines方法写文件。
 - 具体的格式: 文件对象.write(字符串序列)
 - 以序列的形式接受多个字符串作为参数,一次性写入多个串

```
>>> file=open("mydata.txt",'w')
```

- >>> ls=["飞雪连天射白鹿\t","笑书神侠倚碧鸳\n"]
- >>> file.writelines(ls)
- >>> file.writelines(["横批:越女剑\n"])
- >>> file.close()

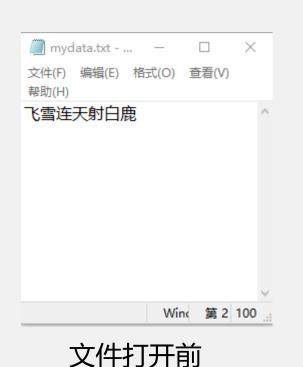
可以通过在字符串结尾加入分隔符将分隔符写入文件



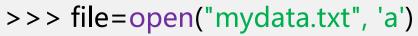


文件操作一写文件

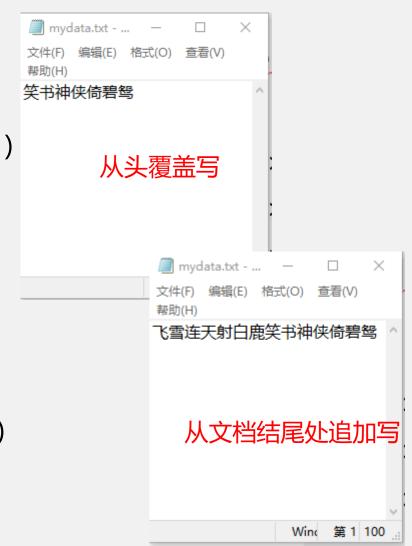
write方法和writelines方法也可以用于追加模式('a'或者'a+')



- >>> file=open("mydata.txt",'w')
- >>> file.write("笑书神侠倚碧鸳")
- >>> file.close()



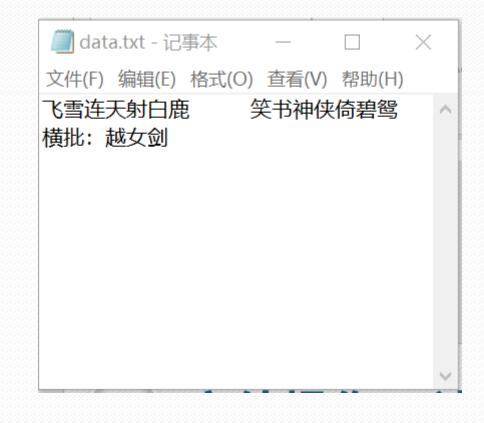
- >>> file.write("笑书神侠倚碧鸳")
- >>> file.close()



文件操作一读文件

- 用文件对象的read方法写文件。
 - 具体的格式:字符串变量 = 文件对象.read()
 - 读出文件所有内容并作为一个字符串返回

```
>>> file = open("data.txt",'r')
>>> text = file.read()
>>> text
'飞雪连天射白鹿\t笑书神侠倚碧鸳\n
横批: 越女剑\n
                   分隔符将一起读出
>>> file.close()
```



文件操作一读文件

- 用文件对象的readline方法写文件。
 - 具体的格式:字符串变量 = 文件对象.readline()
 - 读出文件中当前行,并以字符串的形式返回

```
>>> file = open("data.txt",'r')
```

>>> text = file.readline()

>>> text

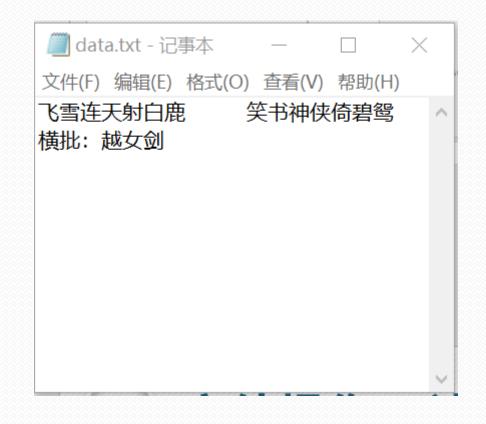
'飞雪连天射白鹿\t笑书神侠倚碧鸳\n'

>>> text = file.readline()

每次读一行

>>> text

'横批:越女剑\n'





文件操作一读文件

将文件的行处理, 转换为列表的元素处理

- ·用文件对象的readlines方法写文件。
 - 具体的格式:字符串变量 = 文件对象.readlines()
 - 读出文件所有内容并作为一个列表返回

```
>>> file = open("data.txt",'r')
>>> ls = file.readlines()
>>> ls
['飞雪连天射白鹿\t笑书神侠倚碧鸳\n',
'横批: 越女剑\n']
—行作为列表的一个元素
```



文件操作—练习

请将第四章例4-6中的问卷调查结果用文本文件"result.txt"保存。编写程序读文件然后统计各评语出现的次数,并将最终统计结果追加至"result.txt"文件中

【例 4-6】为了监督伙食质量,食堂向学生发起了一次简短的问卷调查。请大家在"非常满意"、"满意"、"一般"、"不满意"中选择一个评语评价食堂当天的伙食情况。最后食堂回收了 90 份问卷,并将所有的评语都汇总成了一个字符串:

"不满意,一般,很满意,一般,不满意,很满意,满意,一般,一般,不满意,满意,满意,满意,满意,满意,满意,满意,不满意,满意,一般,不满意,满意,不满意,满意,一般,很满意,不满意,满意,不满意,很满意,不满意,很满意,满意,很满意,不满意,很满意,不满意,很满意,不满意,满意,不满意,满意,不满意,满意,不满意,满意,不满意,满意,不满意,满意,不满意,满意,不满意,满意,不满意,满意,不满意,满意,不贵意,满意,不贵。,积贵,一般,一般,一般,不满意,不满意,满意,很满意,很满意,很满意,不满意,很满意,一般,一般,

请编写程序,利用列表统计其中各个不同评语出现的次数,并找出出现最多的评语。

```
文件
```

2 7

2 8

文件操作—练习

```
# 食 堂 伙 食 质 量 问 卷 调 查 。 4 - 6 . p y
    comments = ['不满意', '一般', '满意', '很满意']
    result = "不满意,一般,很满意,一般,不满意,很满意,满意,一般,一般,"\
       "不满意,满意,满意,满意,满意,满意,一般,很满意,一般,满意,"\
       "不满意,满意,一般,不满意,满意,不满意,满意,很满意,很满意,"\
       "满意,满意,不满意,满意,不满意,满意,一般,很满意,不满意,"\
       "一般,很满意,满意,很满意,不满意,很满意,不满意,很满意,"\
       "满意,满意,很满意,一般,很满意,满意,满意,很满意,不满意,"\
       "很满意,满意,不满意,满意,不满意,满意,很满意,很满意,满意,很满意,"\
       "一般,很满意,很满意,很满意,不满意,满意,一般,一般,一般,"\
9
       "一般,不满意,不满意,满意,很满意,很满意,满意,满意,很满意,"\
1 0
       "很满意,一般,一般,很满意,一般,一般,满意,很满意,一般"
1 1
1 2
    resultList = result.split(',')
1 3
1 4
    commentCnts = [0] * 4
1 5
    for i in range(4):
1 6
       commentCnts[i] = resultList.count(comments[i])
1 7
1 8
    most = max(commentCnts)
1 9
    mostComment = comments[commentCnts.index(most)]
2 0
2 1
    print("根据统计,对今天伙食感觉:")
2 2
    print("'很满意'的学生{}人;".format(commentCnts[3]))
2 3
    print("'满意'的学生{}人;".format(commentCnts[2]))
2 4
    print("'一般'的学生{}人;".format(commentCnts[1]))
2 5
    print("'不满意'的学生{}人。".format(commentCnts[0]))
2 6
```

print("调 查 结 果 中 , 出 现 次 数 最 多 的 评 语 是 ", mostComment)

读文件 返回字符串

结果写回文件



文件操作一练习

- Step1:
 - 新建文件,以字符串存放90个调查 结果, 如右图。
- Step2:
 - 编写读文件函数,返回字符串。

```
def readFile(fileName):
  file = open(fileName,'r')
  res = file.read()
  return res
```



III result.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

不满意,一般,很满意,一般,不满意,很满意,满意,一般,一般, 不满意,满意,满意,满意,满意,一般,很满意,一般,满意 ,不满意,满意,一般,不满意,满意,不满意,满意,很满意,很满 意,满意,满意,不满意,满意,不满意,满意,一般,很满意,不满 意,一般,很满意,满意,很满意,不满意,很满意,不满意,很满 意,满意,满意,很满意,一般,很满意,满意,满意,很满意,不满 意,很满意,满意,不满意,满意,满意,很满意,满意,很 满意,一般,很满意,很满意,很满意,不满意,满意,一般,一般, -般,一般,不满意,不满意,满意,很满意,很满意,满意,满意, 很满意,很满意,一般,一般,很满意,一般,一般,满意,很满意, 一般

 \times

文件操作—练习

Step3:

- 统计各评语出现的次数, 结果放在字典里。

```
def countComments(result):
    resList=result.split(',')
    commentCnts={}
    for res in resList:
        commentCnts[res]=commentCnts.get(res,0)+1
    return commentCnts
```



文件操作一练习

Step4:

- 调用读文件函数和统计函数,并将结果追加至result文件中,如图。

result = readFile("d:\\result.txt")
dicCnts = countComments(result)

```
file = open("d:\\result.txt",'a')
most=max(dicCnts.values())
for k,v in dicCnts.items():
    if v == most:
        mostComment = k
```

break

满意,很满意,很满意,一般,一般,很满意,一般,一般,满意,很满意,一般

根据统计,对今天伙食感觉: '很满意'的学生25人; '满意'的学生28人; '一般'的学生20人; '不满意'的学生17人。 调查结果中,出现次数最多的评语是满意

☐ result.txt - 记事本

file.write("根据统计,对今天伙食感觉:\n")
file.write("'很满意'的学生{}人;\n".format(dicCnts["很满意"]))
file.write("'满意'的学生{}人;\n".format(dicCnts["满意"]))
file.write("'一般'的学生{}人;\n".format(dicCnts["一般"]))
file.write("'不满意'的学生{}人。\n".format(dicCnts["不满意"]))
file.write("出现次数最多的评语是{}\n".format(mostComment))
file.close()



CSV文件操作



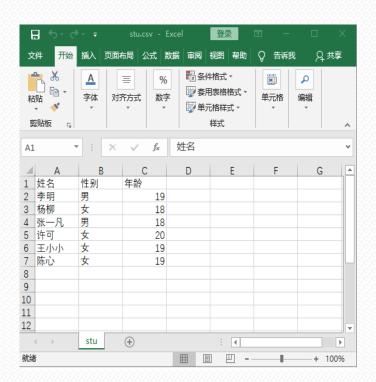
CSV文件简介

CSV,是英文Comma-Separated Values的缩写,翻译为"逗号分隔值"。CSV文件可以理解为以带逗号分隔(也可以是其他简单字符分割)的纯文本形式存储表格数据的文件。

stu.csv文件 可以用记事本, 也可用Excel打开







CSV文件简介

CSV文件,一般具有以下特征:

- 1. 纯文本,使用某个字符集,比如ASCII、Unicode、EBCDIC或GB2312;
- 2. 由记录组成,一行对应一条记录,每行开头不留空格;
- 3. 每条记录被英文半角分隔符(可以使逗号、分号、制表符等)分割为字段;
- 4. 每条记录都有同样的字段序列;
- 5. 如果文件包含字段名,字段名写在文件第一行;
- 6. 不包含特殊字符,文件中均为字符串。

CSV文件广泛应用于在不同程序之间转移表格数据

CSV文件操作

针对CSV文件, Python提供了内置同名模块csv进行读写操作,读操作使用模块中的reader对象,写操作使用模块中的writer对象。
CSV文件进行读、写操作前也需要先打开文件。

>>> import csv

>>> with open(文件名,读写模式) as file对象名:
#读写操作

用with子句打开的文件,当文件读操作结束后,系统自动调用close方法关闭文件。

CSV;

CSV文件操作一读文件

```
>>> import csv
>>> with open("stu.csv", 'r' ) as stucsv:
       reader = csv.reader(stucsv)
       for row in reader:
              print(row)
  ['姓名','性别','年龄']
  ['李明', '男', '19']
  ['杨柳', '女', '18']
  ['张一凡', '男', '18']
  ['许可','女','20']
  ['王小小', '女', '19']
  ['陈心', '女', '19']
```

一行对应一个列表 列表元素都是字符串

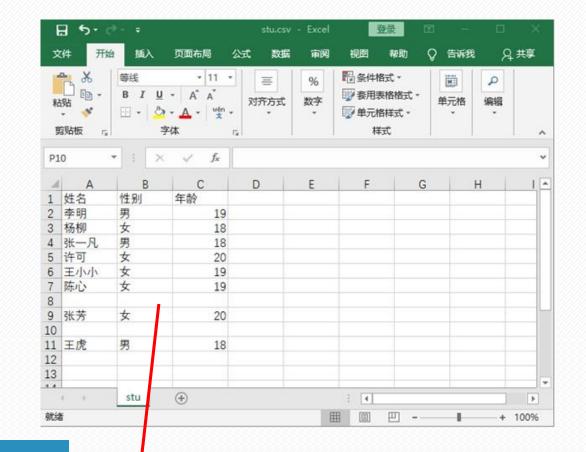


使用csv模块读取CSV文件数据

- 先创建一个reader对象。
- 然后通过迭代遍历reader对象读出文件中的每一行

CSV文件操作—写文件

```
>>> import csv
>>> with open("stu.csv", 'a') as stucsv:
writer=csv.writer(stucsv)
writer.writerow(['张芳','女','20'])
writer.writerow(['王虎','男','18'])
```



使用csv模块写CSV文件时

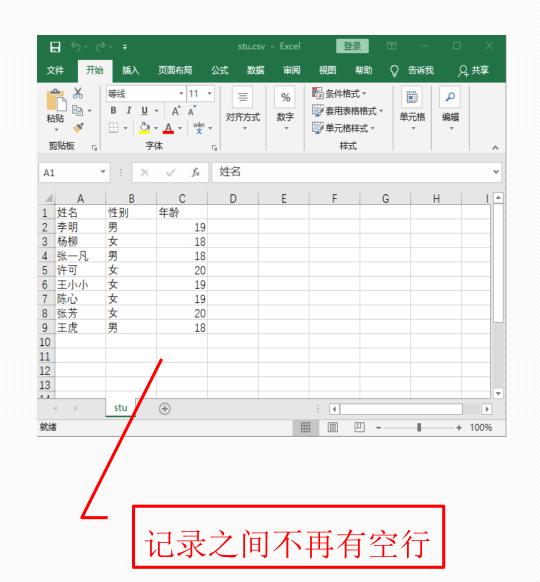
- 使用writer对象的writerow方法按行写入。
- 写入文件的一行内容要用列表存储

记录之间出现了空行, 读文件的时候会出现错误

CSV文件操作—写文件

```
>>> import csv
>>> with open("stu.csv", 'a', newline=") as stucsv:
writer=csv.writer(stucsv)
writer.writerow(['张芳','女','20'])
writer.writerow(['王虎','男','18'])
```

打开文件时增加一个参数 "newline="" , 指明在写入新的记录后不插入空行



>>> import csv

writerows方法只接受一个序列作为参数, 可以是列表, 也可以是元组。

>>> with open("stu.csv", 'a', newline +") as stucsv: writer=csv.writer(stucsv)

writer.writerows([['张芳','女','20'],['王虎','男','18']])

csv模块也可以使用writerows方法依次写入多行:

- 将要写入文件的每一行存储在一个列表中
- 将每一个列表依次作为元素组成一个新序列
- 将带有嵌套列表元素的序列作为参数使用 writerows方法

CSV文件操作—练习

请编写程序根据第五章课后习题1中最后生成的通信录字典创建"通信录.csv"文件;然后在程序中查询大王的手机号、QQ号和微信号。

姓名	手机	QQ	微信号
小新	13913000001	18191220001	xx9907
小亮	13913000002	18191220002	13913000002
小刚	13913000003	18191220003	gang1004
大刘	13914000001	18191230001	liu666
大王	13914000004	18191230002	jack_w
大张	13914000003	18191230003	13914000003

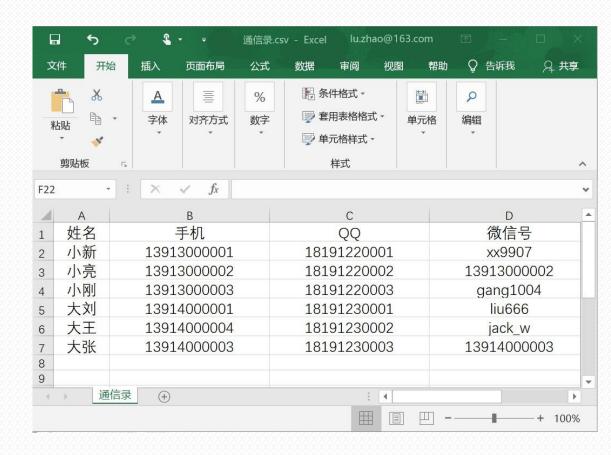


CSV文件操作一练习

将编码指定为UTF-8编码, 避免中文文本读写错误

```
import csv
with open("d:\\通信录.csv",'r',encoding='utf-8')
as txlcsv:
    reader = csv.reader(txlcsv)
    for row in reader:
        if row[0] == '大王':
        res = row
        break
```

print("大王的手机号是: ",res[1]) print("大王的QQ号是: ",res[2]) print("大王的微信号是: ",res[3])



大王的手机号是: 13914000004

大王的QQ号是: 18191230002

大王的微信号是: jack_w



异常和异常处理



异常简介

```
Traceback (most recent call last):
  File "<pyshell#32>", line 1, in <module>
    3/0
ZeroDivisionError: division by zero
Traceback (most recent call last):
  File "<pyshell#40>", line 1, in <module>
    dic[4]
KeyError: 4
Traceback (most recent call last):
  File "<pyshell#37>", line 1, in <module>
    print(y)
NameError: name 'y' is not defined
Traceback (most recent call last):
  File "<pyshell#42>", line 1, in <module>
    1st[6]
```

IndexError: list index out of range

异常

- 影响了程序的正常执行
- 导致程序非正常终止



异常名称	描述	
Exception	常规异常的基类	
ZeroDivisionError	除数为零	
IOError	输入/输出操作失败	
IndexError	序列中没有此索引(index)	
KeyError	映射中没有这个键	
NameError	未声明/初始化对象(没有属性)	

常见异常及其描述



try子句专门用来进行异常的捕捉与处理:

try:

语句

except 异常名称:

捕获异常时处理

else:

未发生异常时处理

程序执行时

- 如果try子句中发生了指定的异常,则执 行except子句部分进行异常的处理;
- 如果try子句执行未发生异常则执行 else子句部分。



【例7-1】从键盘输入a和b,求a除以b的结果并输出。

【分析】从键盘输入除数,就有可能会输入0,而除数为0是个很严重的错误,应该进行ZeroDivisionError异常的捕获和处理。

#从键盘输入a和b,求a除以b的结果并输出。 7-1.py

```
1 try:
2 a=int(input("a="))
3 b=int(input("b="))
4 c=a/b 除数b为0
5 e xcept ZeroDivisionError:
6 print("除数不能为0!")
7 el se:
8 print("c=",c)
```

```
a=10
b=0
除数不能为0!
```



【例7-2】读取并输出F:\documents\python下data2.txt文件内容,如果文件不存在则提醒用户先创建文件。

【分析】题目要求读出文件内容,需要用读模式打开,但是如果文件不存在系统会报错,所以需要进行IOError异常的捕获和处理。



```
#读取并输出F:\documents\python下data2.txt文件内容;
#如果文件不存在则提醒用户先创建文件。
   import os
               设置当前路径为文件所在目录
1
   os.chdir(r'F:\documents\python')
2
3
   try:
4
     file=open("data2.txt",'r')
5
   except IOError:
6
     print("data2.txt文件不存在,请先创建!")
   else:
8
           文件存在时, 执行正常的文件操作
     text=file.read()
9
     print("data2.txt内容:\n",text)
10
     file.close()
11
```

data2.txt文件不存在,请先创建!

data2.txt内容: 我是文件data2.txt



【例7-3】《哈姆雷特》是莎士比亚的一部经典悲剧作品。已知该作品对应的hamlet.txt文件保存在F:\documents\python目录下,请编写程序统计hamlet中出现频率最高的前10个单词,并将结果用文件名"hamlet_词频.txt"保存在同目录下。【分析】本题除了要求从指定文件获取文本外,其余都和例6-19中的要求一致。因此我们可以参考例6-19的代码,并增加文件的读写操作。



异常处理

案例7-3

例6-19中用于词 频分析的核心代码

```
#分析《哈姆雷特》中前10位的高频词
   |#编写函数对待分析文本进行预处理
   def getText(text):
     text = text.lower()
     for ch in ",.;?-:\'":
4
5
      text = text.replace(ch, " ")
6
     return text
7
   |#编写函数统计单词出现频率
8
  # text为待统计文本,topn表示取频率最高的单词个数
10 | def wordFreq(text,topn):
     words = text.split() #将文本分词
11
12
     counts = {}
13
     for word in words:
14
       counts[word] = counts.get(word,0) + 1
     excludes={'the','and','to','of','a','be','it','is','not','but'}
15
     for word in excludes:
16
17
       del(counts[word])
     items = list(counts.items())
18
19
     return items[:topn]
```



#编写主程序,调用函数

print("{:<10}{:>}".format(word, freq))

```
try:
  with open(r"F:\documents\python\hamlet.txt",'r') as file:
   text = file.read()
                                                                 针对读文件操作的异常处理
    text = getText(text)
    freqs = wordFreq(text,10)
except IOError:
  print("文件不存在,请确认!\n")
else:
 try:
    with open(r"F:\documents\python\hamlet_词频.txt",'w')as fileFreq:
                                                                   针对写文件操作的异常处理
        items = [ word + '\t' + str(freq) + '\n' for word, freq in freqs]
       fileFreq.writelines(items)
  except IOError:
    print("写入文件出错")
                                          写文件出错时,直接屏显结果
    for word, freq in freqs:
```