

PYTHON

编程基础

CSV操作一维、二维数据

概述



CSV

(Comma-Separated Values)

是一种国际通用的一维、
二维数据存储格式，其对应文件的扩展名为.csv，可使用Excel软件直接打开。

CSV文件中每行对应一个一维数据，一维数据的各数据元素之间用英文半角逗号分隔（逗号两边不需要加额外的空格）；对于缺失元素，也要保留逗号，使得元素的位置能够与实际数据对应。

概述



CSV文件中的多行形成了一个二维数据，
即一个二维数据由多个一维数据组成；
二维数据中的第一行可以是列标题，
也可以直接存储数据（即没有列标题）。

CSV文件存储示例



对于一维数据，使用CSV文件存储的结果为

90,70,95,98,65



对于二维数据，使用CSV文件存储的结果为

90,98,87

70,89,92

95,78,81

98,90,95

65,72,70

CSV文件的写操作

csv模块的writer方法可以生成一个writer对象，使用该对象可以将数据以逗号分隔的形式写入到CSV文件中。csv.writer方法的语法格式为：

```
csv.writer(csvfile)
```

其中，csvfile是一个具有write方法的对象。

如果将open函数返回的文件对象作为实参传给csvfile，则调用open函数打开文件时必须加上一个关键字参数“newline=”。

CSV文件的写操作

生成writer对象后，就可以使用csv模块的writerow和writerows方法向CSV文件中写入数据。csv.writerow和csv.writerows方法的语法格式分别为

```
writer.writerow(row)
```

```
writer.writerows(rows)
```

其中，writer是csv.writer方法返回的writer对象；row是要写入到CSV文件中的一行数据（如一维列表）；rows是要写入到CSV文件中的多行数据（如二维列表）。

CSV文件的读操作

csv模块的reader方法可以生成一个reader对象，使用该对象可以将以逗号分隔的数据从CSV文件读取出来。csv.reader方法的语法格式为：

```
csv.reader(csvfile)
```

其中，csvfile要求传入一个迭代器。

open函数返回的文件对象除了是可迭代对象，同时也是迭代器。如果将文件对象作为实参传给csvfile，则调用open函数打开文件时应加上一个关键字参数“newline=”。

返回的reader对象是一个可迭代对象，因此可以使用for循环直接遍历CSV文件中的每一行数据，每次遍历会返回一个由字符串组成的列表。

CSV文件读写示例

```
1 import csv #导入csv模块
2 data2D=[[90,98,87], #第1名学生的3门课程成绩
3 [70,89,92], #第2名学生的3门课程成绩
4 [95,78,81], #第3名学生的3门课程成绩
5 [98,90,95], #第4名学生的3门课程成绩
6 [65,72,70]] #第5名学生的3门课程成绩
7 with open('D:\\Python\\score.csv','w',newline='') as f: #打开文件
8     csvwriter=csv.writer(f) #得到writer对象
9     csvwriter.writerow(['语文','数学','英语']) #先将列标题写入CSV文件
10    csvwriter.writerows(data2D) #将二维列表中的数据写入CSV文件
```

CSV文件读写示例

```
11 ls2=[]
12 with open('D:\\Python\\score.csv','r',newline=") as f: #打开文件
13     csvreader=csv.reader(f) #得到reader对象
14     for line in csvreader: #将CSV文件中的一行数据作为列表读取到line中
15         ls2.append(line) #将当前行数据的列表添加到ls2的尾部
16 print(ls2) #输出ls2
```

```
[['语文', '数学', '英语'], ['90', '98', '87'], ['70', '89', '92'],
['95', '78', '81'], ['98', '90', '95'], ['65', '72', '70']]
```