



# Python 程序设计基础

## Python Programming



## CSV 文件

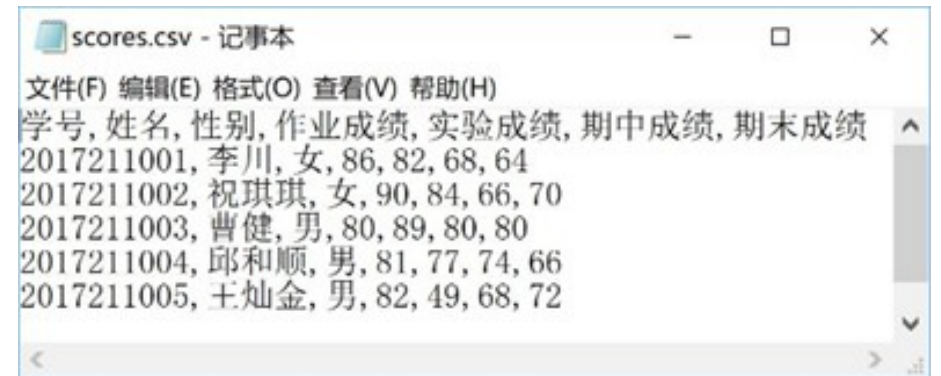
- **逗号分隔值（ Comma-Separated Values ， CSV ）是一种通用的、相对简单的文件格式，在商业和科学上得到广泛应用。 CSV 文件将表格数据存储为纯文本，很多程序可以存储、转换和处理纯文本文件，因此 CSV 文件主要用于在程序之间转移表格数据。**
- **CSV 文件是纯文本文件：**
  - **以 .csv 为文件扩展名；**
  - **以行为单位，一行数据不跨行，开头没有空行，行之间也没有空行；**
  - **每行数据通常以逗号（英文、半角）为分隔符分隔成单元格，单元格数据为空的话也要保留逗号；**
  - **可以包含或不包含列名，若包含，列名必须位于文件第一行。**

## CSV 文件

- 可以使用 Excel 打开 CSV 格式文件，也可以将 Excel 电子表格文件另存为 CSV 格式文件。在 Excel 电子表格文件中，每个单元格都有一个定义好的“类型”（数值、文本、日期、货币等），而 CSV 文件中的单元格则只是原始数据。另外，CSV 文件只能保存数据，不能保存公式。

	A	B	C	D	E	F	G
1	学号	姓名	性别	作业成绩	实验成绩	期中成绩	期末成绩
2	2017211001	李川	女	86	82	68	64
3	2017211002	祝琪琪	女	90	84	66	70
4	2017211003	曹健	男	80	89	80	80
5	2017211004	邱和顺	男	81	77	74	66
6	2017211005	王灿金	男	82	49	68	72

scores.xlsx 文件



```
scores.csv - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
学号, 姓名, 性别, 作业成绩, 实验成绩, 期中成绩, 期末成绩
2017211001, 李川, 女, 86, 82, 68, 64
2017211002, 祝琪琪, 女, 90, 84, 66, 70
2017211003, 曹健, 男, 80, 89, 80, 80
2017211004, 邱和顺, 男, 81, 77, 74, 66
2017211005, 王灿金, 男, 82, 49, 68, 72
```

scores.csv 文件



## CSV 文件

- Python 提供了一个专门处理 CSV 文件的 csv 模块。
- 要使用 csv 模块，必须先导入 csv 模块：`import csv`。

`reader(csvfile, dialect='excel', **fmtparams)`

- 从 CSV 文件读取数据。 `csvfile` 是文件对象或列表对象，如果 `csvfile` 是文件对象，则应使用 `newline=''` 打开它； `dialect` 指定 CSV 文件的格式，默认是 `excel`，即以逗号为分隔符； `fmtparams` 指定特定的格式，可以覆盖 `dialect` 中的格式。
- 返回可迭代的 `reader` 对象。 `reader` 对象其实就是由 CSV 文件的多行数据构成的，每行数据会有一个属性：`line_num`，表示行数； `next` 方法返回一个列表，表示 `reader` 对象的下一行内容。



## CSV 文件

`writer(csvfile, dialect='excel', **fmtparams)`

- **向 CSV 文件写入数据。** `csvfile` 是支持 `write` 方法的任何对象，如果 `csvfile` 是文件对象，则应使用 `newline=''` 打开它；`dialect` 指定 CSV 文件的格式，默认是 `excel`，即以逗号为分隔符；`fmtparams` 指定特定的格式，可以覆盖 `dialect` 中的格式。
- **返回 `writer` 对象，** `writer` 对象的 `writerow` 方法写入一行数据，`writerows` 方法写入多行数据。



## CSV 文件

# 读写CSV文件

```
import csv
```

```
rows = [["2017211006", "路凡义", '男', '90', '85', '78', '98'],  
        ["2017211007", "刘露", '女', '60', '52', '74', '66']]
```

```
with open("scores.csv", newline='', mode='a') as csv_out_file:
```

```
    filewriter = csv.writer(csv_out_file)
```

```
    filewriter.writerows(rows)
```

```
with open("scores.csv", newline='', mode='r') as csv_in_file:
```

```
    filereader = csv.reader(csv_in_file)
```

```
    header = next(filereader)
```

```
    print(filereader.line_num, header)
```

```
    for row in filereader:
```

```
        print(filereader.line_num, row)
```

```
1 ['学号', '姓名', '性别', '作业成绩', '实验成绩', '期中成绩', '期末成绩']
```

```
2 ['2017211001', '李川', '女', '86', '82', '68', '64']
```

```
3 ['2017211002', '祝琪琪', '女', '90', '84', '66', '70']
```

```
4 ['2017211003', '曹健', '男', '80', '89', '80', '80']
```

```
5 ['2017211004', '邱和顺', '男', '81', '77', '74', '66']
```

```
6 ['2017211005', '王灿金', '男', '82', '49', '68', '72']
```

```
7 ['2017211006', '路凡义', '男', '90', '85', '78', '98']
```

```
8 ['2017211007', '刘露', '女', '60', '52', '74', '66']
```



# CSV 文件

## ➤ csv 模块还定义了如下类：

`DictReader(csvfile, fieldnames=None, restkey=None, restval=None, dialect='excel', *args, **kwargs)`

- 从 CSV 文件读取数据，将文件中的每行数据映射到字典中，由可选的 `fieldnames` 参数指定字典的键（即文件中第一行标题字段名）。`fieldnames` 参数是一个键序列。如果省略 `fieldnames` 参数，则文件中第一行的值将作为 `fieldnames`。
- `DictReader` 类创建了一个类似 reader 的对象，对象的 `fieldnames` 属性返回一个由标题字段名构成的列表。



## CSV 文件

`DictWriter(csvfile, fieldnames, restval='', extrasaction='raise', dialect='excel', *args, **kwargs)`

- **向 CSV 文件写入数据。将字典中的数据映射到输出文件中的每行数据。** `fieldnames` 参数指定字典的键（即文件中第一行标题字段名）。`fieldnames` 参数是一个键序列，用于标识传递给 `writerow` 方法的字典中的值写入到文件中的顺序。与 `DictReader` 类不同，`DictWriter` 类的 `fieldnames` 参数不是可选的。
- **`DictWriter` 类创建了一个类似 `writer` 的对象，对象的 `writeheader` 方法将标题字段名写入文件。**





## CSV 文件

# 读写CSV文件，使用DictReader和DictWriter类

```
import csv
```

```
with open("scores.csv", newline='', mode='r') as csv_in_file:
```

```
    filereader = csv.DictReader(csv_in_file)
```

```
    header = filereader.fieldnames # 获取标题字段名
```

```
    header.append("总分") # 添加一个新的标题字段名
```

```
with open("new_scores.csv", newline='', mode='w') as csv_out_file:
```

```
    filewriter = csv.DictWriter(csv_out_file, header)
```

```
    filewriter.writeheader() # 写入标题字段名
```

```
    for row in filereader:
```

```
        total = (eval(row["作业成绩"]) + eval(row["期中成绩"])) * 0.2 + \
                (eval(row["实验成绩"]) + eval(row["期末成绩"])) * 0.3
```

```
        row["总分"] = int(total)
```

```
        filewriter.writerow(row)
```



变量 total 中存放按比例计算的总分，取整并添加到字典 row 中，writerow 方法将字典 row 写入文件。

new\_scores.csv 文件