

自动排名系统(一)





1. 探索新知

姓名	地标挑战赛成绩	历史挑战赛成绩	生活常识挑战赛成绩
小李	88	86	83
小王	77	90	95
小公主	99	66	86
阿花	66	97	82
张三	69	79	81
李四	75	82	88
王五	91	86	73
赵六	76	79	93

计算每个学生的总分，按照分数从高到低排序，最后找到代表班级参赛的学生。输出如下所示：

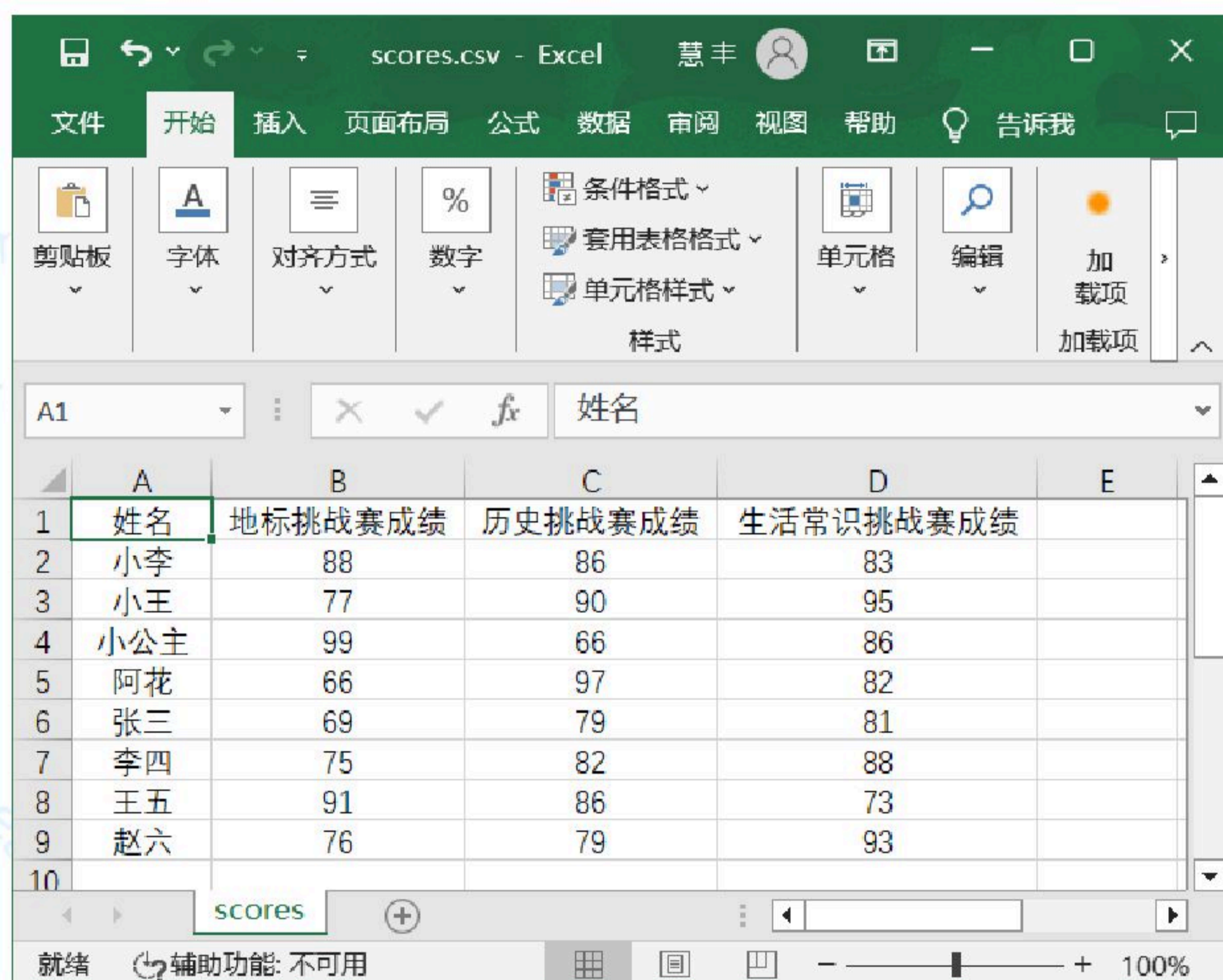
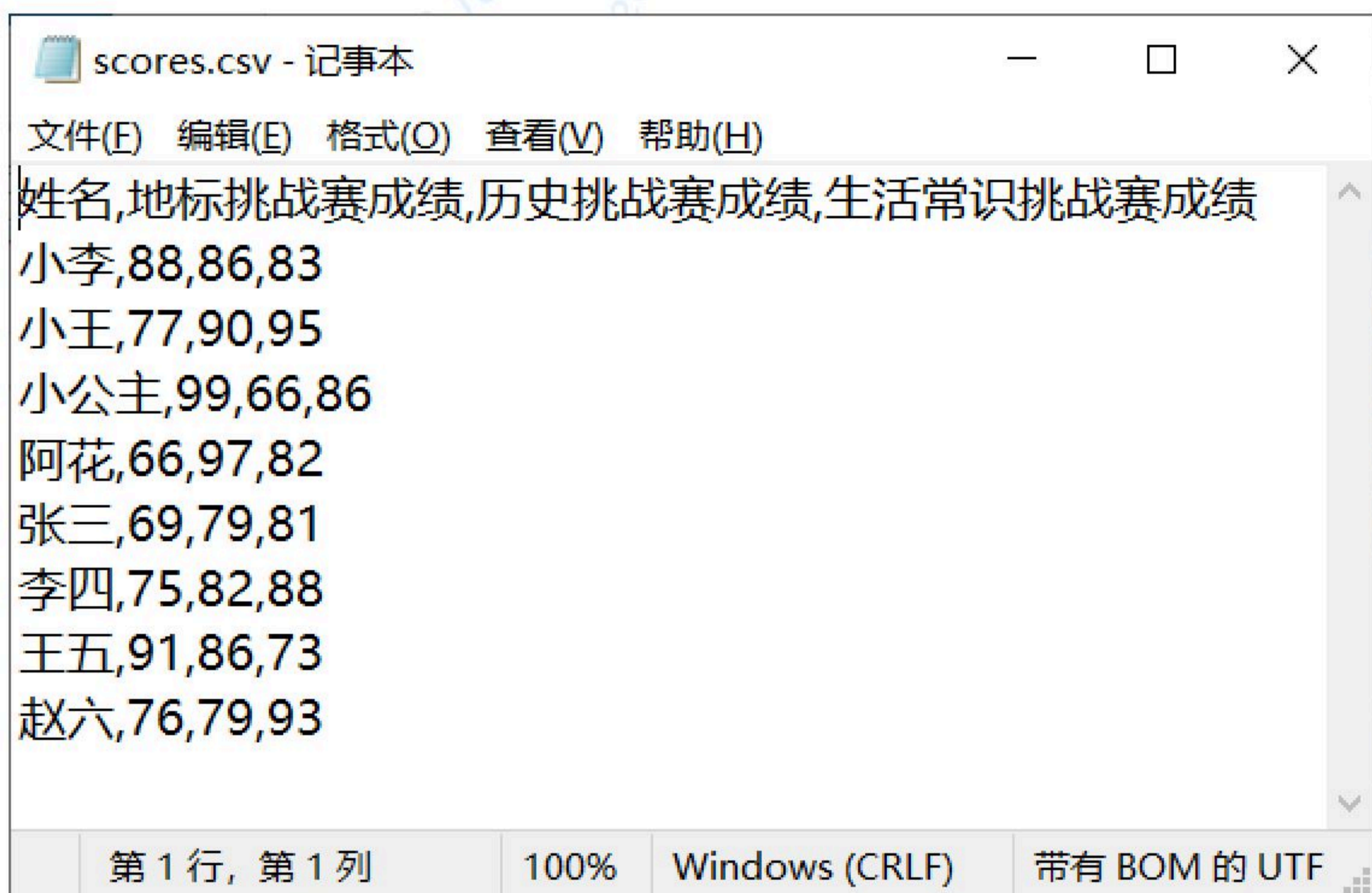
```
控制台
排名 姓名 总分
1 小王 262
2 小李 257
3 小公主 251
4 王五 250
5 赵六 248
6 阿花 245
7 李四 245
8 张三 229
代表班级参加知识竞赛的是：小王
```

1. 了解CSV文件
2. 打开CSV文件
3. 读取CSV文件中的数据
4. 提取学生的成绩并计算总分
5. 存储每个学生的信息
6. 按总分对学生进行排序
7. 打印排名结果
8. 找出分数最高的学生

1.1

了解CSV文件

CSV文件是一种保存数据的文件，它看起来就像一个表格。CSV文件以纯文本形式存储数据，每行表示一条记录，不同字段之间使用逗号进行分隔。



我们要做什么？

- 我们要读取一个包含学生挑战赛成绩的CSV文件
- 我们的目标是计算每个学生的总分，然后按照分数从高到低排名，并找出分数最高的学生

```
import csv
```

接着，咱们定义一个函数，功能是读取CSV文件并计算出总分进行排序



```
def process(filename):
    <...函数体...>

process('scores.csv')
```

调用函数时，填入需要打开的文件名即可

【process】：过程
【filename】：文件名

1.1

了解CSV文件

任务一新增代码如下：

```
# 导入CSV库
import csv
def process(filename):
    <...函数体...>
process('scores.csv')
```

1.2

打开CSV文件



先来实现读取CSV文件。
上节课我们已经学会了打开文件的方法
(只读模式、只写模式)

```
my_file = open(filename, 'r')
```

```
my_file = open(filename, 'w')
```

如果用上述方法打开文件，使用完成后需要用`close()`关闭文件

Python也有一个特殊的魔法，可以帮助我们自动关闭文件，那就是 `with` 语句。当我们使用 `with` 语句时，一旦我们完成了文件操作，Python会自动帮我们关闭文件

```
with open('magic_paper.txt', 'r') as my_file:
    content = my_file.read()
    print(content)
```

1.2

打开CSV文件

as 的作用就是给 `open()` 返回的文件对象起一个名字 (`my_file`)，以便在后续的代码中使用这个名字来引用该文件对象



接下来我们试着用with方法打开CSV文件

```
with open(filename, 'r') as csvfile:
```

任务二新增代码如下：

```
# 函数体  
with open(filename, 'r') as csvfile:
```

1.3

读取文件数据

CSV库提供了两种方式来读取CSV文件：使用`csv.reader()`函数逐行读取和使用`csv.DictReader()`函数按字典方式读取。

1. 使用`csv.reader()`函数读取CSV文件

`csv.reader()`函数是最基本的读取CSV文件的方法。它返回一个迭代器，可以用来逐行读取CSV文件。

以下是读取名为 `example.csv` 的CSV文件的示例代码：

```
import csv
with open('example.csv', 'r') as csvfile:
    reader = csv.reader(csvfile)
    for row in reader:
        print(row)
```

【row】：
行

代码执行结果如下：

```
['Name', 'Age', 'City']
['May', '15', 'Tianjin']
['John', '20', 'Hangzhou']
```

拓展：迭代器

迭代器在Python中是一个非常有用的工具。我们可以把它想象成一个“魔法盒子”，这个盒子里有很多东西，但是我们一次只能拿出一个。每当我们想要下一个东西时，我们就告诉这个“魔法盒子”给我们下一个，直到盒子里的东西都被拿完。

迭代器是一个特殊的对象，通常用于遍历一个可迭代对象的元素，如列表、元组或字典等。但**迭代器只能逐个返回可迭代对象中的元素**，而且不会占用额外的内存来存储这些元素。这使得迭代器在处理大量数据时非常高效。

1.3

读取文件数据

2. 使用csv.DictReader()函数读取CSV文件

如果CSV文件的第一行是字段名，我们可以使用csv.DictReader()函数按字典方式读取CSV文件。

它每次读取一行数据并将其转换为字典，键值对的形式为字段名和对应的值。

	A	B	C	D	E
1	姓名	地标挑战赛成绩	历史挑战赛成绩	生活常识挑战赛成绩	
2	小李	88	86	83	
3	小王	77	90	95	
4	小公主	99	66	86	
5	阿花	66	97	82	
6	张三	69	79	81	
7	李四	75	82	88	
8	王五	91	86	73	
9	赵六	76	79	93	
10					

字段名

以下是读取名为 example.csv 的CSV文件的示例代码：

```
import csv
with open('example.csv', 'r') as csvfile:
    reader = csv.DictReader(csvfile)
    for row in reader:
        print(row)
```

代码执行结果如下：

```
{'Name': 'May', 'Age': '15', 'City': 'Tianjin'}
{'Name': 'John', 'Age': '20', 'City': 'Hangzhou'}
```

我们需要挨个计算每个学生的平均分。

如果采用第一种方法，第一个元素是字段名组成的列表，没有需要的数据；

而如果采用第二种方法，每个字典都是学生的相关数据。

所以，我们本节课选择第二种方法读取csv文件。



成功打开csv文件后，我们需要遍历迭代器reader提取出学生姓名和成绩，并转换为整数

```
import csv
def process(filename):
    with open(filename, 'r') as csvfile:
        reader = csv.DictReader(csvfile)
        for row in reader:
            print(row)
```

存储每行数据的字典



我们试着自己完成文件的初步读取

任务三新增代码如下：

```
# 函数体
with open(filename, 'r') as csvfile:
    reader = csv.DictReader(csvfile) # 迭代器reader
    for row in reader:
        print(row) # print语句仅测试使用，后面可以删掉
```

1.4

计算总分

将row逐行输出，如下所示：

```
控制台
{'姓名': '小李', '地标挑战赛成绩': '88', '历史挑战赛成绩': '86', '生活常识挑战赛成绩': '83'}
{'姓名': '小王', '地标挑战赛成绩': '77', '历史挑战赛成绩': '90', '生活常识挑战赛成绩': '95'}
{'姓名': '小公主', '地标挑战赛成绩': '99', '历史挑战赛成绩': '66', '生活常识挑战赛成绩': '86'}
{'姓名': '阿花', '地标挑战赛成绩': '66', '历史挑战赛成绩': '97', '生活常识挑战赛成绩': '82'}
{'姓名': '张三', '地标挑战赛成绩': '69', '历史挑战赛成绩': '79', '生活常识挑战赛成绩': '81'}
{'姓名': '李四', '地标挑战赛成绩': '75', '历史挑战赛成绩': '82', '生活常识挑战赛成绩': '88'}
{'姓名': '王五', '地标挑战赛成绩': '91', '历史挑战赛成绩': '86', '生活常识挑战赛成绩': '73'}
{'姓名': '赵六', '地标挑战赛成绩': '76', '历史挑战赛成绩': '79', '生活常识挑战赛成绩': '93'}
```



建立变量，分别存储学生姓名、各项比赛成绩，注意成绩要转换为整数~

【history】：历史
【life】：生活
【score】：分数

```
for row in reader:
    # 提取学生姓名和成绩，并转换成整数
    name = row['姓名']
    landmark_score = int(row['地标挑战赛成绩'])
    history_score = int(row['历史挑战赛成绩'])
    life_score = int(row['生活常识挑战赛成绩'])
```



如何计算三个比赛的总分？
三个比赛成绩加起来就好了呗。
So easy!

```
total_score = landmark_score + history_score + life_score
print(total_score) # print语句仅测试使用，后面可以删掉
```

【total】：全部的

任务四新增代码如下：

```
# 函数体
with open(filename, 'r') as csvfile:
    reader = csv.DictReader(csvfile)
    for row in reader:
        # 提取学生姓名和成绩，并转换成整数
        name = row['姓名']
        landmark_score = int(row['地标挑战赛成绩'])
        history_score = int(row['历史挑战赛成绩'])
        life_score = int(row['生活常识挑战赛成绩'])
        # 计算总分
        total_score = landmark_score + history_score + life_score
        print(total_score) # print语句仅测试使用，后面可以删掉
```

完整代码

```
import csv

# 读取CSV文件并计算总分
def process(filename):
    with open(filename, 'r') as csvfile:
        reader = csv.DictReader(csvfile)
        for row in reader:
            # 提取学生姓名和成绩, 并转换成整数
            name = row['姓名']
            landmark_score = int(row['地标挑战赛成绩'])
            history_score = int(row['历史挑战赛成绩'])
            life_score = int(row['生活常识挑战赛成绩'])
            # 计算总分
            total_score = landmark_score + history_score + life_score
            print(total_score) # print语句仅测试使用, 后面可以删掉

process('scores.csv')
```



2. 强化练习

1. `with open('my_file.csv', 'r') as csvfile`: 这行代码的作用是什么? ()

- A. 打开一个名为 `csvfile` 的CSV文件以写入数据
- B. 打开一个名为 `my_file` 的CSV文件以读取数据, 并确保文件在操作完成后正确关闭
- C. 打开一个名为 `my_file` 的CSV文件以读取数据, 操作完成后需要`close()`函数关闭文件
- D. 打开一个名为 `my_file` 的文本文件以读取数据

2. 在使用 `csv.DictReader` 读取CSV文件时, 以下哪个描述是正确的? ()

- A. 它直接返回CSV文件的所有内容
- B. 它返回的是一个列表
- C. 它返回的是一个字典
- D. 它返回的是一个迭代器, 每次读取一行数据并将其转换为字典

3. 要读取下图“书目.csv”文件的全部内容, 小明编写了后面4行代码。请问, 横线①处, 应该填写哪种打开模式? ()

- A. "w"
- B. "a"
- C. "r"
- D. "a+"

书目.csv - 记事本				
文件(F)	编辑(E)	格式(O)	查看(V)	帮助(H)
书名	价格	库存量		
西游记	40	1000		
红楼梦	80	1000		
聊斋志异	60	1000		
水浒传	50	1000		
三国志	110	1000		

```
f = open("书目.csv", ① )
a = f.read()
print(a)
f.close
```



2. 强化练习

4. 下列哪个说法不正确? ()

- A. 字典是通过 key (键) 来进行访问的, 跟列表、元组不同
- B. 字典没有下标, 跟 value 对应的就只有 key (键)
- C. Python 中, 可以通过重新给key赋值的方式来修改已有的value, 不能通过新增 key 的方式来添加新的键值对
- D. 用 Python 内置函数是可以删除字典当中的元素的, 整个字典也可以被清空或者删除

5. 如果CSV文件中有一列名为“年龄”, 要提取这一列的数据, 横线处应该使用以下哪个代码片段? ()

```
import csv
with open('example.csv', 'r') as csvfile:
    reader = csv.DictReader(csvfile)
    for row in reader:
        _____
```

- A. age = row['姓名']
- B. age = row.年龄
- C. age = row['年龄']
- D. age = row.get('年龄', 0)

3. 术语箱

filename	文件名	life	生活
row	行	total	全部的
history	历史		

4. 课后挑战

谁跑得最快

学校举办了一场小型运动会，记录了学生们的百米赛跑成绩。这些成绩保存在一个CSV文件中，你需要编写一个简单的程序来打印出学生的跑步成绩，格式如下：

××的成绩是××秒

