

# 自动排名系统(二)





## 1. 探索新知

### 1.1

### 存储学生信息

上节课咱们已经逐行获取到了学生的姓名和总分，临时存在变量`name`和`total_score`中。这个任务中，咱们要存储每个学生的信息，便于后续排序、打印排名



学生的信息有两个，分别是姓名和总分，以字典的形式进行存储，将'姓名'、'总分'作为键，获取到的数据作为值

```
stu_info = {'姓名': name, '总分': total_score}
```

【information】：  
信息



我们将存储每个学生信息的字典，汇总到一个列表中，再进行排序。那么如何将这字典一一放入列表中呢？

1. 首先建立一个空列表

```
students = []
```

2. 逐个加入学生信息 `stu_info`

```
students.append(stu_info)
```

## 1.1

## 存储学生信息

任务一新增代码如下：

```
def process(filename):  
    students = []  
    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  
            # 计算总分  
            # 创建一个字典存储学生信息和总分  
            stu_info = {'姓名': name, '总分': total_score}  
            # 将字典添加到列表中  
            students.append(stu_info)
```

## 1.2

## 按总分对学生进行排序

对列表内元素进行排序的两种方法：

### 1. 使用 `sort()` 方法

`sort()` 方法会修改原来的列表，使其元素按照一定的顺序排列。

默认情况下，`sort()` 方法按照升序排列列表元素

```
lst = [20, 40, 10, 98, 54]  
lst.sort()  
print(lst) # 输出: [10, 20, 40, 54, 98]
```

## 1.2

## 按总分对学生进行排序

【reverse】：  
翻转

如果想要按照降序排列，可以传递参数 `reverse=True` 给 `sort()` 方法：

```
lst = [20, 40, 10, 98, 54]
lst.sort(reverse=True)
print(lst) # 输出: [98, 54, 40, 20, 10]
```

2. 使用 `sorted()` 函数

`sorted()` 函数会返回一个新的已排序列表，原列表不会被修改

```
lst = [20, 40, 10, 98, 54]
sorted_lst = sorted(lst)
print(sorted_lst) # 输出: [10, 20, 40, 54, 98]
print(lst) # 输出: [20, 40, 10, 98, 54] (原列表未改变)
```

如果想要按照降序排列，可以传递参数 `reverse=True` 给 `sorted()` 函数：

```
lst = [20, 40, 10, 98, 54]
sorted_lst = sorted(lst, reverse=True)
print(sorted_lst) # 输出: [98, 54, 40, 20, 10]
print(lst) # 输出: [20, 40, 10, 98, 54] (原列表未改变)
```

先输出 `students` 列表，如下：

```
[{'学生信息1': '小李', '总分': 257},
 {'学生信息2': '小王', '总分': 262},
 {'学生信息3': '公主', '总分': 251},
 {'学生信息4': '阿花', '总分': 245},
 {'学生信息5': '张三', '总分': 229},
 {'学生信息6': '李四', '总分': 245},
 {'学生信息7': '王五', '总分': 250},
 {'学生信息8': '赵六', '总分': 248}]
```

根据总分对列表内元素（学生信息字典）进行排序



那么问题来了，  
如何获取各字典中的总分呢？

```
def ave(s): # s是列表中的每一项
    return s['总分']
```



那么，根据总分对列表元素进行降  
序排列的代码该如何书写呢？

```
def ave(s): # s是列表中的每一项
    return s['总分']
students.sort(key=ave, reverse=True)
```

### 拓展：匿名函数

在Python中，函数是一种强大的工具，用于将一组指令打包成可重复使用的块。然而在某些情况下，我们只需要一个简单的函数来执行一些操作，而不希望定义一个完整的函数。

这就是需要用到Python中的匿名函数，也被称为lambda函数。

## 拓展：匿名函数

lambda函数是一种匿名函数，它允许我们在不使用def关键字定义函数的情况下创建一个简单的函数。lambda函数的基本语法如下：

```
lambda arguments: expression
```

lambda : 是Python的关键字，表示这是一个匿名函数

arguments : 是函数的参数，它们可以是一个或多个，用逗号分隔

expression : 是一个单一的表达式，lambda函数将返回这个表达式的结果

带参数的简单函数：

```
def add(x, y):  
    return x + y  
print(add(3,5))
```

当您只需要一个简单的、仅在一处使用的函数时，定义一个命名函数可能显得繁琐。lambda函数可以提供一种更简洁的方式来表示这样的操作

匿名函数：

```
add = lambda x, y: x + y  
print(add(3, 5)) # 输出: 8
```



试着将下面带参数的简单函数改写成匿名函数

```
def ave(s): # s是列表中的每一项  
    return s['总分']  
students.sort(key=ave, reverse=True)
```

```
students.sort(key=lambda s: s['总分'], reverse=True)
```

## 1.2

### 按总分对学生进行排序



最后我们将排好序的列表进行返回

```
def process(filename):  
    <.....>  
    return students
```

任务二新增代码如下：

# 获取各字典中的总分

```
def ave(s): # s是列表中的每一项
```

```
    return s['总分']
```

```
def process(filename):
```

```
    students = []
```

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

```
        <.....>
```

# 根据总分对学生进行排序（降序）

```
    students.sort(key=ave, reverse=True)
```

```
    return students
```

## 1.3

### 打印排名结果

咱们创建一个函数，起名为print\_rank，专门打印排名的结果



【rank】：  
列队，排列

```
def print_rank(students):
```

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

# 1.3

## 打印排名结果

观察整个输出，打印的是一个简单的表格，每一列的文本都尽量左对齐



如何输出这样的表格形式呢？

### 制表符 '\t'

在Python中，制表符通常使用 \t 来表示，它是一个特殊的转义字符。当你在字符串中放置一个 \t 时，它会被解释为一个制表符（tab字符），在输出中创建一个水平的空白间隔，通常用于使文本在垂直方向上对齐。

```
print("Name\tAge\tCountry")
print("Alice\t28\tUSA")
print("Bob\t23\tCanada")
print("Charlie\t30\tUK")
```

控制台		
Name	Age	Country
Alice	28	USA
Bob	23	Canada
Charlie	30	UK

控制台		
排名	姓名	总分
1	小土	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

咱们先试着使用制表符输出第一行的内容

```
print("排名\t姓名\t平均分")
```

# 1.3

## 打印排名结果

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

将列表中的元素从1开始排名，并输出排名、姓名和总分

**【enumerate】：**  
枚举，列举

enumerate()函数

在Python中，enumerate()是一个内置函数，它能在遍历一个可迭代对象（如列表、元组或字符串）时跟踪当前元素的索引。用于将可遍历的数据对象组合为一个索引序列。通常，当需要知道每个元素的索引位置时，我们会使用enumerate()

可以遍历的对象  
(如列表、元组等)

基本用法如下：

```
for index, value in enumerate(iterable, start=0):
    # 循环体
```

可选参数，指定索引的起始值，默认为0

```
fruits = ['apple', 'banana', 'cherry']
for index, fruit in enumerate(fruits):
    print(index, fruit)
```

控制台
0 apple
1 banana
2 cherry

```
fruits = ['apple', 'banana', 'cherry']
for index, fruit in enumerate(fruits, start=1):
    print(index, fruit)
```

控制台
1 apple
2 banana
3 cherry

```
for i, student in enumerate(students, start=1):
    print(f"{i}\t{student['姓名']}\t{student['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

## 拓展

在 f-string 中，如果键名包含特殊字符或者关键字，你可以在外层使用双引号，内部使用单引号，反之亦然。

例如：

```
print(f'{i}\t{student['姓名']}\t{student['总分']}')
```



```
print(f"{i}\t{student['姓名']}\t{student['总分']}")
```



```
print(f'{i}\t{student["姓名"]}\t{student["总分"]}')
```

任务三新增代码如下：

```
# 打印排名结果
```


```
def print_rank(students):
    print("排名\t姓名\t总分")
    for i, student in enumerate(students, start=1):
        print(f"{i}\t{student['姓名']}\t{student['总分']}")
```

## 1.4

## 找出分数最高的学生



现在咱们准备好了计算总分并排序的函数以及打印排序的函数，接下来就传入CSV文件开始自动排名啦！

 scores.csv

```
# 传入csv文件，得到排序后的列表
students = process('scores.csv')
# 打印排名结果
print_rank(students)
```

1. 如何获取第一名学生信息？

```
students[0]
```

```
{'姓名': '小王', '总分': 262}
```

2. 如何获取第一名学生姓名？

```
students[0]['姓名']
```

```
print(f"代表班级参加知识竞赛的是: {students[0]['姓名']}")
```

任务四新增代码如下：

```
# 主程序
students = process('scores.csv') # 指定CSV文件
print_rank(students)
print(f"代表班级参加知识竞赛的是: {students[0]['姓名']}")
```

最后我们将打印出，代表班级参加知识竞赛的同学姓名



## 完整代码

```
import csv
# 获取各字典中的总分
def ave(s): # s是列表中的每一项
    return s['总分']
# 读取CSV文件并计算总分进行排序
def process(filename):
    students = []
    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
            # 创建一个字典存储学生信息和总分
            stu_info = {'姓名': name, '总分': total_score}
            # 将字典添加到列表中
            students.append(stu_info)
            # 根据总分对学生进行排序 (降序)
        students.sort(key=ave, reverse=True)
    return students
# 打印排名结果
def print_rank(students):
    print("排名\t姓名\t总分")
    for i, student in enumerate(students, start=1):
        print(f"{i}\t{student['姓名']}\t{student['总分']}")
# 主程序
students = process('scores.csv') # 指定CSV文件
print_rank(students)
print(f"代表班级参加知识竞赛的是: {students[0]['姓名']}")
```



## 2. 强化练习

1. 小明运行下面的程序，运行结果正确的是？（ ）

A. (0, 1, 5, 6, 7, 9, 15)

B. (15, 9, 7, 6, 5, 1, 0)

C. [0, 1, 5, 6, 7, 9, 15]

D. [15, 9, 7, 6, 5, 1, 0]

```
h = [6, 15, 1, 0, 7, 9, 5]
p = sorted(h)
print(p)
```

2. 下面程序执行结果是？（ ）

A. [(0, '春'), (1, '夏'), (2, '秋'), (3, '冬')]

B. [[0, '春'], [1, '夏'], [2, '秋'], [3, '冬']]

C. [(1, '春'), (2, '夏'), (3, '秋'), (4, '冬')]

D. [[1, '春'], [2, '夏'], [3, '秋'], [4, '冬']]

```
a=['春','夏','秋','冬']
c=list(enumerate(a))
print(c)
```

3. 若有代码如下，则显示的结果是？（ ）

```
name=["李明", "王宁", "朱倩"]
for a,i in enumerate(name):
    print(a,i)
```

A. 0 李明  
1 王宁  
2 朱倩

B. 李明  
王宁  
朱倩

C. 李明, 王宁, 朱倩

D. 1 李明  
2 王宁  
3 朱倩



## 2. 强化练习

4. (拓展)小方编写计算长方形面积的匿名函数,下列哪一个语句是正确的?  
( )

A. `rst = lambda a,b : a*b`

B. `lambda a,b : a*b`

C. `lambda a,b,a*b`

D. `rst = lambda a,b,a*b`

5. 在Python中, `\t` 代表什么? ( )

A. 一个空格

B. 一个换行符

C. 一个制表符 (Tab)

D. 一个字符串结束符

### 3. 术语箱

sort 排序

reverse 翻转

enumerate 枚举, 列举

rank 列队, 排列

arguments 参数

expression 表达式

### 4. 课后挑战

在一所魔法学校，学生们刚刚完成了一场重要的魔法考试。每位学生都获得了不同的分数，现在需要根据分数对学生进行排名，并颁发奖励给分数最高的三位学生。

#### 任务要求：

1. 已知包含学生姓名和对应分数的列表为：  
`students = [ {'姓名': '赫敏', '分数': 92}, {'姓名': '哈利', '分数': 95}, {'姓名': '罗恩', '分数': 88} ]`
2. 使用列表排序功能，根据分数对所有学生进行降序排序
3. 使用枚举函数打印出排名前三的学生，结构为“第×名：××”

