

实例 2：商品价格区间设置与排序

在网上购物时，面对琳琅满目的商品，我们应该如何快速选择适合自己的商品呢？为了能够让用户快速地定位到适合自己的商品，每个电商购物平台都提供价格排序与设置价格区间功能。假设现在某平台共有 10 件商品，每件商品对应的价格如表 1 所示。

表 1 商品价格

| 序号 | 价格 |
|----|------|
| 1 | 399 |
| 2 | 4369 |
| 3 | 539 |
| 4 | 288 |
| 5 | 109 |
| 6 | 749 |
| 7 | 235 |
| 8 | 190 |
| 9 | 99 |
| 10 | 1000 |

用户根据提示“请输入最大价格:”和“请输入最小价格:”分别输入最大价格和最小价格，选定符合自己需求的价格区间，并按照提示“1.价格降序排序（换行）2.价格升序排序（换行）请选择排序方式:”输入相应的序号，程序根据用户输入将排序后的价格区间内的价格全部输出。

本实例要求编写程序，实现以上描述的设置价格区间和价格排序的功能。

实例目标

- 掌握列表的排序，能分别采用降序和升序两种方式排列列表元素
- 掌握列表的遍历

实例分析

本实例的目的是从 10 个商品价格中选出位于价格区间的部分商品价格，进行排序并输出，这个过程可拆分为两步：

（1）按用户设置的价格区间选择部分商品价格。根据用户输入的最大价格和最小价格确定价格区间，依次取出 10 个商品价格进行比较：若某商品价格位于此区间中，则将其进行保存。待所有的商品价格均做完比较后，此时得到位于价格区间内的部分商品价格。

（2）按用户选择的排序方式排列商品价格。根据用户输入的排序方式，将上一步骤中

得到的部分商品价格重新排列。

由于上述过程中涉及到遍历、排序、动态存储等操作，这里使用列表来存储商品价格。

代码实现

本实例的具体实现代码如下所示：

```
price_li = [399, 4369, 539, 288, 109, 749, 235, 190, 99, 1000]
section_li = []

max_section = int(input("请输入最大价格:"))
min_section = int(input("请输入最小价格:"))

for i in price_li:
    if min_section <= i <= max_section:
        section_li.append(i)

print("1.价格降序排序")
print("2.价格升序排序")

choice_num = int(input("请选择排序方式:"))

if choice_num == 1:
    section_li.sort(reverse=True)
else:
    section_li.sort()

print(section_li)
```

以上代码定义了包含 10 个元素的列表 `price_li` 和空列表 `section_li`，之后定义了变量 `max_section` 和 `min_section` 接收用户输入的两个数值，并将遍历列表 `price_li`，将其中大于等于 `min_section` 或小于等于 `max_section` 的元素添加到列表 `section_li` 中，然后再次接收了用户选择的排序方式，根据排序方式排列列表 `section_li` 中的元素：若用户输入“1”，说明选择降序排列，即按从大到小的顺序排列列表元素；若用户输入“2”，说明选择升序排列，即按从小到大的顺序排列列表元素，最后将排序后的列表 `section_li` 进行输出。

代码测试

运行程序，在控制台分别输入最大价格为 1000、最小价格为 300、排序方式为 1 之后的输出结果如下所示：

```
请输入最大价格:1000
请输入最小价格:300
1.价格降序排序
2.价格升序排序
请选择排序方式:1
[1000, 749, 539, 399]
```

再次运行程序，在控制台分别输入最大价格为 1000、最小价格为 300、排序方式为 2

之后的结果如下所示：

请输入最大价格:1000

请输入最小价格:300

1. 价格降序排序

2. 价格升序排序

请选择排序方式:2

[399, 539, 749, 1000]