江苏大学京江学院
JIANGSU UNIVERSITY JINGJIANG COLLEGE

Python编程与科学计算
练习报告

题　　目：	第三课　课后练习
授课教师：	王洪金
姓　　名：	马云骥
学　　号：	4211153047
专　　业：	软件工程
班　　级：	J软件(嵌入)(专转本)2102
日　　期：	2024.03.29

第三课　课后练习

1　练习1：问候语

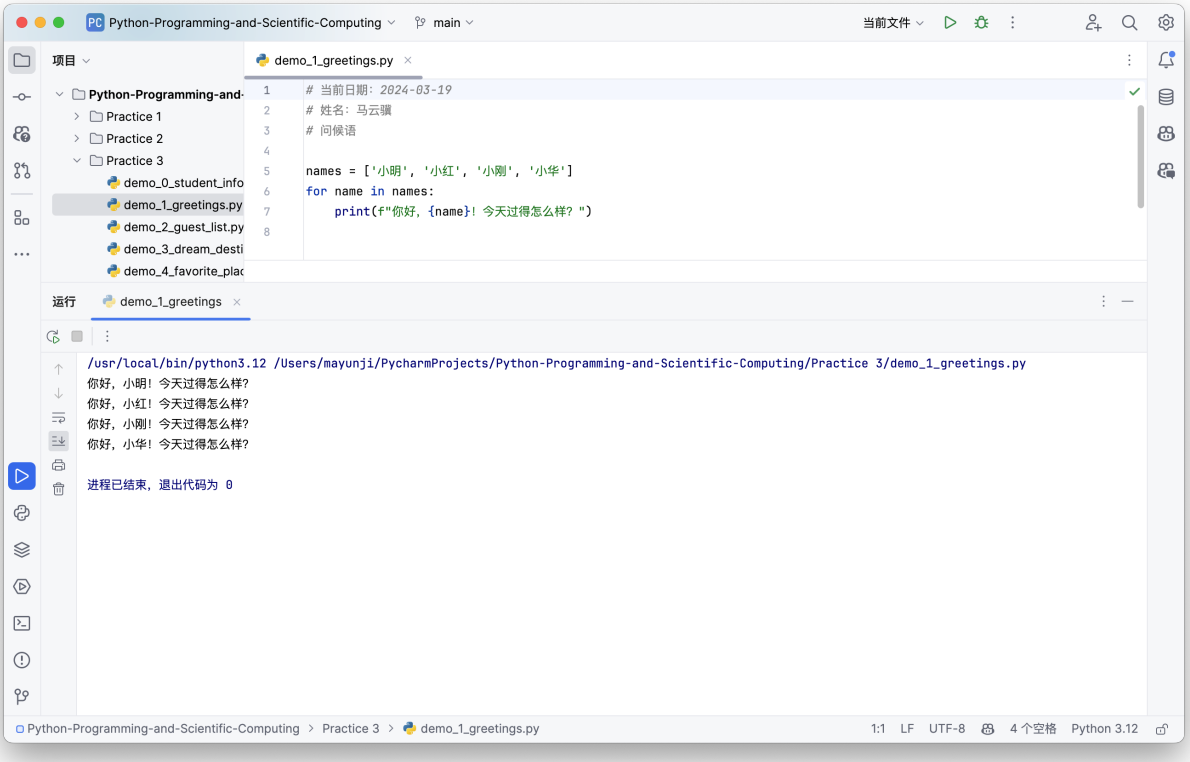
1.1　题目叙述

将一些朋友的姓名存储在一个列表中，并将其命名为 `names`。依次访问该列表中的每个元素，为每人打印一条消息。每条消息都包含相同的问候语，但抬头为相应朋友的姓名。

1.2　程序代码

```python
# 当前日期：2024-03-19
# 姓名：马云骥
# 问候语

names = ['小明', '小红', '小刚', '小华']
for name in names:
    print(f"你好，{name}！今天过得怎么样？")

```

1.3　结果输出



2 练习2：嘉宾名单

2.1 题目叙述

1. **嘉宾名单**：如果你可以邀请任何人一起共进晚餐(无论是在世的还是故去的)，你会邀请哪些人？请创建一个列表，其中包含至少三个你想邀请的人，然后使用这个列表打印消息，邀请这些人来与你共进晚餐。

2. **修改嘉宾名单**：你刚得知有位嘉宾无法赴约，因此需要另外邀请一位嘉宾。在程序末尾添加一条print 语句，指出哪位嘉宾无法赴约。修改嘉宾名单，将无法赴约的嘉宾的姓名替换为新邀请的嘉宾的姓名。再次打印一系列消息，向名单中的每位嘉宾发出邀请。

3. **添加嘉宾**：你刚找到了一个更大的餐桌，可容纳更多的嘉宾。请想想你还想邀请哪三位嘉宾。在程序末尾添加一条print 语句，指出你找到了一个更大的餐桌。

   - 使用 `insert()` 将一位新嘉宾添加到名单开头。
   - 使用 `insert()` 将另一位新嘉宾添加到名单中间。
   - 使用 `append()` 将最后一位新嘉宾添加到名单末尾。

   打印一系列消息，向名单中的每位嘉宾发出邀请。

4. **缩减名单**：你刚得知新购买的餐桌无法及时送达，因此只能邀请两位嘉宾。在程序末尾添加一行代码，打印一条你只能邀请两位嘉宾共进晚餐的消息。

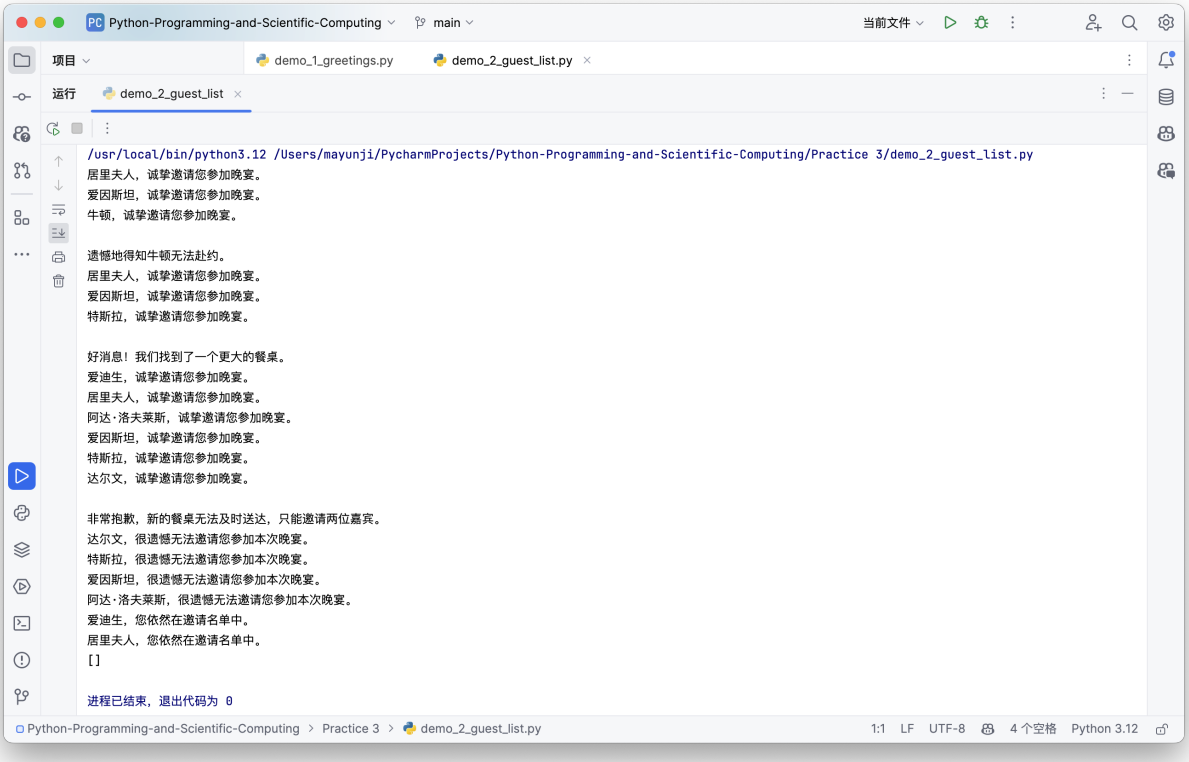
   - 使用 `pop()` 不断地删除名单中的嘉宾，直到只有两位嘉宾为止。每次从名单中弹出一位嘉宾时，都打印一条消息，让该嘉宾知悉你很抱歉，无法邀请他来共进晚餐。
   - 对于余下两位嘉宾中的每一位，都打印一条消息，指出他依然在受邀人之列。
   - 使用 `del` 将最后两位嘉宾从名单中删除，让名单变成空的。打印该名单，核实程序结束时名单确实是空的。

2.2 程序代码

```python
# 当前日期：2024-03-19
# 姓名：马云骥
# 嘉宾名单

# 初始名单
guests = ['居里夫人', '爱因斯坦', '牛顿']
for guest in guests:
    print(f"{guest}, 诚挚邀请您参加晚宴。")

# 修改嘉宾名单
print(f"\n遗憾地得知{guests[2]}无法赴约。")
guests[2] = '特斯拉'
for guest in guests:
    print(f"{guest}, 诚挚邀请您参加晚宴。")

# 添加嘉宾
print("\n好消息！我们找到了一个更大的餐桌。")
guests.insert(0, '爱迪生')
guests.insert(2, '阿达·洛夫莱斯')
guests.append('达尔文')
for guest in guests:

```

```
    print(f"{guest}, 诚挚邀请您参加晚宴。")

# 缩减名单
print("\n非常抱歉，新的餐桌无法及时送达，只能邀请两位嘉宾。")
while len(guests) > 2:
    removed_guest = guests.pop()
    print(f"{removed_guest}, 很遗憾无法邀请您参加本次晚宴。")
for guest in guests:
    print(f"{guest}, 您依然在邀请名单中。")

# 删除剩余嘉宾
del guests[0:2]
print(guests)

```

2.3 结果输出



3 练习3：放眼世界

3.1 题目叙述

想出至少5个你渴望去旅游的地方。将这些地方存储在一个列表中，并确保其中的元素不是按字母顺序排列的。

1. 使用 `sorted()` 按字母顺序打印这个列表，同时不要修改它。再次打印该列表，核实排列顺序未变。
2. 使用 `sorted()` 按与字母顺序相反的顺序打印这个列表，同时不要修改它。再次打印该列表，核实排列顺序未变。
3. 使用 `reverse()` 修改列表元素的排列顺序。打印该列表，核实排列顺序确实变了。

4. 使用 `reverse()` 再次修改列表元素的排列顺序。打印该列表，核实已恢复到原来的排列顺序。

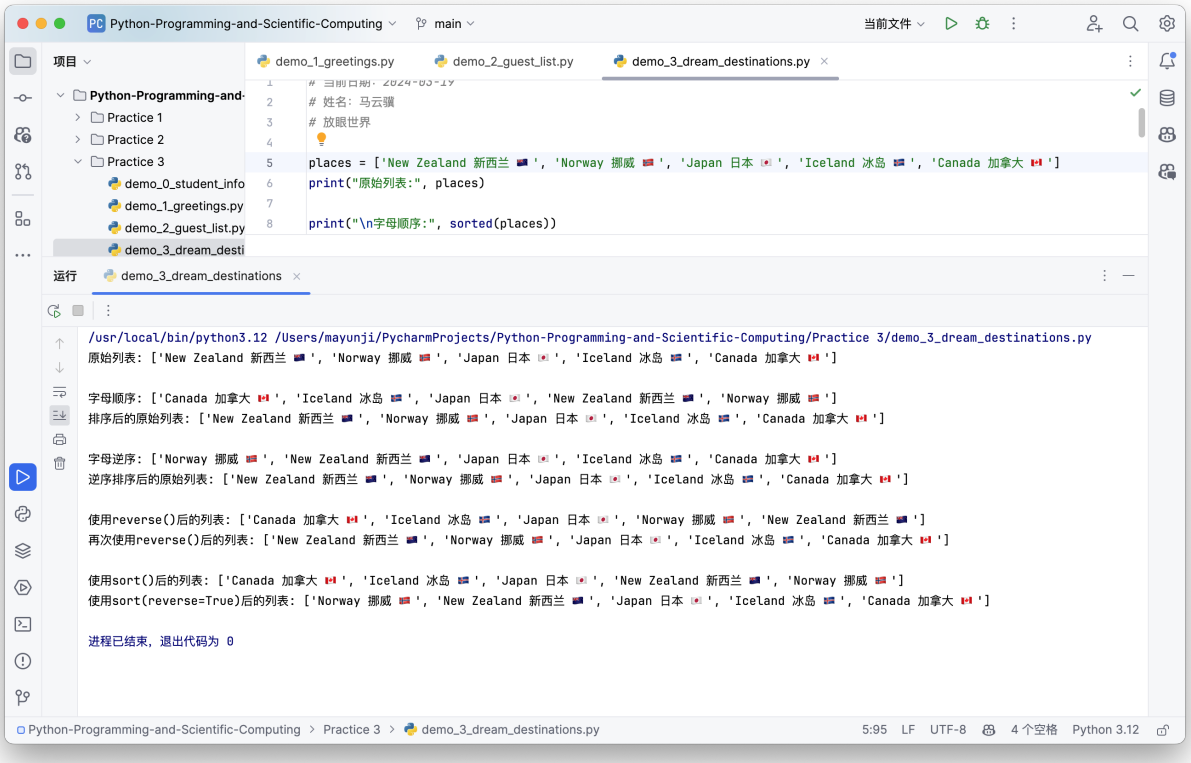
5. 使用 `sort()` 修改该列表，使其元素按字母顺序排列。打印该列表，核实排列顺序确实变了。

6. 使用 `sort()` 修改该列表，使其元素按与字母顺序相反的顺序排列。打印该列表，核实排列顺序确实变了。

3.2 程序代码

```
# 当前日期：2024-03-19
# 姓名：马云骥
# 放眼世界

places = ['New Zealand 新西兰 NZ', 'Norway 挪威 NO', 'Japan 日本 JP', 'Iceland 冰岛 IS',
'Canada 加拿大 CA']
print("原始列表:", places)

print("\n字母顺序:", sorted(places))
print("排序后的原始列表:", places)

print("\n字母逆序:", sorted(places, reverse=True))
print("逆序排序后的原始列表:", places)

places.reverse()
print("\n使用reverse()后的列表:", places)

places.reverse()
print("再次使用reverse()后的列表:", places)

places.sort()
print("\n使用sort()后的列表:", places)

places.sort(reverse=True)
print("使用sort(reverse=True)后的列表:", places)

```

3.3 结果输出



4 练习4：喜欢的地方

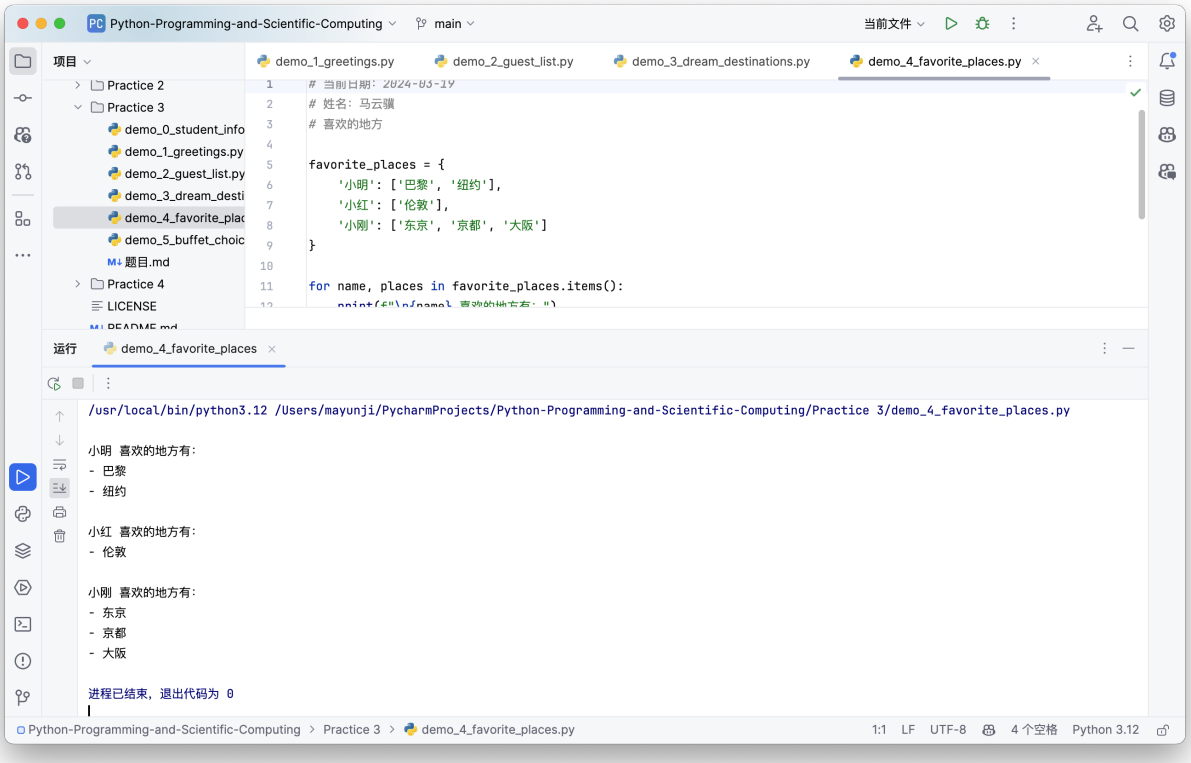
4.1 题目叙述

创建一个名为 `favorite_places` 的字典。在这个字典中，将三个人的名字用作键，并存储每个人喜欢的1～3个地方。为了让这个练习更有趣些，可以让一些朋友说出他们喜欢的几个地方。遍历这个字典，并将其中每个人的名字及其喜欢的地方打印出来。

4.2 程序代码

```python
# 当前日期：2024-03-19
# 姓名：马云骥
# 喜欢的地方

favorite_places = {
    '小明': ['巴黎', '纽约'],
    '小红': ['伦敦'],
    '小刚': ['东京', '京都', '大阪']
}

for name, places in favorite_places.items():
    print(f"\n{name} 喜欢的地方有：")
    for place in places:
        print(f"- {place}")

```

4.3 结果输出



5 练习5：自助餐

5.1 题目叙述

有一家自助式餐馆，只提供五种简单的食品。请想出五种简单的食品，并将其存储在一个元组中。

1. 使用一个 `for` 循环将该餐馆提供的五种食品都打印出来。
2. 尝试修改其中的一个元素，核实Python确实会拒绝你这样做。
3. 餐馆调整了菜单，替换了它提供的其中两种食品。请编写一个这样的代码块：给元组变量赋值，并使用一个 `for` 循环将新元组的每个元素都打印出来。

5.2 程序代码

```
# 当前日期：2024-03-19
# 姓名：马云骥
# 自助餐

# 初始菜单
foods = ('披萨🍕', '炸鸡🍗', '汉堡🍔', '沙拉🥗', '冰淇淋🍦')
for food in foods:
    print(food)

# 尝试修改其中一个元素
# foods[0] = '寿司🍣' # 这会引发TypeError

# 新菜单
```

```
foods = ('寿司🍣', '炸鸡🍗', '汉堡🍔', '沙拉🥗', '布丁🍮')
print("\n新菜单：")
for food in foods:
    print(food)

```

5.3 结果输出