

## 实战练习 2

本练习将为您提供不同类型序列的实际实现，包括列表、元组、集合和字典。我们将使用列表和循环来了解如何索引这些容器的各个项目。该练习还允许学生编写自己的函数。

### 1. 练习目标：

本练习教您以下主题：

- 如何使用列表、元组、集合和字典
- 如何将循环与列表一起使用
- 如何编写自定义函数

### 2. 介绍：

Python 提供了不同类型的数据结构作为序列。

在**序列**中，有多个值，每个值都有自己的索引。在 `python` 中，第一个值的索引为 0，第二个值的索引为 1，依此类推。这些索引用于访问序列中的特定值。

Python 提供了不同类型的序列。

**列表**是 Python 中使用的最重要的序列类型。它是相同或不同类型对象的集合。这些对象用逗号分隔，以区分括在方括号中的彼此。以下活动展示了如何在 Python 中使用列表。

**元组是不可变的 Python 对象的序列**。元组是序列，就像列表一样。元组和列表之间的区别在于，与列表不同，元组不能更改，并且元组使用括号，而列表使用方括号。

数据类型“**set**”是一种集合类型，自 2.4 版本以来一直是 Python 的一部分。集合包含唯一且不可变对象的无序集合。顾名思义，集合数据类型是数学中集合的 Python 实现。这解释了为什么与列表或元组不同的集合不能多次出现同一元素。

字典是一个无序、可更改且有索引的集合。在 Python 中，字典是用大括号编写的，并且它们有键和值。

### 3. 活动 1：

使用循环接受来自用户的 5 个值并将它们存储在**列表**中。显示列表的所有值（对象）。

**解决方案：**

```

myList=[]
for i in range(5):
    val=input("Enter a value:")
    myList.append(val)

print("The given list is")
print(myList)

```

您将得到以下输出。

```

Enter a value:1
Enter a value:Mango
Enter a value:Fruit
Enter a value:Pakistan
Enter a value:1947
The given list is
['1', 'Mango', 'Fruit', 'Pakistan', '1947']
>>>

```

#### 4. 活动 2:

通过接受来自 `user` 的 5 个整数值来重复上述代码。将这些值存储在列表中并显示给定值的总和。

**解决方案:**

```

myList=[]
for i in range(5):
    val=input("Enter a value:")
    n=int(val)
    myList.append(n)

sum=0
for n in myList:
    sum=sum+n
print("The sum of given values is ",sum)

```

您将得到以下输出。

```
Enter a value:1
Enter a value:2
Enter a value:3
Enter a value:4
Enter a value:5
The sum of given values is 15
>>>
```

### 5. 活动 3:

接受用户提供的 5 个整数值。将这些值存储在列表中并按升序显示该列表。

#### 解决方案:

创建一个新的 Python 文件并键入以下代码。  
运行代码。

```
myList=[]
for i in range(5):
    val=input("Enter a value:")
    n=int(val)
    myList.append(n)

myList.sort()
print(myList)
```

您将得到以下输出。

```
Enter a value:1
Enter a value:5
Enter a value:2
Enter a value:4
Enter a value:3
[1, 2, 3, 4, 5]
```

### 6. 活动 4:

接受用户的两个列表并显示它们的连接。

#### 解决方案:

```

myList1=[]
print("Enter objects of first list...")
for i in range(5):
    val=input("Enter a value:")
    n=int(val)
    myList1.append(n)

myList2=[]
print("Enter objects of second list...")
for i in range(5):
    val=input("Enter a value:")
    n=int(val)
    myList2.append(n)

list3=myList1+myList2;
print(list3)

```

您将得到以下输出。

```

Enter objects of first list...
Enter a value:1
Enter a value:2
Enter a value:3
Enter a value:4
Enter a value:5
Enter objects of second list...
Enter a value:6
Enter a value:7
Enter a value:8
Enter a value:9
Enter a value:0
[1, 2, 3, 4, 5, 6, 7, 8, 9, 0]
>>>

```

## 7. 活动 5:

编写 Python 代码来接受用户的列表并在其中查找所需的元素。

解决方案:

```
myList=[]
print("Enter objects of first list...")
for i in range(5):
    val=input("Enter a value:")
    n=int(val)
    myList.append(n)
print(myList)
s=input("Enter a value to find in the list ")
n=int(s)
found=n in myList
if found==True:
    print("Found")
else:
    print("Not Found")
```

您将得到以下输出。

```
Enter objects of first list...
Enter a value:1
Enter a value:2
Enter a value:3
Enter a value:4
Enter a value:5
[1, 2, 3, 4, 5]
Enter a value to find in the list 3
Found
>>>
```

如果我们再次运行该程序并输入 55 在列表中查找，则输出将如下所示。



```
Enter objects of first list...
Enter a value:1
Enter a value:2
Enter a value:3
Enter a value:4
Enter a value:5
[1, 2, 3, 4, 5]
Enter a value to find in the list 55
Not Found
>>>
```

## 8. : 活动 6:

创建一个包含姓名和电话号码字典的 Python 程序。对键字段使用单独的名字和姓氏值的元组。使用至少三个名称和数字初始化字典。要求用户通过输入名字和姓氏来搜索电话号码。如果找到则显示匹配的号码，如果未找到则显示一条消息。

### 解决方案:

```
sample={ ("sohaib", "ali"):"0246585468445", ("aib", "li"):"02465854645",
          ("sib", "ai"):"0246585468445", }
firstName = input("enter first name")
lastName = input("enter last name")

searchTuple = (firstName, lastName)
if searchTuple in sample:
    print(sample[searchTuple])
else:
    print("name not found")
```

## 课后作业:

### 9. 活动 1:

根据用户值创建两个列表。合并两个列表并按排序顺序显示。

### 10.活动 2:

重复上述活动以找到列表中的最小和最大元素。（假设所有元素都是整数值）