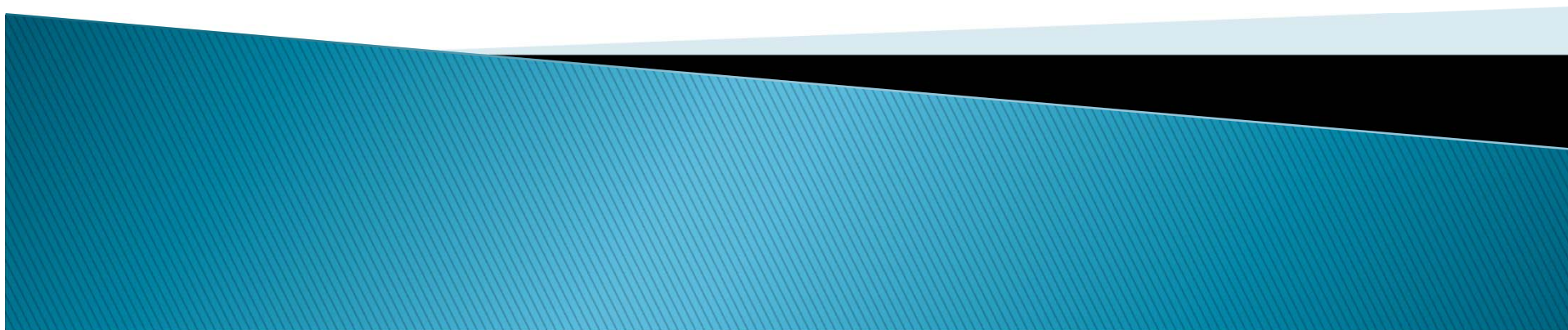


## 第四讲：列表(II)



# 遍历列表元素

- ▶ 当你要对列表中的每一个元素做同个操作时，需要逐一去访问列表的每个元素，这种访问方式称为遍历列表

```
1 magicians = ['alice', 'david', 'carolina']
2 for magician in magicians:
3     print(magician.title() + ", that was a great trick!")
4     print("I can't wait to see your next trick, " + magician.title() + ".\n")
5
6 print("Thank you everyone, that was a great magic show!")
```

运行结果：

```
Alice, that was a great trick!
I can't wait to see your next trick, Alice.

David, that was a great trick!
I can't wait to see your next trick, David.

Carolina, that was a great trick!
I can't wait to see your next trick, Carolina.

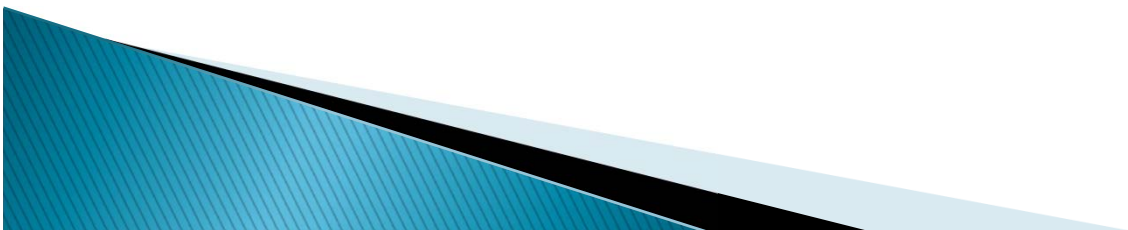
Thank you everyone, that was a great magic show!
```

# 创建数值列表

- ▶ 内置函数`range(a, b)` – 返回a到**b-1**的数
- ▶ `range`结合列表类的构造函数`list`使用可以方便的得到数值列表

```
1 numbers = list(range(1, 6))  
2 print(numbers)
```

运行结果: `[1, 2, 3, 4, 5]`



# 创建数值列表

- ▶ 内置函数`range(a, b, c)` – 返回a到**b-1**的数，步长为c

```
1 even_numbers = list(range(2, 11, 2))  
2 print(even_numbers)
```

运行结果: `[2, 4, 6, 8, 10]`

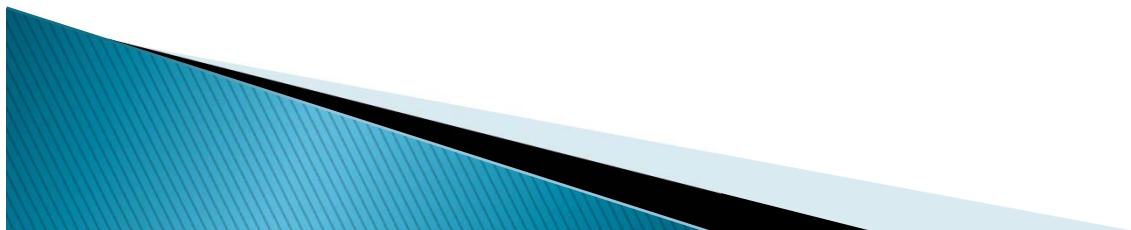


# 创建数值列表

- ▶ Range和for循环结合，可以构造出更复杂的列表，  
例如：

```
1 squares = []  
2 for value in range(1,11):  
3     square = value**2  
4     squares.append(square)  
5  
6 print(squares)
```

运行结果: [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]



# 对数值列表进行简单统计

- ▶ `min(list)` – 获得列表list中的最小值
- ▶ `max(list)` – 获得列表list中的最大值
- ▶ `sum(list)` – 求列表list中元素的和

```
>>> digits = [1, 4, 2, 7]
>>> min(digits)
1
>>> max(digits)
7
>>> sum(digits)
14
```

# 通过列表解析来生成列表

- ▶ 列表解析将for循环和创建新元素的代码合并成一行，可以更简洁的方式生成列表，例如：

```
1 squares = [value ** 2 for value in range(1, 11)]  
2 print(squares)
```

运行结果: [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

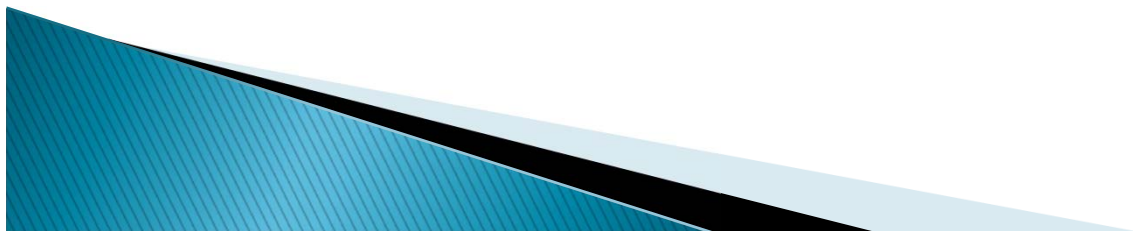


# 切片

- ▶ 当需要访问列表list的一部分元素时，可以用 `list[a:b]` 访问列表索引从a到**b-1**的元素，这种方式称为切片（slice）

```
1 players = ['charles', 'martina', 'michael', 'florence', 'eli']  
2 print(players[1:4])
```

运行结果: `['martina', 'michael', 'florence']`





# 切片

- ▶ 如果切片从头开始，可以用list[:b]的形式

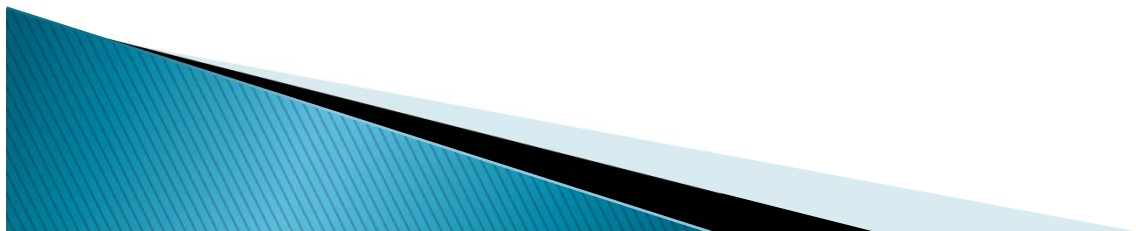
```
1 players = ['charles', 'martina', 'michael', 'florence', 'eli']  
2 print(players[:4])
```

运行结果: ['charles', 'martina', 'michael', 'florence']

- ▶ 如果切片到末尾结束，可以用list[a:]的形式

```
1 players = ['charles', 'martina', 'michael', 'florence', 'eli']  
2 print(players[2:])
```

运行结果: ['michael', 'florence', 'eli']



# 切片

- ▶ 可以用list[:]表示整个列表

```
1 players = ['charles', 'martina', 'michael', 'florence', 'eli']  
2 print(players[:])
```

运行结果: ['charles', 'martina', 'michael', 'florence', 'eli']

- ▶ 切片当中的索引也可以是负数

```
1 players = ['charles', 'martina', 'michael', 'florence', 'eli']  
2 print(players[-2:])
```

运行结果: ['florence', 'eli']



# 遍历切片

- ▶ 列表的切片也是列表，所以同样可以通过for语句遍历

```
1 players = ['charles', 'martina', 'michael', 'florence', 'eli']
2
3 print("Here are the first three players on my team:")
4 for player in players[:3]:
5     print(player.title())
```

运行结果:

```
Here are the first three players on my team:
Charles
Martina
Michael
```



# 复制列表

- 可以通过下列语句将列表listb赋值给列表lista: lista = listb[:]

```
1 my_foods = ['pizza', 'falafel', 'carrot cake']
2 friend_foods = my_foods[:]
3
4 my_foods.append('cannoli')
5 friend_foods.append('ice cream')
6
7 print("My favorite foods are:")
8 print(my_foods)
9
10 print("\nMy friend's favorite foods are:")
11 print(friend_foods)
```

运行结果:

```
My favorite foods are:
['pizza', 'falafel', 'carrot cake', 'cannoli']

My friend's favorite foods are:
['pizza', 'falafel', 'carrot cake', 'ice cream']
```

# 复制列表

- 注意，如果直接用 `lista = listb`，则 `lista` 实际上是 `listb` 的别名，并未新建列表副本，注意以下程序的不同

```
1 my_foods = ['pizza', 'falafel', 'carrot cake']
2 friend_foods = my_foods
3
4 my_foods.append('cannoli')
5 friend_foods.append('ice cream')
6
7 print("My favorite foods are:")
8 print(my_foods)
9
10 print("\nMy friend's favorite foods are:")
11 print(friend_foods)
```

运行结果：

```
My favorite foods are:
['pizza', 'falafel', 'carrot cake', 'cannoli', 'ice cream']

My friend's favorite foods are:
['pizza', 'falafel', 'carrot cake', 'cannoli', 'ice cream']
```

# 元组

- ▶ 元组就是不可修改的列表，元素用()括起来

```
1 dimensions = (200, 50)
2 print(dimensions[0])
3 print(dimensions[1])
```

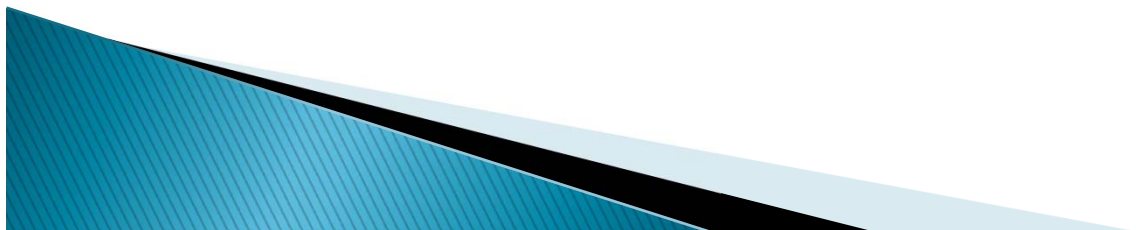
运行结果:

```
200
50
```

```
1 dimensions = (200, 50)
2 dimensions[0] = 100
```

运行结果:

```
Traceback (most recent call last):
  File "dimensions.py", line 2, in <module>
    dimensions[0] = 100
TypeError: 'tuple' object does not support item assignment
```



# 遍历元组

- ▶ 可以像列表一样，用for语句访问元组每个元素

```
1 dimensions = (200, 50)
2 for dimension in dimensions:
3     print(dimension)
```

运行结果：

```
200
50
```



# 修改元组变量

- ▶ 虽然不能单独修改元组中的一个元素，但我们可以给元组变量赋新的值

```
1 dimensions = (200, 50)
2 print("Original dimensions:")
3 for dimension in dimensions:
4     print(dimension)
5
6 dimensions = (400, 100)
7 print("\nModified dimensions:")
8 for dimension in dimensions:
9     print(dimension)
```

运行结果：

```
Original dimensions:
200
50

Modified dimensions:
400
100
```



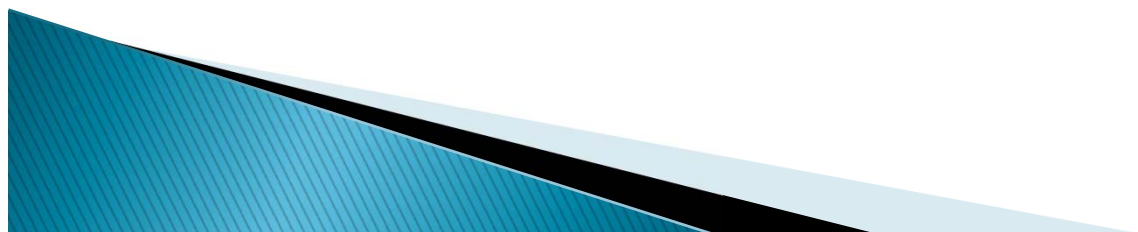
# 总结

- ▶ for循环遍历列表
- ▶ range和列表解析
- ▶ 列表切片
- ▶ 复制列表
- ▶ 元组及其常用操作
- ▶ 下节课我们将学习if语句



# 作业

- ▶ 教材中课后的练习，4-1到4-15，选一些写到你的博客上



谢谢！

