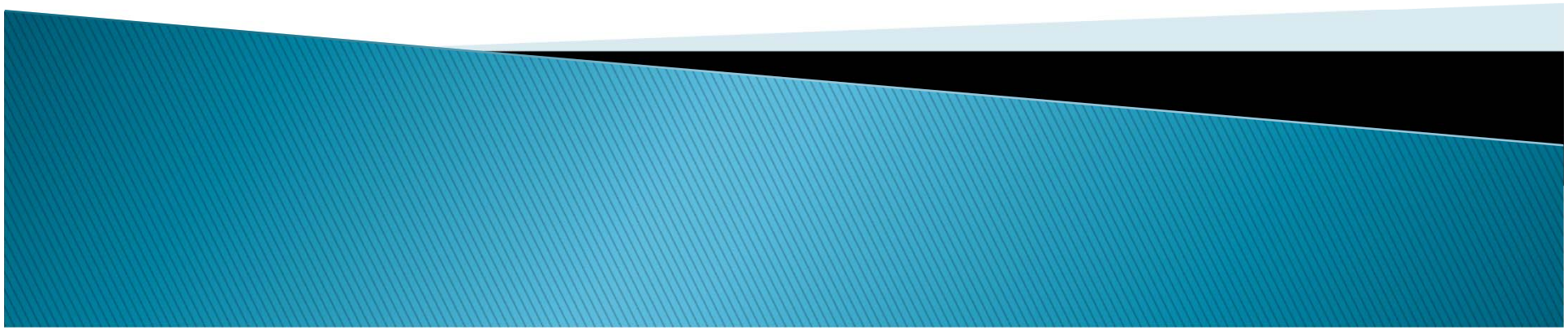
第三讲：列表(I)

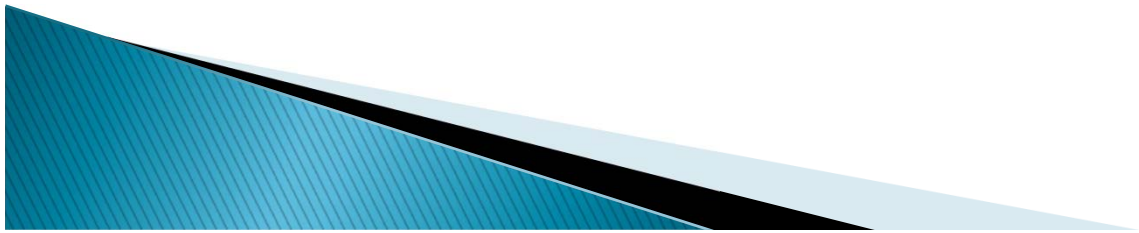


什么是列表

- 由一系列按特定顺序排列的元素组成
- 用[]将列表中的元素括起来，列表中的元素用,分隔

```
bicycles = ['trek', 'cannondale', 'redline', 'specialized']
```

- 不同于C++的数组，列表中的元素可以是不同的数据类型，例如一个列表中可以既包含字符串，也包含整数

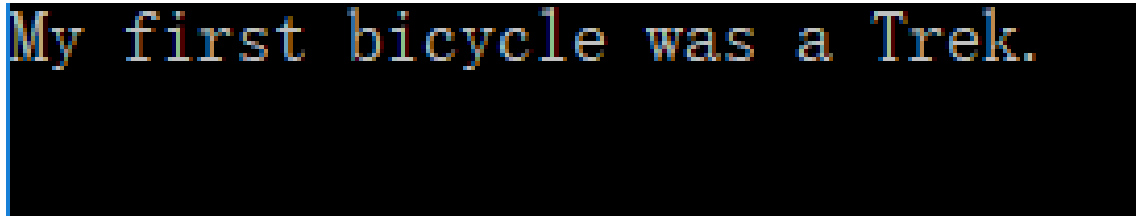


访问列表元素

- 类似C++，可以用下标的方式来访问列表中的元素，列表的下标从0开始

```
1    bicycles = ['trek', 'cannondale', 'redline', 'specialized']
2    message = "My first bicycle was a " + bicycles[0].title() + "."
3
4    print(message)
```

运行结果：



访问列表元素

- Python允许列表的下标为负数，list[-i]表示访问列表的倒数第i个元素，例如list[-1]访问list的最后一个元素

```
1    bicycles = ['trek', 'cannondale', 'redline', 'specialized']
2    message = "My first bicycle was a " + bicycles[-2].title() + "."
3
4    print(message)
```

运行结果：

```
My first bicycle was a Redline.
```

修改列表元素

- 可以通过赋值语句指定修改列表下标为i的元素

```
1  motorcycles = ['honda', 'yamaha', 'suzuki', 'ducati']
2  print(motorcycles)
3
4  motorcycles[0] = 'ducati'
5  print(motorcycles)
```

运行结果：

```
['honda', 'yamaha', 'suzuki', 'ducati']
['ducati', 'yamaha', 'suzuki', 'ducati']
```

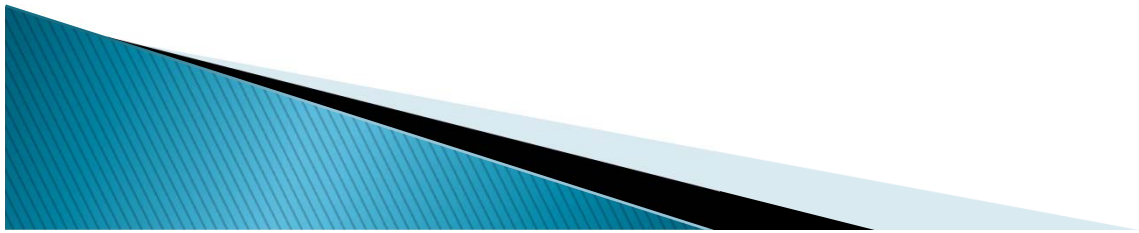
在列表末尾插入元素

- 通过列表的append(x) 方法，将x添加到列表的末尾

```
1  motorcycles = ['honda', 'yamaha', 'suzuki', 'ducati']
2  print(motorcycles)
3
4  motorcycles.append('ducati')
5  print(motorcycles)
```

运行结果：

```
['honda', 'yamaha', 'suzuki', 'ducati']
['honda', 'yamaha', 'suzuki', 'ducati', 'ducati']
```



在列表指定位置插入元素

- 通过列表的insert(i,x) 方法，将x插入到列表的第i个位置，列表原有的元素，从第i个开始，每个往后移一个位置

```
1    motorcycles = ['honda', 'yamaha', 'suzuki', 'ducati']
2    print(motorcycles)
3
4    motorcycles.insert(0, 'ducati')
5    print(motorcycles)
```

运行结果：

```
['honda', 'yamaha', 'suzuki', 'ducati']
['ducati', 'honda', 'yamaha', 'suzuki', 'ducati']
```

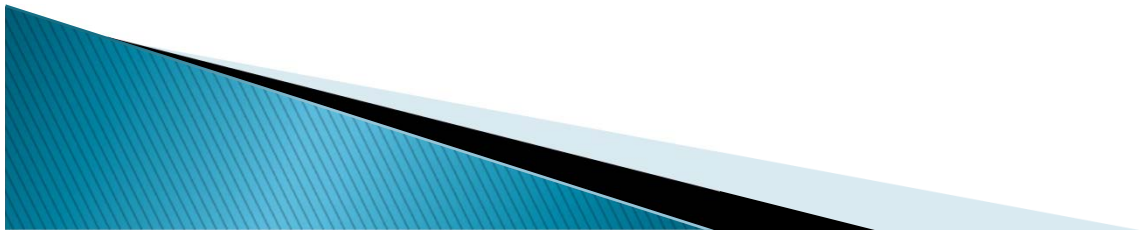
删除列表指定位置的元素

- del list[i] -- 删除列表list第i个元素

```
1    motorcycles = ['honda', 'yamaha', 'suzuki', 'ducati']
2    print(motorcycles)
3
4    del motorcycles[1]
5    print(motorcycles)
```

运行结果：

```
['honda', 'yamaha', 'suzuki', 'ducati']
['honda', 'suzuki', 'ducati']
```



弹出列表的最后一个元素

- list.pop() – 弹出列表list的最后一个元素，并返回最后一个元素的值

```
1   motorcycles = ['honda', 'yamaha', 'suzuki', 'ducati']
2   print(motorcycles)
3
4   popped_motor = motorcycles.pop()
5   print(motorcycles)
6   print(popped_motor)
```

运行结果：

```
['honda', 'yamaha', 'suzuki', 'ducati']
['honda', 'yamaha', 'suzuki']
ducati
```

弹出列表第i个元素

- list.pop(i) – 弹出列表list的第i个元素，并返回该元素的值

```
1   motorcycles = ['honda', 'yamaha', 'suzuki', 'ducati']
2   print(motorcycles)
3
4   popped_motor = motorcycles.pop(2)
5   print(motorcycles)
6   print(popped_motor)
```

运行结果：

```
['honda', 'yamaha', 'suzuki', 'ducati']
['honda', 'yamaha', 'ducati']
suzuki
```

根据值删除列表元素

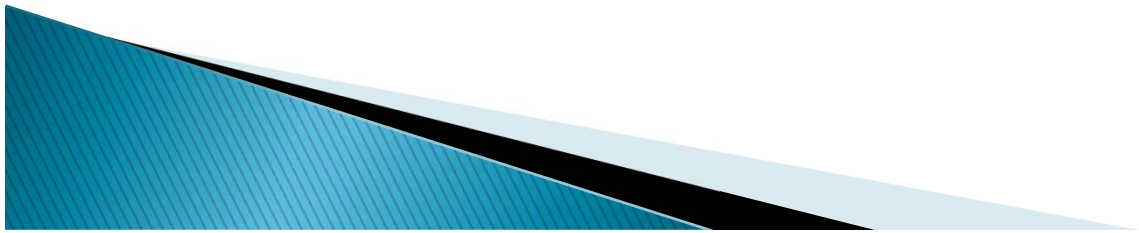
- list.remove(x) – 删除列表中第一个值为x的元素

```
1  motorcycles = ['honda', 'yamaha', 'suzuki', 'ducati']
2  print(motorcycles)
3
4  too_expensive = 'ducati'
5  motorcycles.remove(too_expensive)
6  print(motorcycles)
7  print("\nA " + too_expensive.title() + " is too expensive for me.")
```

运行结果：

```
['honda', 'yamaha', 'suzuki', 'ducati']
['honda', 'yamaha', 'suzuki']

A Ducati is too expensive for me.
```



根据值删除列表元素

- 注意，如果列表中有多个元素和要删掉的值相等，remove只删掉最先出现的元素

```
1    motorcycles = ['honda', 'ducati', 'yamaha', 'suzuki', 'ducati']
2    print(motorcycles)
3
4    motorcycles.remove('ducati')
5    print(motorcycles)
```

运行结果：

```
['honda', 'ducati', 'yamaha', 'suzuki', 'ducati']
['honda', 'yamaha', 'suzuki', 'ducati']
```

- 如果要删掉所有指定值的元素，需要用到循环语句，我们将很快学到

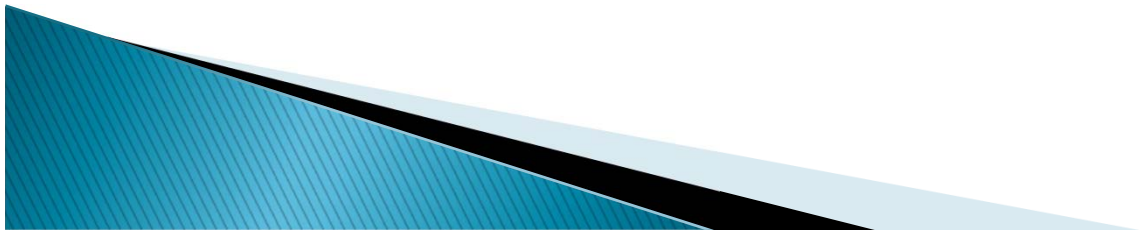
列表的排序

- list.sort() – 对列表list中的元素进行排序

```
1  cars = ['bmw', 'audi', 'toyota', 'subaru']
2  cars.sort()
3  print(cars)
```

运行结果：

```
['audi', 'bmw', 'subaru', 'toyota']
```



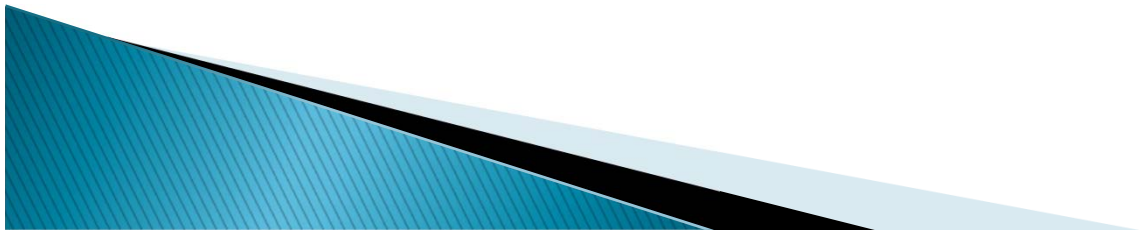
列表的反排序

- list.sort(reverse=True) – 对列表list中的元素进行排序

```
1  cars = ['bmw', 'audi', 'toyota', 'subaru']
2  cars.sort(reverse=True)
3  print(cars)
```

运行结果：

```
['toyota', 'subaru', 'bmw', 'audi']
```



列表的临时排序

- 内置函数sorted(list) – 返回对列表list中的元素进行排序后的结果，当list本身不变

```
1   cars = ['bmw', 'audi', 'toyota', 'subaru']
2
3   print("Here is the original list:")
4   print(cars)
5
6   print("\nHere is the sorted list:")
7   print(sorted(cars))
8
9   print("\nHere is the reverse alphabetical list:")
10  print(sorted(cars, reverse=True))
11
12  print("\nHere is the original list again:")
13  print(cars)
```

运行结果：

```
Here is the original list:
['bmw', 'audi', 'toyota', 'subaru']

Here is the sorted list:
['audi', 'bmw', 'subaru', 'toyota']

Here is the reverse alphabetical list:
['toyota', 'subaru', 'bmw', 'audi']

Here is the original list again:
['bmw', 'audi', 'toyota', 'subaru']
```

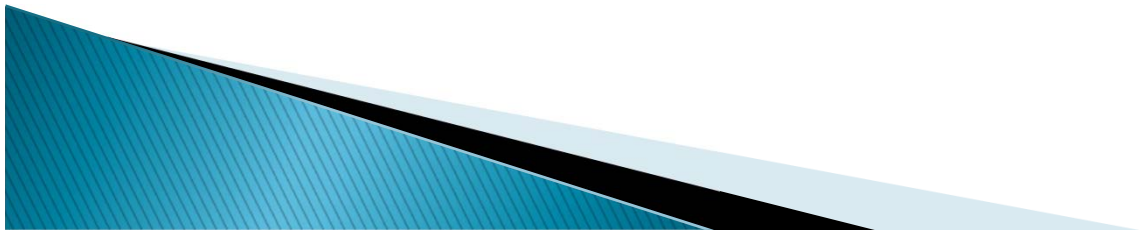
反转列表

- list.reverse() – 反转列表list

```
1    cars = ['bmw', 'audi', 'toyota', 'subaru']
2
3    cars.reverse()
4    print(cars)
```

运行结果：

```
['subaru', 'toyota', 'audi', 'bmw']
```



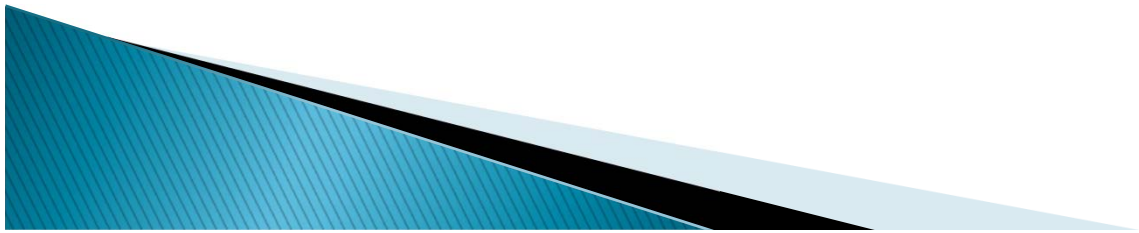
获得列表的长度

- 内置函数len(list) – 返回列表list中的元素个数

```
1    cars = ['bmw', 'audi', 'toyota', 'subaru']
2
3    print(len(cars))
```

运行结果：

```
4
```



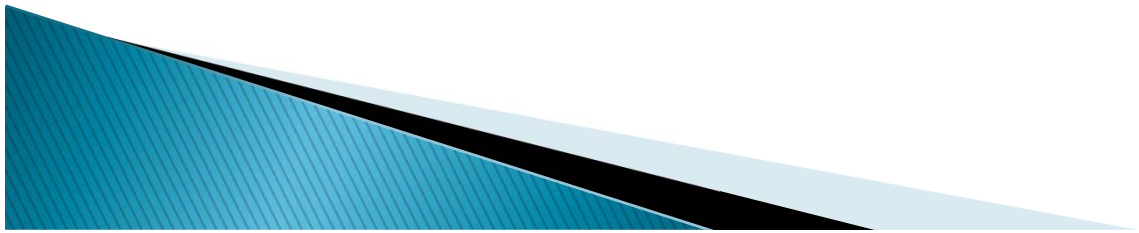
避免访问列表越界

- 类似C++的数组元素访问下标不能越界，我们在访问列表元素时也要注意这一问题，例如：

```
1    cars = ['bmw', 'audi', 'toyota', 'subaru']
2
3    print(cars[4])
```

运行结果：

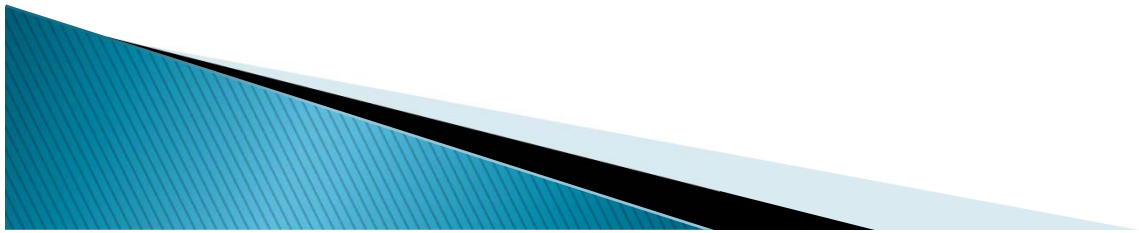
```
Traceback (most recent call last):
  File "cars.py", line 3, in <module>
    print(cars[4])
IndexError: list index out of range
```



总结

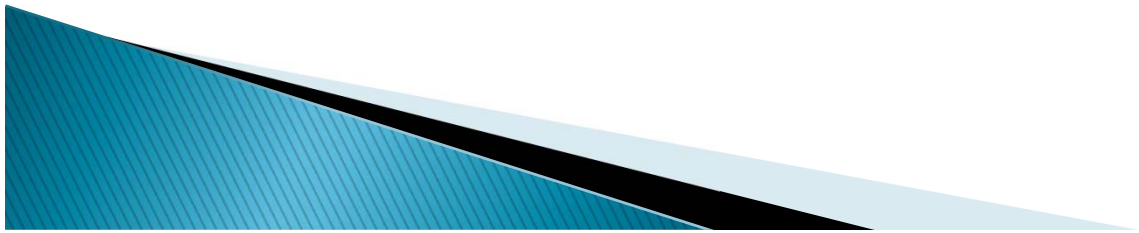
- 列表和对列表元素的访问
- 列表元素的修改、插入和删除
- 列表的排序和反转
- 求列表的长度

- 下节课我们将进一步学习更多使用列表的方法



作业

- 教材中课后的练习，3-1到3-11，选一些写到你的博客上



谢谢！