

Python 第三课

列表 (list)

蒋一歌 孙超逸

香港中文大学 (深圳) 数据科学学院

2025 年 4 月 12 日



香港中文大学 (深圳)
The Chinese University of Hong Kong, Shenzhen

① 列表的创建

② 列表索引与切片

③ 列表操作符

④ 列表的函数

⑤ 列表常用方法

创建方式 1-直接定义

使用方括号直接创建

```
1 list_a = ["皮卡丘","哆啦A梦","孙悟空"]  
2 list_b = [98,87,92,85]  
3 list_c = ["苹果",3,True]
```

创建方式 2-类型转换

list () 转换

```
1 list_a = list("abc") # ['a', 'b', 'c']
2 list_b = list((4,5,6)) # [4, 5, 6]
```

创建方式 3- range 生成

`range (start,end[,step])`

```
1 list_a = list(range(3)) # [0, 1, 2]
2 list_b = list(range(3,7)) # [3, 4, 5, 6]
3 list_c = list(range(9,3,-2)) # [9, 7, 5]
```

- 1 列表的创建
- 2 列表索引与切片
- 3 列表操作符
- 4 列表的函数
- 5 列表常用方法

访问列表元素

基础索引

```
1 stu = ["张三", "李四", "王五", "赵六"]  
2  
3 # 获取单个元素  
4 print(stu[0])    # 输出: "张三"  
5 print(stu[-1])   # 输出: "赵六" (倒数第一个)
```

- 索引从 0 开始计数
- 负数索引表示 从后向前 访问
- 越界访问会引发 IndexError

神奇的切片操作

切片语法

列表 [开始: 结束: 步长]

实际应用

```
1 num = [0,1,2,3,4,5,6,7,8,9]
2
3 print(num[2:5])      # [2,3,4] (左闭右开)
4 print(num[:3])       # [0,1,2] (省略开始)
5 print(num[6:])       # [6,7,8,9] (省略结束)
6 print(num[::2])      # [0,2,4,6,8] (步长为2)
7 print(num[::-1])     # 逆序 [9,8,...,0]
```


神奇的切片操作

索引	0	1	2	3	4	5	6	7	8	9
值	0	1	2	3	4	5	6	7	8	9

逆序区间

- 蓝色: $[2:5] \rightarrow [2, 3, 4]$
- 红色: $[1:8:2] \rightarrow [1, 3, 5, 7]$
- 绿色: $::-1 \rightarrow$ 逆序

注意

- 产生新列表 不影响原数据
- 结束索引 不包含 在结果中
- 步长可为负数实现 逆序

实战练习

基础练习

```
1 lst = ['A', 'B', 'C', 'D', 'E', 'F']
2
3 # 1. 获取前3个字母
4 print(lst[_____])
5
6 # 2. 获取最后2个字母
7 print(lst[_____])
8
9 # 3. 获取B到D的子列表
10 print(lst[_____])
```

实战练习

进阶挑战

```
1 num = list(range(20))
2
3 # 1. 获取所有偶数索引的元素
4 print(num[____])
5
6 # 2. 创建逆序列表
7 inv = num[____]
8
9 # 3. 获取第5到第15的元素，步长3
10 print(num[____])
```

参考答案

基础练习答案

```
1 print(lst[:3])      # ['A', 'B', 'C']
2 print(lst[-2:])     # ['E', 'F']
3 print(lst[1:4])     # ['B', 'C', 'D']
```

进阶挑战答案

```
1 print(num[::2])     # 偶数索引
2 inv = num[::-1]
3 print(num[5:16:3])  # 步长3
```

- 1 列表的创建
- 2 列表索引与切片
- 3 列表操作符
- 4 列表的函数
- 5 列表常用方法

列表的操作符

列表的运算符重载

表达式	结果	描述
<code>[1,2,3] + [4,5]</code>	<code>[1,2,3,4,5]</code>	列表拼接
<code>['A','B'] * 3</code>	<code>['A','B','A','B']</code>	列表重复
<code>3 in [1,2,3]</code>	<code>True</code>	成员检测
<code>for x in [1,2,3]: print(x)</code>	<code>1 2 3</code>	迭代遍历

- 1 列表的创建
- 2 列表索引与切片
- 3 列表操作符
- 4 列表的函数
- 5 列表常用方法

列表的三大核心函数

基础函数

函数	功能
len()	获取元素个数
max()	找最大值
min()	找最小值

实战应用

```
1 grade = [88,92,79,95]
2 print("人数:",len(grade))
3 print("最高分:",max(grade))
4 print("最低分:",min(grade))
```

注意

- 混合类型列表比较需谨慎
- 空列表调用 max/min 会报错

综合挑战

挑战 1-密码生成器

```
1 p = ['A', '2', '#']  
2 password = p * 2 + ['9', 'z']  
3 print("".join(password)) # 输出应该是 ?
```

挑战 2-成绩分析

```
1 grade = [85, 92, 78] + [90, 88]  
2 print("平均:", sum(grade)/len(grade))  
3 print("差距:", max(grade)-min(grade))
```

- 1 列表的创建
- 2 列表索引与切片
- 3 列表操作符
- 4 列表的函数
- 5 列表常用方法

列表的超级工具箱

append() - 添加元素

```
1 shopping = []  
2 shopping.append("牛奶")  
3 shopping.append("面包")  
4 # ['牛奶', '面包']
```

count() - 元素计数

```
1 grade = [85,90,85,88,90,85]  
2 print(grade.count(85)) # 输出 3
```

列表的超级工具箱

index() - 查找位置

```
1 fruit = ["苹果", "香蕉", "橙子"]  
2 print(fruit.index("香蕉")) # 输出 1  
3 # 找不到会引发 ValueError
```

pop() - 移除元素

```
1 do = ["作业", "购物", "洗衣"]  
2 end = do.pop(1)  
3 # end="购物", do=["作业", "洗衣"]
```

列表的超级工具箱

remove() - 删除匹配项

```
1 animal = ["狗", "猫", "鸟", "猫"]
2 animal.remove("猫")
3 print (animal) # ["狗", "鸟", "猫"]
4 # 只删除第一个匹配项
```

reverse() - 反转列表

```
1 num = [1,2,3,4]
2 num.reverse()
3 print(num) # [4,3,2,1]
```

列表的超级工具箱

sort() - 列表排序

```
1 grade = [88,92,75,85]
2 grade.sort() # 默认升序
3 print(grade) # [75, 85, 88, 92]
4 grade.sort(reverse=True) # 降序
5 print(grade) # [92, 88, 85, 75]
```

pop() vs remove()

方法	参数	返回值
pop()	索引 (可选)	被移除的元素
remove()	元素值	None