

列表数学运算

- 列表相加
 - 列表 + 列表

```
L1 = [11, 22, 33, 44]
L2 = [55, 66, 77, 88]
L3 = L1 + L2
print(L3)
# [11, 22, 33, 44, 55, 66, 77, 88]
```

- 列表相乘
 - 列表 * 正整数

```
L1 = [11, 22, 33, 44]
L2 = L1 * 3
print(L2)
# [11, 22, 33, 44, 11, 22, 33, 44, 11, 22, 33, 44]
```

列表专有方法

- append()

```
L1 = [11, 22, 33, 44]
L1.append(55)
print(L1)
# [11, 22, 33, 44, 55]
```

- clear()

```
L1 = [11, 22, 33, 44]
L1.clear()
print(L1)
# []
```

- copy()

```
L1 = [11, 22, 33, 44]
L2 = L1 # 相当于给L1起了一个新的名字叫L2, L1与L2本质是一个列表
L3 = L1.copy() # 将L1列表拷贝了一份, 叫做L2, L1与L2是不同的两个列表
L4 = L1[:] # 通过列表分片, 实现列表拷贝, L1与L4是不同的两个列表
L1.append(55)
print(L1) # [11, 22, 33, 44, 55]
print(L2) # [11, 22, 33, 44, 55]
print(L3) # [11, 22, 33, 44]
print(L4) # [11, 22, 33, 44]
```

- count()

```
L1 = [1, 2, 3, 1, 2, 1, 4, 5, 6]
print(L1.count(1)) # L1.count(n) 统计n这个元素在列表L1中出现的次数 结果=0
# 3
```

- extend()

```
L1 = [11, 22, 33, 44, 55]
L2 = [11, 22, 33, 44, 55]
L3 = [66, 77]
L1.extend(L3) # 在指定列表L1后面添加列表L3的内容
L2 = L2 + L3 # 通过列表相加实现extend操作
print(L1)    # [11, 22, 33, 44, 55, 66, 77]
print(L2)    # [11, 22, 33, 44, 55, 66, 77]
```

- insert()

```
L1 = [11, 44, 55]
L2 = L1[:1] + [22, 33] + L1[1:]
L1.insert(1, 22)    # 在列表L1下标1位置插入元素22
L1.insert(2, 33)    # 在列表L1下标2位置插入元素33
print(L1)
print(L2)
# [11, 22, 33, 44, 55]
# [11, 22, 33, 44, 55]
```

- pop()

```
L1 = [11, 22, 33, 44, 55]
num = L1.pop(3) # 将列表L1中下标为3的元素从列表中弹出，并赋值给num
print(num)
print(L1)
# 44
# [11, 22, 33, 55]
```

- remove()

```
L1 = [11, 22, 33, 44, 44, 55]
L1.remove(44) # 在列表L1中移出第一个出现的44
print(L1)
# [11, 22, 33, 44, 55]
```

- reverse()

```
L1 = [11, 22, 33, 44, 55]
L1 = L1[::-1]    # 通过列表分片实现列表的逆置
print(L1)
L1.reverse()    # 通过列表的reverse()方法实现列表逆置
print(L1)
# [55, 44, 33, 22, 11]
# [11, 22, 33, 44, 55]
```

序列共有方法

- len(L1): 求序列L1的元素个数, 求序列L1的长度
- max(L1): 求序列L1中最大元素
- min(L1): 求序列L1中最小元素

```
L1 = [11, 22, 33, 44, 55, 33, 22]
print(len(L1))
print(max(L1))
print(min(L1))
# 7
# 55
# 11
```

元组

- 元组可以理解为不能修改的列表
- 元组定义使用小括号 例如 L1 = (1, 2, 3)

字符串

- 由多个字符组成的串
- 在python中字符串使用双引号 "" 或 ' ' 引起来
- 在 "" 表示的字符串中可以直接使用 ' '
- 在 ' ' 表示的字符串中可以直接使用 "
- 可以在字符串中使用转义字符表示 " 和 ' , 例如 "ab\\'c\\'d"