

# 课程：公共操作

## 目标

- 运算符
- 公共方法
- 容器类型转换

## 一. 运算符

运算符	描述	支持的容器类型
+	合并	字符串、列表、元组
*	复制	字符串、列表、元组
in	元素是否存在	字符串、列表、元组、字典
not in	元素是否不存在	字符串、列表、元组、字典

### 1.1 +

```
1  # 1. 字符串
2  str1 = 'aa'
3  str2 = 'bb'
4  str3 = str1 + str2
5  print(str3)  # aabb
6
7
8  # 2. 列表
9  list1 = [1, 2]
10 list2 = [10, 20]
11 list3 = list1 + list2
12 print(list3)  # [1, 2, 10, 20]
13
14 # 3. 元组
15 t1 = (1, 2)
16 t2 = (10, 20)
17 t3 = t1 + t2
18 print(t3)  # (1, 2, 10, 20)
```

## 1.2 \*

---

```
1 # 1. 字符串
2 print('-' * 10) # -----
3
4 # 2. 列表
5 list1 = ['hello']
6 print(list1 * 4) # ['hello', 'hello', 'hello', 'hello']
7
8 # 3. 元组
9 t1 = ('world',)
10 print(t1 * 4) # ('world', 'world', 'world', 'world')
```

## 1.3 in或not in

---

```
1 # 1. 字符串
2 print('a' in 'abcd') # True
3 print('a' not in 'abcd') # False
4
5 # 2. 列表
6 list1 = ['a', 'b', 'c', 'd']
7 print('a' in list1) # True
8 print('a' not in list1) # False
9
10 # 3. 元组
11 t1 = ('a', 'b', 'c', 'd')
12 print('aa' in t1) # False
13 print('aa' not in t1) # True
```

## 二. 公共方法

---

函数	描述
len()	计算容器中元素个数
del 或 del()	删除
max()	返回容器中元素最大值
min()	返回容器中元素最小值
range(start, end, step)	生成从start到end的数字，步长为 step，供for循环使用
enumerate()	函数用于将一个可遍历的数据对象(如列表、元组或字符串)组合为一个索引序列，同时列出数据和数据下标，一般用在 for 循环当中。

## 2.1 len()

```
1  # 1. 字符串
2  str1 = 'abcdefg'
3  print(len(str1))  # 7
4
5  # 2. 列表
6  list1 = [10, 20, 30, 40]
7  print(len(list1))  # 4
8
9  # 3. 元组
10 t1 = (10, 20, 30, 40, 50)
11 print(len(t1))  # 5
12
13 # 4. 集合
14 s1 = {10, 20, 30}
15 print(len(s1))  # 3
16
17 # 5. 字典
18 dict1 = {'name': 'Rose', 'age': 18}
19 print(len(dict1))  # 2
```

## 2.2 del()

```
1 # 1. 字符串
2 str1 = 'abcdefg'
3 del str1
4 print(str1)
5
6 # 2. 列表
7 list1 = [10, 20, 30, 40]
8 del(list1[0])
9 print(list1) # [20, 30, 40]
```

## 2.3 max()

```
1 # 1. 字符串
2 str1 = 'abcdefg'
3 print(max(str1)) # g
4
5 # 2. 列表
6 list1 = [10, 20, 30, 40]
7 print(max(list1)) # 40
```

## 2.4 min()

```
1 # 1. 字符串
2 str1 = 'abcdefg'
3 print(min(str1)) # a
4
5 # 2. 列表
6 list1 = [10, 20, 30, 40]
7 print(min(list1)) # 10
```

## 2.5 range()

```
1 # 1 2 3 4 5 6 7 8 9
2 for i in range(1, 10, 1):
3     print(i)
4
5 # 1 3 5 7 9
6 for i in range(1, 10, 2):
7     print(i)
8
9 # 0 1 2 3 4 5 6 7 8 9
10 for i in range(10):
11     print(i)
```

注意：range()生成的序列不包含end数字。

## 2.6 enumerate()

- 语法

```
1 enumerate(可遍历对象, start=0)
```

注意：start参数用来设置遍历数据的下标的起始值，默认为0。

- 快速体验

```
1 list1 = ['a', 'b', 'c', 'd', 'e']
2
3 for i in enumerate(list1):
4     print(i)
5
6 for index, char in enumerate(list1, start=1):
7     print(f'下标是{index}, 对应的字符是{char}')
```

```
C:\Users\黑马程序员\AppData\Local\Programs\Python\Python37\python3.exe
(0, 'a')
(1, 'b')
(2, 'c')
(3, 'd')
(4, 'e')
下标是1, 对应的字符是a
下标是2, 对应的字符是b
下标是3, 对应的字符是c
下标是4, 对应的字符是d
下标是5, 对应的字符是e

Process finished with exit code 0
```

## 三. 容器类型转换

### 3.1 tuple()

作用：将某个序列转换成元组

```
1 list1 = [10, 20, 30, 40, 50, 20]
2 s1 = {100, 200, 300, 400, 500}
3
4 print(tuple(list1))
5 print(tuple(s1))
```

## 3.2 list()

---

作用：将某个序列转换成列表

```
1 t1 = ('a', 'b', 'c', 'd', 'e')
2 s1 = {100, 200, 300, 400, 500}
3
4 print(list(t1))
5 print(list(s1))
```

## 3.3 set()

---

作用：将某个序列转换成集合

```
1 list1 = [10, 20, 30, 40, 50, 20]
2 t1 = ('a', 'b', 'c', 'd', 'e')
3
4 print(set(list1))
5 print(set(t1))
```

注意：

1. 集合可以快速完成列表去重
2. 集合不支持下标

## 四. 总结

---

- 运算符
  - +
  - in / not in
- 公共方法
  - len()
  - del()
  - range()
  - enumerate()
- 数据类型转换
  - tuple()
  - list()
  - set()