

数据容器公共操作

- 目标
 - 运算符
 - 公共方法
 - 容器类型转换

1、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. 字符串
str1 = 'aa'
str2 = 'bb'
str3 = str1 + str2
print(str3) # aabb

# 2. 列表
list1 = [1, 2]
list2 = [10, 20]
list3 = list1 + list2
print(list3) # [1, 2, 10, 20]

# 3. 元组
t1 = (1, 2)
t2 = (10, 20)
t3 = t1 + t2
print(t3) # (1, 2, 10, 20)
```

1、2 *

```
# 1. 字符串
print('-' * 10) # -----

# 2. 列表
list1 = ['hello']
print(list1 * 4) # ['hello', 'hello', 'hello', 'hello']

# 3. 元组
t1 = ('world',)
print(t1 * 4) # ('world', 'world', 'world', 'world')
```

1、3 in或not in

```
# 1. 字符串
print('a' in 'abcd') # True
print('a' not in 'abcd') # False

# 2. 列表
list1 = ['a', 'b', 'c', 'd']
print('a' in list1) # True
print('a' not in list1) # False

# 3. 元组
t1 = ('a', 'b', 'c', 'd')
print('aa' in t1) # False
print('aa' not in t1) # True
```

2、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. 字符串
str1 = 'abcdefg'
print(len(str1)) # 7
```

```
# 2. 列表
list1 = [10, 20, 30, 40]
print(len(list1)) # 4

# 3. 元组
t1 = (10, 20, 30, 40, 50)
print(len(t1)) # 5

# 4. 集合
s1 = {10, 20, 30}
print(len(s1)) # 3

# 5. 字典
dict1 = {'name': 'Rose', 'age': 18}
print(len(dict1)) # 2
```

2、2 del()

```
# 1. 字符串
str1 = 'abcdefg'
del str1
print(str1)

# 2. 列表
list1 = [10, 20, 30, 40]
del(list1[0])
print(list1) # [20, 30, 40]
```

2、3 max()

```
# 1. 字符串
str1 = 'abcdefg'
print(max(str1)) # g

# 2. 列表
list1 = [10, 20, 30, 40]
print(max(list1)) # 40
```

2、4 min()

```
# 1. 字符串
str1 = 'abcdefg'
print(min(str1)) # a

# 2. 列表
list1 = [10, 20, 30, 40]
print(min(list1)) # 10
```

2、5 range()

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

# 1 3 5 7 9
for i in range(1, 10, 2):
    print(i)

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

注意：range()生成的序列不包含end数字。

2、6 enumerate()

- 语法

`enumerate`(可遍历对象, `start=0`)

`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。

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

```
list1 = ['a', 'b', 'c', 'd', 'e']

for i in enumerate(list1):
    print(i)

for index, char in enumerate(list1, start=1):
    print(f'下标是{index}, 对应的字符是{char}')
```

```
D:\Anaconda3\python.exe "Z:/01 青灯教育课程/01 青灯教育-vip课程/
(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
```

3、容器类型转换

3、1 tuple()

作用：将某个序列转换成元组

```
list1 = [10, 20, 30, 40, 50, 20]
s1 = {100, 200, 300, 400, 500}

print(tuple(list1))
print(tuple(s1))
```

3、2 list()

作用：将某个序列转换成列表

```
t1 = ('a', 'b', 'c', 'd', 'e')
s1 = {100, 200, 300, 400, 500}

print(list(t1))
print(list(s1))
```

3、3 set()

作用：将某个序列转换成集合

```
list1 = [10, 20, 30, 40, 50, 20]
t1 = ('a', 'b', 'c', 'd', 'e')

print(set(list1))
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

4、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()