

# 元组

与列表非常相似，但有一个关键的区别：元组是不可变的。

```
# 创建元组
my_tuple = (1, 2, 3)
print(my_tuple) # 输出: (1, 2, 3)
```

```
(1, 2, 3)
```

```
# 访问元组元素
print(my_tuple[0])
```

```
1
```

```
# 元组不可变
my_tuple[0] = -1
```

```
-----
TypeError                                Traceback (most recent call last)

<ipython-input-3-94c00a2d121b> in <module>()
      1 # 元组不可变
----> 2 my_tuple[0] = -1
```

```
TypeError: 'tuple' object does not support item assignment
```

```
# 元组的空值和单一值:
# 创建只有一个元素的元组时，需要在该元素后面加上逗号，否则 Python 不会识别它为元组。
my_tuple = (1,)
print(type(my_tuple))
```

```
<class 'tuple'>
```

```
# 元组与列表转换
my_list = [1, 2, 3]
my_tuple = tuple(my_list)
print(my_tuple)

new_list = list(my_tuple)
print(new_list)
```

```
(1, 2, 3)
[1, 2, 3]
```

## 集合

集合（Set）是一个无序的不重复元素序列，基本用途包括成员测试和消除重复元素。集合对象还支持数学运算，如并集、交集、差集和对称差集。

```
# 创建集合
my_set = {1, 2, 3, 3, 2}
print(my_set) # 输出: {1, 2, 3}, 自动去除重复项

# 创建一个空集合，必须使用 set() 而不是 {}, 后者用于创建一个空字典
new_set=set()
print(type(new_set))
```

```
{1, 2, 3}
<class 'set'>
```

```
my_set={1,2,3,4}
# 添加
my_set.add(5)
print(my_set)

# 删除
my_set.remove(2)
print(my_set)
```

```
{1, 2, 3, 4, 5}
{1, 3, 4, 5}
```

```
# 数据去重： 利用集合无重复元素的特性，可以方便地从列表或其他序列中移除重复项。
items = [1, 2, 2, 3, 3, 3]
unique_items = set(items)
print(unique_items) # 输出: {1, 2, 3}
```

```
{1, 2, 3}
```

```
# 集合支持数学运算，如并集、交集、差集等。
# 定义两个集合
set1 = {1, 2, 3, 4, 5}
set2 = {4, 5, 6, 7, 8}

# 并集：使用 | 运算符，得到包含两个集合中所有元素的新集合
union = set1 | set2 # 结果为 {1, 2, 3, 4, 5, 6, 7, 8}
print("并集:", union)

# 交集：使用 & 运算符，得到同时存在于两个集合中的元素集合
```

```
intersection = set1 & set2 # 结果为 {4, 5}
print("交集:", intersection)

# 对称差集: 使用 ^ 运算符, 得到只存在于其中一个集合中的元素集合
symmetric_difference = set1 ^ set2 # 结果为 {1, 2, 3, 6, 7, 8}
print("对称差集:", symmetric_difference)

# 差集: 使用 - 运算符, 得到存在于第一个集合但不在第二个集合中的元素集合
difference = set1 - set2 # 结果为 {1, 2, 3}
print("差集:", difference)

# 超集检查: 使用 >= 运算符, 检查第一个集合是否包含第二个集合的所有元素
is_superset = set1 >= set2 # 结果为 False, 因为set1不包含set2的所有元素
print("set1 是否为 set2 的超集:", is_superset)

# 子集检查: 使用 <= 运算符, 检查第一个集合的所有元素是否都存在于第二个集合中
is_subset = set1 <= set2 # 结果为 False, 因为set1包含set2中没有的元素
print("set1 是否为 set2 的子集:", is_subset)
```

```
并集: {1, 2, 3, 4, 5, 6, 7, 8}
交集: {4, 5}
对称差集: {1, 2, 3, 6, 7, 8}
差集: {1, 2, 3}
set1 是否为 set2 的超集: False
set1 是否为 set2 的子集: False
```

## 作业

oj平台: [www.cqvist.club](http://www.cqvist.club)

- 并集
- 没出现的数字个数
- 区间合并
- 集福卡