

# 第5章 集合和字典

CS, ZJU  
2018年12月

# Overview

- ◎ 集合和集合的运算
- ◎ 字典和字典的运算及函数
- ◎ 集合和字典的应用

# 5.1 集合

集合（set）是一类容器，元素没有先后顺序，并且元素的值不重复。

集合的字面量用花括号 {}

`{1, 5, 7}`

`{'apple', 'orange', 'pear', 'banana'}`

# 创建集合

- 直接给变量赋值一个集合字面量

```
fruit = {'apple', 'orange', 'pear', 'banana'}
```

- 使用set()创建一个空集合

```
emp = set()
```

注：emp = {} #创建一个空字典

- 使用set()将列表或元组转换成集合

```
prime = set([1,3,5,7,11])
```

结果是：

```
{1, 3, 5, 7, 11}
```

# 创建集合（续）

- ◎ 集合的值不重复。

创建集合的时候，python会消除重复的值。

```
fruit = {'apple', 'orange', 'apple', 'pear', 'orange', 'banana'}
```

结果是：

```
{'apple', 'pear', 'banana', 'orange'}
```

# 操作和访问集合的元素

- ◎ 可以用add()和remove()添加和删除集合元素
- ◎ 可以用min()、max()、len()和sum()对集合操作。
- ◎ 集合内的元素是无序的，所以不能像列表那样通过下标来访问集合元素。

# 操作和访问集合的元素（续）

下表给出了集合的操作和访问函数。表中示例中的s的值为{2,3,5,7,11}。

函数	示例	结果	说明
len()	len(s)	5	返回集合中元素的数量
min()	min(s)	2	返回集合中最小的元素
max()	max(s)	11	返回集合中最大的元素
sum()	sum(s)	27	将集合中所有的元素累加起来
add()	s.add(13)	{2,3,5,7,11,13}	将一个元素加入集合中
remove()	s.remove(3)	{2,5,7,11}	从集合中删除一个元素，如果这个元素在集合中不存在，则抛出KeyError异常

# 操作和访问集合的元素（续）

- ◎ 用循环实现遍历集合中的各个元素

```
s = {2,3,5,7,11}
```

```
for i in s:  
    print(i)
```

输出：

```
2  
3  
5  
7  
11
```



# 元素、子集、超集和相等判断

- 运算符in和not in 用于判断某个元素是否在集合中。

```
s = {2,3,5,7,11}
```

```
print(5 in s)
```

```
print(4 not in s)
```

输出：

```
True
```

```
True
```

# 元素、子集、超集和相等判断（续）

- ◎ 如果集合s1中的元素，同样都在集合s2中，则称s1为s2的子集，s2为s1的超集。
  - 使用s1.issubset(s2)来判断s1是否为s2的子集。
  - 使用s2.issuperset(s1)来判断s1是否为s2的超集。

```
s1 = {2,3,5,7}
```

```
s2 = {1,2,3,4,5,6,7}
```

```
print(s1.issubset(s2))
```

```
print(s2.issuperset(s1))
```

输出：

```
True
```

```
True
```

## 元素、子集、超集和相等判断（续2）

- 使用关系运算符==和!=

判断2个集合是否包含完全相同的元素。

```
s1 = {2,3,5,7}  
s2 = {3,2,7,5}  
print(s1 == s2)
```

输出：

True

# 元素、子集、超集和相等判断（续3）

## ◎ 使用关系运算符<,<=,>,>=。

- 如果s1是s2的真子集，则s1<s2是True
- 如果s1是s2的子集，则s1<=s2是True
- 如果s1是s2的真超集，则s1>s2是True
- 如果s1是s2的超集，则s1>=s2是True

注：s1是s2的真子集的意思是s1是s2的子集，但是s2中至少有一个s1中不存在的元素；s1是s2的真超集的意思是s1是s2的超集，但是s1中至少有一个s2中不存在的元素。

# 集合运算

- 通过集合的函数或运算符进行集合的并集、交集、差集和对称差的集合运算。

假设2个集合：s1 = {2,3,5,7,11},s2 = {2,3,4,5,6,7}，下列给出4种集合运算的示例。

运算	函数	运算符	示例	结果	说明
并集	union()		s1.union(s2)	{2,3,4,5,6,7,11}	结果是包含两个集合中所有元素的新集合
交集	intersection()	&	s1 & s2	{2,3,5,7}	交集是只包含两个集合中都有的元素的新集合
差集	difference()	-	s1 - s2	{11}	s1-s2的结果是出现在s1但不出现在s2的元素的 新集合
对称差	symmetric_difference()	^	s1 ^ s2	{4,6,11}	结果是一个除了共同元素之外的所有元素

找鞍点（一个矩阵元素的“鞍点”是指该位置上的元素值在该行上最大、在该列上最小，设只有一个鞍点）

- ⦿ `n=int(input())`
- ⦿ `a=[]`
- ⦿ `for i in range(0,n):`
- ⦿     `b=input().split()`
- ⦿     `a.insert(i,b)`
- ⦿ `c=[]`
- ⦿ `d=[]`
- ⦿ `for i in range(0,n):`
- ⦿     `maxa=max(int(a[i][j]) for j in range(n))`
- ⦿     `mina=min(int(a[k][i]) for k in range(0,n))`
- ⦿     `c+=[(i,j) for j in range(n) if int(a[i][j])==maxa]`
- ⦿     `d+=[(k,i) for k in range(n) if int(a[k][i])==mina]`
- ⦿ `c=list(set(c)&set(d))`
- ⦿ `if (c!=[]):`
- ⦿     `print(c[0])`
- ⦿ `else:`
- ⦿     `print("NONE")`

# 列表去重,保持原有顺序

- ◎ `mailto = ['cc', 'bbbb', 'afa', 'sss', 'bbbb', 'cc', 'shafa']`
- ◎ `addr_to = list(set(mailto))`
- ◎ `print(addr_to)`
  
- ◎ `addr_to.sort(key = mailto.index)`
- ◎ `print(addr_to)`