

王者荣耀化妆舞会





1. 探索新知

1.1

创建签到表



我邀请了好朋友：静静、铁柱、米粒、瞳瞳、嘟嘟



我邀请了好朋友：小奶酪、倩倩、冠冠、卡卡、轩轩

我邀请了好朋友：马里奥、狗蛋、二丫、冠冠、卡卡



p仔和他的朋友博克斯和小E，邀请了各自的好朋友们来参加这次的王者荣耀主题舞会。他们需要制作好各自的好友签到表。



最近新学习了一种名为“集合”的数据类型，就用它来存储大家各自的签到表吧！

1.1

创建签到表

• 集合【set】

- 在Python中，集合【set】是一种无序且不重复的数据集合
- 它是由一对花括号 {} 内放置的唯一元素组成
- 集合的主要目的是用于去重

```
num = {1, 2, 3, 1, 2, 3, 4}
name = {'李白', '杜甫', '李白', '大强', '大强'}
print(num)           # 输出 -----
print(name)         # 输出 -----
```

• 创建集合

- 创建集合一般格式：集合名 = {元素1, 元素2, 元素3}
- 创建空集合：集合名 = set()
- 注意：创建空集合的方法是set()，而不能直接集合名 = {}

```
a = {}
print(type(a))
```

输出：
<class 'dict'> ... 字典类型

```
a = set()
print(type(a))
```

输出：
<class 'set'> ... 集合类型

1.1

创建签到表

试着用集合列出三人各自的好友签到表：

```
# 三人列出各自的好友签到表
pzai = {'马里奥','狗蛋','二丫','冠冠','卡卡'}
xiaoE = {'静静','铁柱','米粒','瞳瞳','嘟嘟'}
boks = {'小奶酪','倩倩','冠冠','卡卡','轩轩'}
```

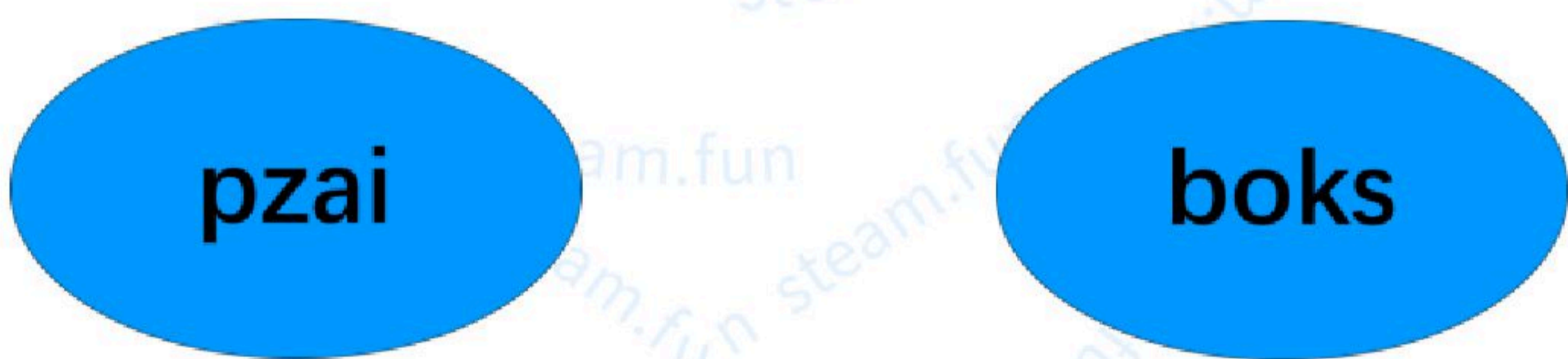
1.2

交集、差集、并集

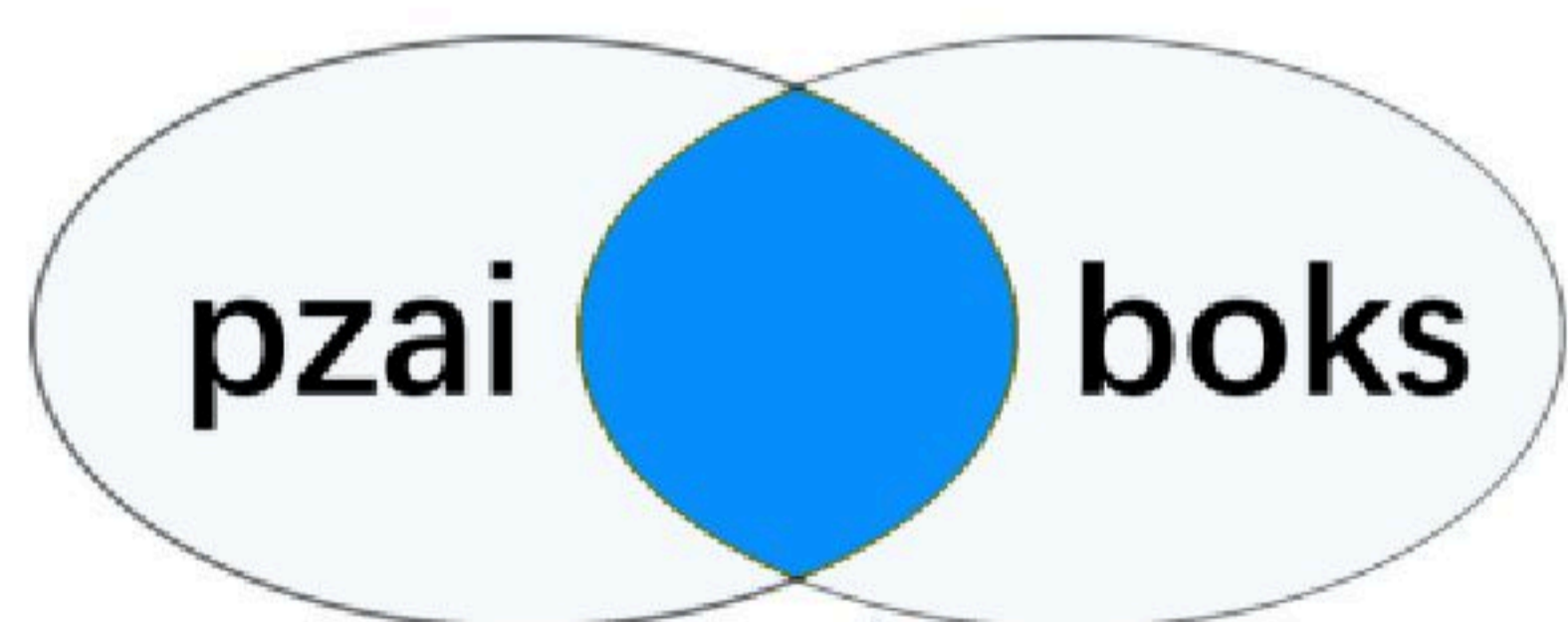
pzai和boks有一些共同的好友，请找出这些共同好友，防止重复邀请

交集

假设有如下所示的两个集合pzai、boks，分别表示pzai的好朋友和boks的好朋友



集合pzai、boks的关系如下，中间蓝色区域表示pzai和boks共同的好朋友



```
pzai = {'马里奥','狗蛋','二丫','冠冠','卡卡'}
boks = {'小奶酪','倩倩','冠冠','卡卡','轩轩'}
```

用 common 表示 pzai、boks 的共同好友

```
common = pzai.intersection(boks)
print(common)
```

输出：

common:
共同的

intersection:
交叉，交叉点

1.2

交集、差集、并集

签到表中有一些是独属于pzai、boks的朋友，请找出这些朋友

• 差集

给定两个集合圈pzai、boks，所有独属于pzai的朋友所组成的集合（图1蓝色区域），叫做集合pzai减去集合boks的差集；

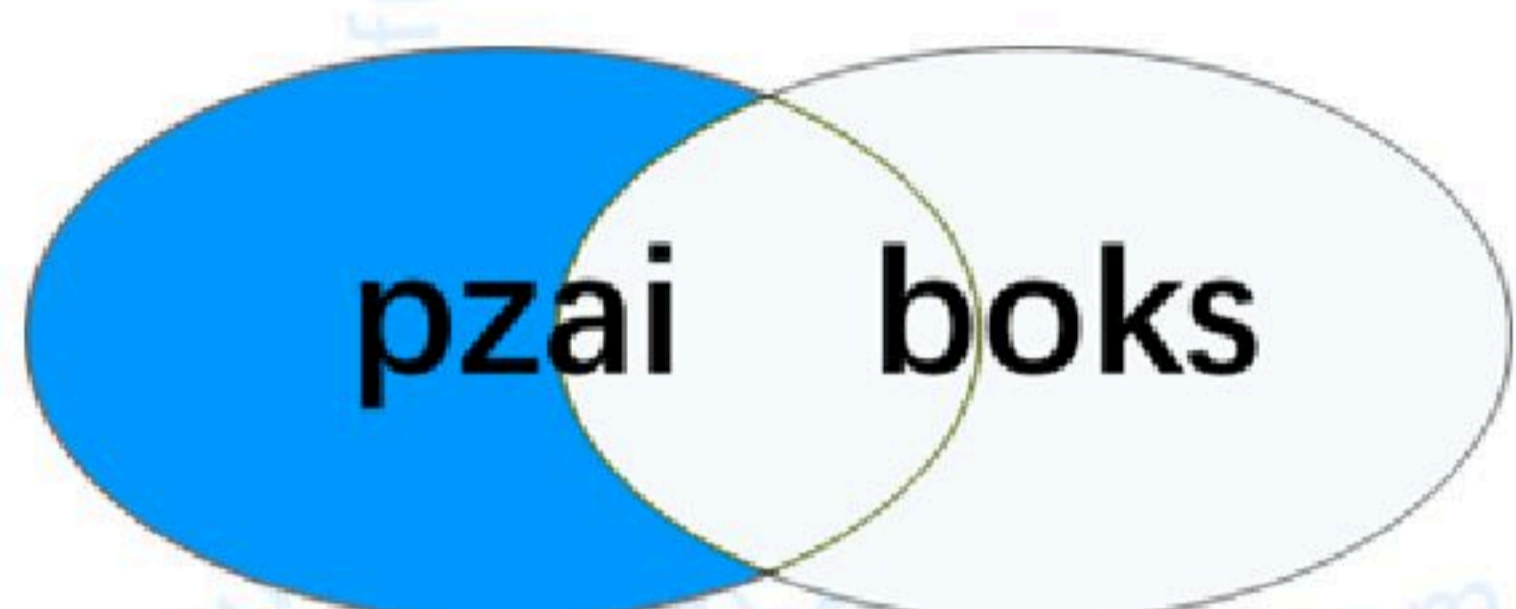


图1

pzai = {'马里奥','狗蛋','二丫','冠冠','卡卡'}

boks = {'小奶酪','倩倩','冠冠','卡卡','轩轩'}

```
own_p = pzai.difference(boks)
print(own_p)
```

输出：

own:
属于自己的

difference:
不同，差异

所有独属于boks的朋友所组成的集合（图2蓝色区域），叫做集合boks减去集合pzai的差集

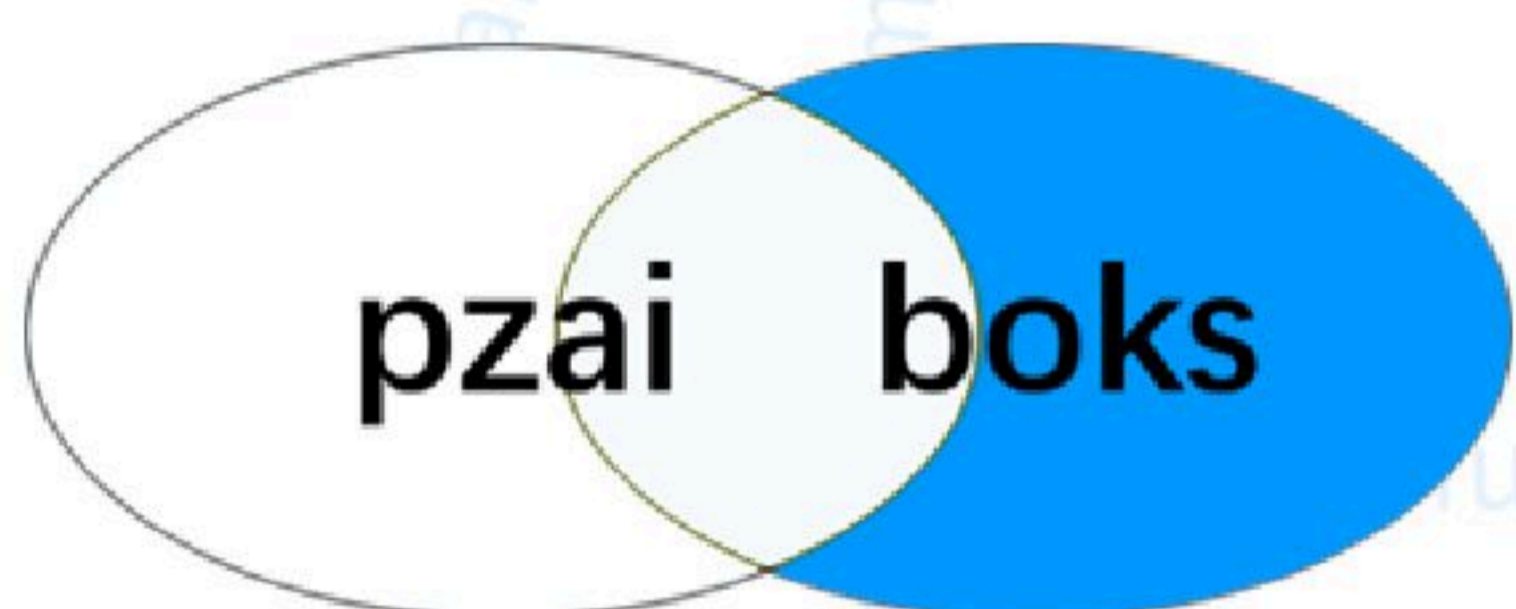


图2

pzai = {'马里奥','狗蛋','二丫','冠冠','卡卡'}

boks = {'小奶酪','倩倩','冠冠','卡卡','轩轩'}

```
own_b = boks.difference(pzai)
print(own_b)
```

输出：

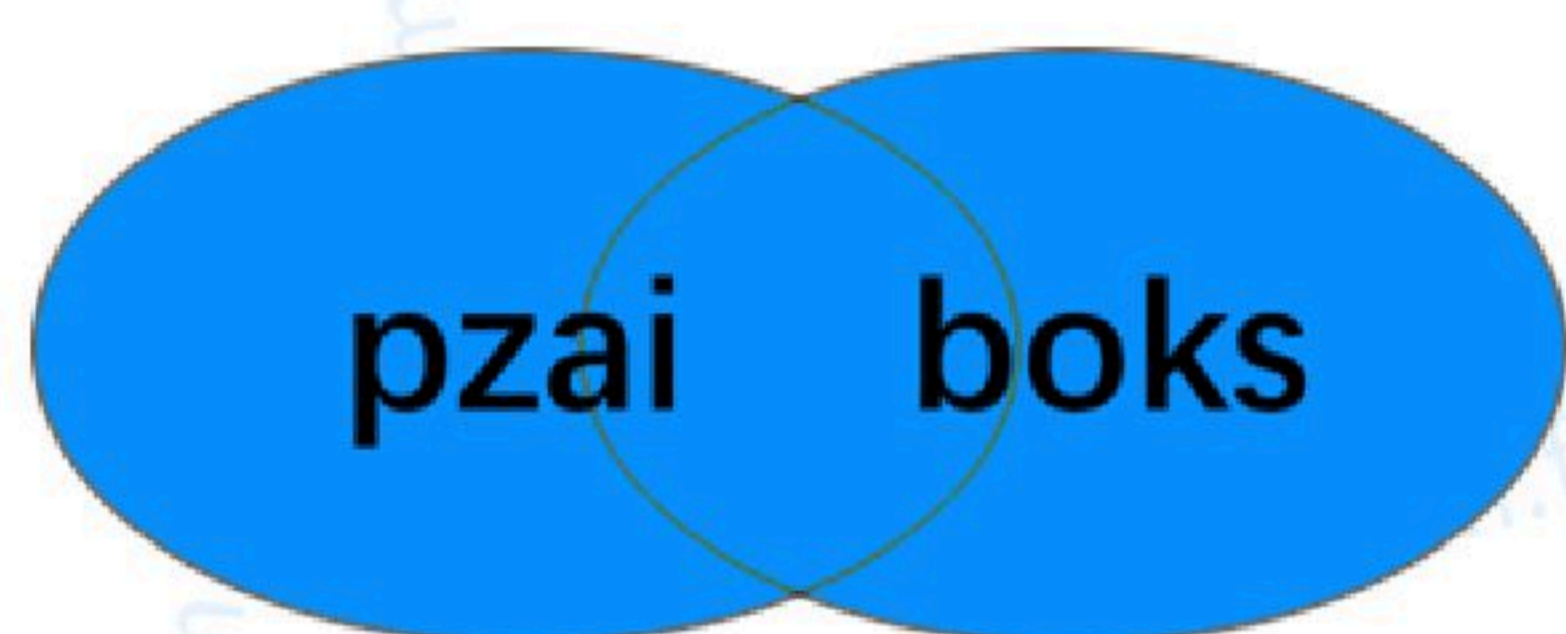
1.2

交集、差集、并集

将所有人的好友都写在一张签到表上，注意不重不漏

- 并集

给定两个集合圈pzai、boks，所有独属于pzai的朋友所组成的集合（图1蓝色区域），叫做集合pzai减去集合boks的差集；



pzai = {'马里奥','狗蛋','二丫','冠冠','卡卡'}

boks = {'小奶酪','倩倩','冠冠','卡卡','轩轩'}

```
set1 = pzai.union(boks)
print(set1)
```

输出：

union: 联合

三人好友签到总表：

```
# 将所有人的好友都写在一张签到表上，注意不重不漏
set1 = pzai.union(boks)
all_set = set1.union(xiaoE)
print(all_set)
```

输出：

1.3

签到表人员增删

舞会的前两天，收到大家的信息，“卡卡”临时有事无法到场，

“小叶子”听说活动主动报名想要参加，所以需要调整签到表



我们需要移除集合中的元素“卡卡”，并添加新的元素“小叶子”。该怎么操作呢？

● 移除集合中的元素 `remove()`

```
# 移除集合中的元素“卡卡”  
all_set.remove('卡卡')  
print(all_set)
```

输出：_____

● 添加元素到集合 `add()`

```
# 添加“小叶子”到集合中  
all_set.add('小叶子')  
print(all_set)
```

深入理解无序性：
集合内的元素无特定顺序，所以新插入的元素也可能出现在任意位置

输出：

{'狗蛋', '轩轩', '二丫', '静静', '嘟嘟', '小奶酪', '铁柱', '小叶子', '马里奥', '冠冠', '瞳瞳', '倩倩', '米粒'}



思考：

想要按照特定顺序存储元素，可以使用什么数据类型？

1.3

签到表人员增删

完成签到表的增删操作：

```
all_set.remove('卡卡')  
all_set.add('小叶子')
```

最后我们来输出签到表吧！
注意要挨个输出好友姓名，不重不漏哦！



1.4

输出签到表

遍历签到表：



想要挨个输出签到表中的好友姓名，
该怎么实现呢？

```
print('—————签到表—————')  
for i in all_set:  
    print(i)
```

完整代码

```
# 三人列出各自的好友签到表
pzai = {'马里奥', '狗蛋', '二丫', '冠冠', '卡卡'}
xiaoE = {'静静', '铁柱', '米粒', '瞳瞳', '嘟嘟'}
boks = {'小奶酪', '倩倩', '冠冠', '卡卡', '轩轩'}

# pzai 和 boks 有一些共同的好友, 请找出这些共同好友, 防止重复邀请
common = pzai.intersection(boks)
print(common)

# 有一些是独属于pzai、boks的朋友, 请找出这些朋友
own_p = pzai.difference(boks)
print(own_p)
own_b = boks.difference(pzai)
print(own_b)

# 所有人的好友都写在一张签到表上, 注意不重不漏
set1 = pzai.union(boks)
all_set = set1.union(xiaoE)
print(all_set)

# 舞会的前两天, 收到大家的信息, “卡卡”临时有事无法到场, “小叶子”听说
# 活动主动报名想要参加, 所以需要调整签到表
all_set.remove('卡卡')
all_set.add('小叶子')
print('—————签到表—————')
for i in all_set:
    print(i)
```





2. 强化练习

1. 集合的特性是什么？ ()

- A. 元素按照插入顺序排序
- B. 元素可以重复
- C. 元素是无序的
- D. 元素可以通过索引访问

2. 假设集合A中有元素{1, 2, 3}, 集合B中有元素{3, 4, 5}, 两个集合的交集是什么？ ()

- A. {1, 2, 3, 4, 5}
- B. {1, 2}
- C. {3}
- D. {4, 5}

3. 下面哪个方法可以遍历集合？ ()

- A. add()
- B. discard()
- C. remove()
- D. for x in my_set:
print(x)

4. 想要将一组数据1、1、2、3、4、4进行去重, 应该使用下面哪种数据类型存储？ ()

- A. 列表
- B. 字典
- C. 元组
- D. 集合

5. 下面哪个方法可以求取两个集合的并集？ ()

- A. intersection()
- B. union()
- C. difference()
- D. add()

3. 术语箱

intersection 交叉, 交叉点

common 共同的

difference 不同, 差异

own 属于自己的

union 联合

4. 课后挑战

火眼金睛

统计一共出现了多少个不同字母

编写一段代码, 要求如下:

已知 `str1 = "apple"`, `str2 = "banana"`

- (1) 将两字符串合并成一个字符串`str_all`
- (2) 使用`set(str_all)`将字符串转换为集合
- (3) 输出集合的长度`len()`

