

# Python语言基础与应用

数据类型 / 容器类型：集合

陈斌 北京大学 [gischen@pku.edu.cn](mailto:gischen@pku.edu.cn)



# 容器类型：集合

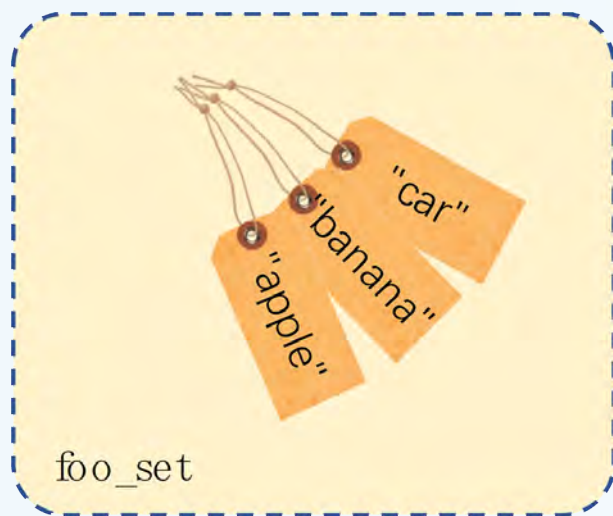
- › 标签的容器
- › 创建一个集合
- › 访问集合中的元素
- › 集合的运算
- › 什么时候用集合

# 标签的容器

## › “标签袋”

通过改造字典类型，去掉关联数据值，只留下标签的新容器类型

## › 集合是**不重复元素**的**无序**组合





# 创建一个集合

## › 创建集合：{}或者set()

用set()创建空集

可用set()从其它序列转换生成集合

## › 集合会自动忽略重复的数据

## › 集合中不能加入可变类型数据

# 更新一个集合

## › 增长集合

add: 添加一个数据

update: 批量添加数据

## › 缩减集合

remove/discard: 删除指定数据

pop: 删除任意数据并返回  
值

clear: 清空集合

## › 集合大小

len函数

```
>>> aset
{'c', 'a', 'b'}
>>> aset.add(1.23)
>>> aset
{'c', 1.23, 'a', 'b'}
>>> aset.remove('b')
>>> aset
{'c', 1.23, 'a'}
>>> aset.pop()
'c'
>>> aset
{1.23, 'a'}
>>> aset.clear()
>>> aset
set()
```

# 访问集合中的元素

## › in

判断元素是否属于集合

## › pop

删除数据元素的同时，返回它的值

取遍所有数据元素之后，集合成为一个空集

可以用copy操作先给集合制作一个“替身”

## › 迭代循环

```
for a in aset:
```

# 集合运算

## › 生成新集合运算

运算	运算符	新集合方法	更新原集合方法
并	$a \mid b$	union	update
交	$a \& b$	intersection	intersection_update
差	$a - b$	difference	difference_update
对称差	$a \wedge b$	symmetric_difference	symmetric_difference_update

## › 关系判定

$\leq, =, >$ : 子集/真子集/超集/真超集

## › 交集

`isdisjoint()`: 两集合交集是否为空

# 集合运算

```
>>> set()
set()
>>> aset = set('abc')
>>> aset
{'c', 'a', 'b'}
>>> 'a' in aset
True
>>> aset | set('bcd')
{'c', 'd', 'a', 'b'}
>>> aset & set(['b', 'c', 'd'])
{'c', 'b'}
>>> aset - set(['b', 'c', 'd'])
{'a'}
>>> aset ^ set('bcd')
{'a', 'd'}
>>> aset <= set('abcd')
True
>>> aset > set('abcd')
False
```



# 什么时候用集合？

## › 快速去除重复的数据项

154, 233, 432, 154, 236, 213, 342,  
432, 236, 332, 321, 243, 319, 326,  
324, 321, 163, 183, 235,...

## › 判断元素是否在一组数据中，如果这些数据的次序不重要，使用集合可以获得比列表更好的性能

如，325是否在上述的一组数据中？