

# Python3 集合

集合 ( set ) 是一个无序的不重复元素序列。

可以使用大括号 { } 或者 set() 函数创建集合，注意：创建一个空集合必须用 set() 而不是 { }，因为 { } 是用来创建一个空字典。

创建格式：

```
parame = {value01,value02,...}  
或者  
set(value)
```

## 实例(Python 3.0+)

```
>>>basket = {'apple', 'orange', 'apple', 'pear', 'orange', 'banana'}  
>>> print(basket) # 这里演示的是去重功能  
{'orange', 'banana', 'pear', 'apple'}  
>>> 'orange' in basket # 快速判断元素是否在集合内  
True  
>>> 'crabgrass' in basket  
False  
>>> # 下面展示两个集合间的运算。  
...  
>>> a = set('abracadabra')  
>>> b = set('alacazam')  
>>> a  
{'a', 'r', 'b', 'c', 'd'}  
>>> a - b # 集合a中包含而集合b中不包含的元素  
{'r', 'd', 'b'}  
>>> a | b # 集合a或b中包含的所有元素  
{'a', 'c', 'r', 'd', 'b', 'm', 'z', 'l'}  
>>> a & b # 集合a和b中都包含了的元素  
{'a', 'c'}  
>>> a ^ b # 不同时包含于a和b的元素  
{'r', 'd', 'b', 'm', 'z', 'l'}
```

类似列表推导式，同样集合支持集合推导式(Set comprehension):

## 实例(Python 3.0+)

```
>>>a = {x for x in 'abracadabra' if x not in 'abc'}  
>>> a  
{'r', 'd'}
```

## 集合的基本操作

### 1、添加元素

语法格式如下：

```
s.add( x )
```

将元素 x 添加到集合 s 中，如果元素已存在，则不进行任何操作。

### 实例(Python 3.0+)

```
>>>thisset = set(("Google", "Runoob", "Taobao"))
>>> thisset.add("Facebook")
>>> print(thisset)
{'Taobao', 'Facebook', 'Google', 'Runoob'}
```

还有一个方法，也可以添加元素，且参数可以是列表，元组，字典等，语法格式如下：

```
s.update( x )
```

x 可以有多个，用逗号分开。

### 实例(Python 3.0+)

```
>>>thisset = set(("Google", "Runoob", "Taobao"))
>>> thisset.update({1,3})
>>> print(thisset)
{1, 3, 'Google', 'Taobao', 'Runoob'}
>>> thisset.update([1,4],[5,6])
>>> print(thisset)
{1, 3, 4, 5, 6, 'Google', 'Taobao', 'Runoob'}
>>>
```

## 2、移除元素

语法格式如下：

```
s.remove( x )
```

将元素 x 从集合 s 中移除，如果元素不存在，则会发生错误。

### 实例(Python 3.0+)

```
>>>thisset = set(("Google", "Runoob", "Taobao"))
>>> thisset.remove("Taobao")
>>> print(thisset)
{'Google', 'Runoob'}
>>> thisset.remove("Facebook") # 不存在会发生错误
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
KeyError: 'Facebook'
>>>
```

此外还有一个方法也是移除集合中的元素，且如果元素不存在，不会发生错误。格式如下所示：

```
s.discard( x )
```

### 实例(Python 3.0+)

```
>>>thisset = set(("Google", "Runoob", "Taobao"))
>>> thisset.discard("Facebook") # 不存在不会发生错误
>>> print(thisset)
{'Taobao', 'Google', 'Runoob'}
```

我们也可以设置随机删除集合中的一个元素，语法格式如下：

```
s.pop()
```

### 脚本模式实例(Python 3.0+)

```
thisset = set(("Google", "Runoob", "Taobao", "Facebook"))
x = thisset.pop()
print(x)
```

输出结果：

```
$ python3 test.py
Runoob
```

多次执行测试结果都不一样。

然而在交互模式，pop 是删除集合的第一个元素（排序后的集合的第一个元素）。

### 实例(Python 3.0+)

```
>>>thisset = set(("Google", "Runoob", "Taobao", "Facebook"))
>>> thisset.pop()
'Facebook'
>>> print(thisset)
{'Google', 'Taobao', 'Runoob'}
>>>
```

## 3、计算集合元素个数

语法格式如下：

```
len(s)
```

计算集合 s 元素个数。

### 实例(Python 3.0+)

```
>>>thisset = set(("Google", "Runoob", "Taobao"))
>>> len(thisset)
```

## 4、清空集合

语法格式如下：

```
s.clear()
```

清空集合 s。

### 实例(Python 3.0+)

```
>>>thisset = set(("Google", "Runoob", "Taobao"))
>>> thisset.clear()
>>> print(thisset)
set()
```

## 4、判断元素是否在集合中存在

语法格式如下：

```
x in s
```

判断元素 x 是否在集合 s 中，存在返回 True，不存在返回 False。

### 实例(Python 3.0+)

```
>>>thisset = set(("Google", "Runoob", "Taobao"))
>>> "Runoob" in thisset
True
>>> "Facebook" in thisset
False
>>>
```

## 集合内置方法完整列表

方法	描述
<a href="#">add()</a>	为集合添加元素
<a href="#">clear()</a>	移除集合中的所有元素
<a href="#">copy()</a>	拷贝一个集合
<a href="#">difference()</a>	返回多个集合的差集
<a href="#">difference_update()</a>	移除集合中的元素，该元素在指定的集合也存在。
<a href="#">discard()</a>	删除集合中指定的元素
<a href="#">intersection()</a>	返回集合的交集

<code>intersection_update()</code>	删除集合中的元素，该元素在指定的集合中不存在。
<code>isdisjoint()</code>	判断两个集合是否包含相同的元素，如果没有返回 True，否则返回 False。
<code>issubset()</code>	判断指定集合是否为该方法参数集合的子集。
<code>issuperset()</code>	判断该方法的参数集合是否为指定集合的子集
<code>pop()</code>	随机移除元素
<code>remove()</code>	移除指定元素
<code>symmetric_difference()</code>	返回两个集合中不重复的元素集合。
<code>symmetric_difference_update()</code>	移除当前集合中在另外一个指定集合相同的元素，并将另外一个指定集合中不同的元素插入到当前集合中。
<code>union()</code>	返回两个集合的并集
<code>update()</code>	给集合添加元素

← Python MongoDB 排序

Python3 zip() 函数 →



3 篇笔记

 写笔记