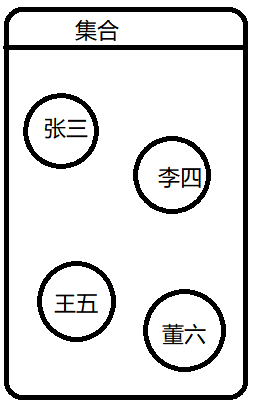
[1.集合](#header-n90)  
 [1.1创建可变集合](#header-n97)  
 [1.2修改可变集合](#header-n102)  
 [1.3遍历集合](#header-n118)

# 1.集合

集合（set）是一种可迭代的、无序的、不能包含重复元素的数据结构。图中是一个班级的集合，其中包含一些学生，这些学生是无序的，不能通过序号访问，而且不能重复。

**提示**：与序列比较，序列中的元素是有序的，可以**重复出现**，而且集合中的元素是无序的，且不能有重复的元素。

**序列强调的是有序，集合强调的是不重复**，而且当没有重复的元素时，序列和集合可以互相替换。

集合又分为**可变集合**和**不可变集合**

## 1.1创建可变集合

可变集合类型是set，创建可变集合可以使用set([iterable])函数，或者用大括号{ }将元素括起来，元素之间用逗号分隔

Python Shell实例：

>>> a = {'张三', '李四', '王五'}  
>>> a  
{'张三', '李四', '王五'}  
>>> a = {'张三', '李四', '王五', '王五'}  
>>> a  
{'张三', '李四', '王五'}  
>>> #集合中如果有重复元素，创建时会自动删除重复元素↑  
...   
>>> len(a)  
3  
>>> b = { }  
>>> type(a)  
<class 'set'>  
>>> type(b)  
<class 'dict'>

空的集合会变成字典，代码中b不是集合，是字典dict，如果要创建空集合，要使用set()函数。

## 1.2修改可变集合

可变集合类似于列表，可变集合的内容可被修改，可以**插入**和**删除**元素。修改可变集合有几个常用的方法。

* add(elem)：添加元素，如果元素已经存在，则不能添加，不会抛出错误
* remove(elem)：删除元素，如果元素不存在，则会抛出错误
* discard(elem)：删除元素，如果元素不存在，不会抛出错误
* pop()：删除返回集合中任意一个元素，返回值是删除的元素
* clear()：清空

Python Shell实例：

>>> student\_set = {'张三', '李四', '王五'}  
>>> student\_set.add('董六') # 随机添加元素，因为集合没有顺序  
>>> student\_set  
{'张三', '董六', '李四', '王五'}  
>>> student\_set.remove('董六')  
>>> student\_set  
{'张三', '李四', '王五'}  
>>> student\_set.remove('董六')  
Traceback (most recent call last):  
 File "<stdin>", line 1, in <module>  
KeyError: '董六'  
>>> #报错，没有董六  
...   
>>> student\_set.discard('董六')  
>>> student\_set  
{'张三', '李四', '王五'}  
>>> #discard()不会抛出错误  
...   
>>> student\_set.pop()  
'张三'  
>>> student\_set  
{'李四', '王五'}  
>>> student\_set.pop()  
'李四'  
>>> student\_set  
{'王五'}  
>>> student\_set.clear()  
>>> student\_set  
set()

## 1.3遍历集合

集合是无序的，没有索引，不能通过下标访问。但可以遍历集合，访问集合的每一个元素。

实例代码：

#coding = utf-8
  
   
student\_set = {'张三', '李四', '王五'}
  
   
for item in student\_set:
  
 print(item)
  
   
print('-----------')
  
for i, item in enumerate(student\_set):
  
 print('{0} - {1}'.format(i, item))

输出结果

张三
  
李四
  
王五
  
----------
  
0 - 张三
  
1 - 李四
  
2 - 王五