

1.集合

1.1创建可变集合

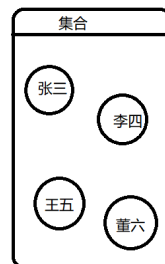
1.2修改可变集合

1.3遍历集合

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：

```
1  >>> a = {'张三', '李四', '王五'}
2  >>> a
3  {'张三', '李四', '王五'}
4  >>> a = {'张三', '李四', '王五', '王五'}
5  >>> a
6  {'张三', '李四', '王五'}
7  >>> #集合中如果有重复元素，创建时会自动删除重复元素†
8  ...
9  >>> len(a)
10 3
11 >>> b = { }
12 >>> type(a)
13 <class 'set'>
14 >>> type(b)
15 <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：

```
1  >>> student_set = {'张三', '李四', '王五'}
2  >>> student_set.add('董六')           # 随机添加元素，因为集合没有顺序
3  >>> student_set
4  {'张三', '董六', '李四', '王五'}
5  >>> student_set.remove('董六')
6  >>> student_set
7  {'张三', '李四', '王五'}
8  >>> student_set.remove('董六')
9  Traceback (most recent call last):
10     File "<stdin>", line 1, in <module>
11  KeyError: '董六'
12  >>> #报错，没有董六
13  ...
14  >>> student_set.discard('董六')
15  >>> student_set
16  {'张三', '李四', '王五'}
17  >>> #discard()不会抛出错误
18  ...
19  >>> student_set.pop()
20  '张三'
21  >>> student_set
22  {'李四', '王五'}
23  >>> student_set.pop()
24  '李四'
25  >>> student_set
26  {'王五'}
27  >>> student_set.clear()
28  >>> student_set
29  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：

```
1  #coding = utf-8
2
3  student_set = {'张三', '李四', '王五'}
4
5  for item in student_set:
6      print(item)
7
8  print('-----')
9  for i, item in enumerate(student_set):
10     print('{0} - {1}'.format(i, item))
```

输出结果

```
1  张三
2  李四
3  王五
4  -----
5  0 - 张三
6  1 - 李四
7  2 - 王五
```