

# 人工智能实践教程

## 从Python入门到机器学习

- 所有代码及ppt均可以由以下链接下载
- <https://github.com/shao1chuan/pythonbook>
- <https://gitee.com/shao1chuan/pythonbook>

# 目录

CONTENT  
S

1. 集合
2. 集合应用案例
3. 字典
4. 字典应用案例
5. defaultdict
6. 内置数据结构总结

集合（**set**）是一个无序的不重复元素的集体。

创建:

- 1). 使用大括号 {} 或者 **set()** 函数创建集合;
- 2). **注意:** 创建一个空集合必须用 **set()** 而不是 {}  
{} 是用来创建一个空字典。

# 01

## 集合

看看是否都可以成功定义集合？

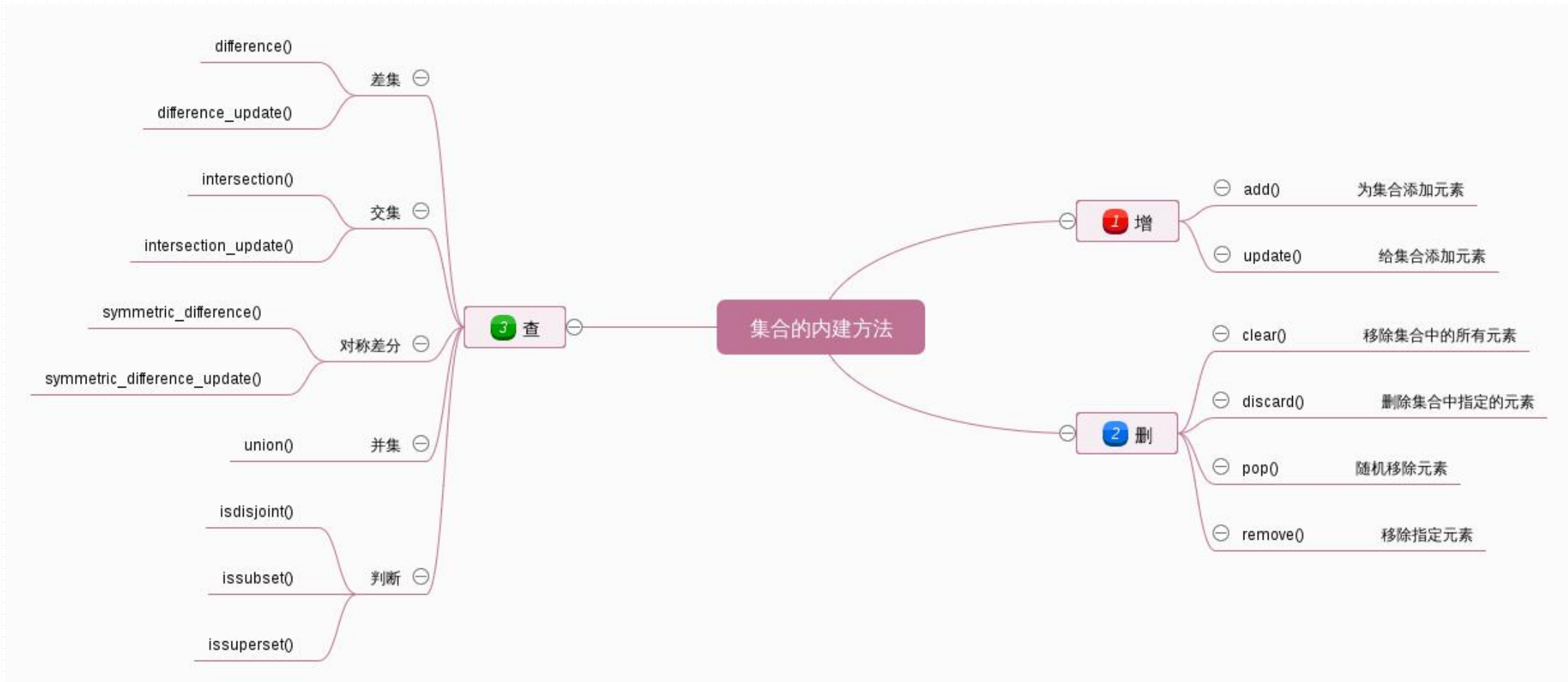
```
s1 = {}
```

```
s2 = {1,2,3}
```

```
s3 = {1,2,3,'hello',(1,2,3),[1,2,3]}
```

```
basket = {'apple', 'orange', 'apple', 'pear', 'orange', 'banana'}
```

```
s4 = set('abracadabra')
```



在抓取页面图片时，为避免重复抓取，将抓取的img结果（结果集是list类型的）通过集合去重。

方法一：依次遍历并判断

```
ids = [1,2,3,3,4,2,3,4,5,6,1]
news_ids = []
for id in ids:
    if id not in news_ids:
        news_ids.append(id)
print news_ids
```

在抓取页面图片时，为避免重复抓取，将抓取的结果（结果集是list类型的）通过集合去重。

方法二：通过set方法进行处理

```
ids = [1,4,3,3,4,2,3,4,5,6,1]
ids = list(set(ids))
```

明明想在学校中请一些同学一起做一项问卷调查，为了实验的客观性，他先用计算机生成了 $N$ 个1到1000之间的随机整数（ $N \leq 1000$ ），对于其中重复的数字，只保留一个，把其余相同的数去掉，不同的数对应着不同的学生的学号。然后再把这些数从小到大排序，按照排好的顺序去找同学做调查。请你协助明明完成“去重”与“排序”的工作(同一个测试用例里可能会有多组数据，希望大家能正确处理)。







itchat是一个开源的微信个人号接口，使用python调用微信从未如此简单。

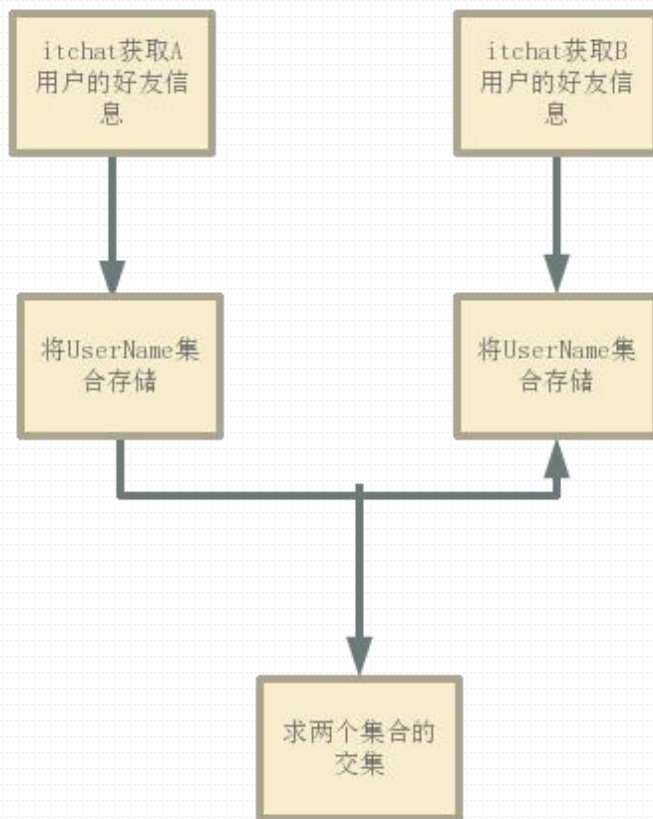
思路：

- 1). 通过itchat微信个人号接口扫码**登录**个人微信网页版，获取可以识别好友身份的数据。此项目需要分别登录两人微信的，拿到两人各自的好友信息**存到列表**中。
- 2). 查共同好友就转化成了**查两个列表中相同元素**的问题。获取到共同好友信息后，可以通过命令行窗口**print**出来，也可以**写入txt文件**



```
"Uin": 0,  
"UserName": 用户名称, 一个"@为好友, 两个"@为群组  
"NickName": 昵称  
"HeadImgUrl": 头像图片链接地址  
"ContactFlag": 1-好友, 2-群组, 3-公众号  
"MemberCount": 成员数量, 只有在群组信息中才有效,  
"MemberList": 成员列表,  
"RemarkName": 备注名称  
"HideInputBarFlag": 0,  
"Sex": 性别, 0-未设置(公众号、保密), 1-男, 2-女  
"Signature": 公众号的功能介绍 or 好友的个性签名  
"VerifyFlag": 0,  
"OwnerUin": 0,
```

```
"PYInitial": 用户名拼音缩写  
"PYQuanPin": 用户名拼音全拼  
"RemarkPYInitial": 备注拼音缩写  
"RemarkPYQuanPin": 备注拼音全拼  
"StarFriend": 是否为星标朋友 0-否 1-是  
"AppAccountFlag": 0,  
"Statues": 0,  
"AttrStatus": 119911,  
"Province": 省  
"City": 市  
"Alias":  
"SnsFlag": 17
```





```
import itchat

def get_list():
    # 在命令行生成登录二维码
    #itchat.auto_login(enableCmdQR=True)

    # 获取登录二维码图片, 扫码登录微信网页版
    itchat.auto_login()
    # 获取好友信息列表
    friendList = itchat.get_friends(update=True)[1:]

    # 每个登录的微信号生成一个好友信息列表
    contactlist = set()
    for i in friendList:
        # 将该好友添加到列表中
        contactlist.add(i['NickName'])

    # 登出微信号
    itchat.logout()
    # 返回该微信号好友信息列表
    return contactlist

#获取第一位扫码登录微信号的好友信息列表
contactlist1 = get_list()

#获取第二位扫码登录微信号的好友信息列表
contactlist2 = get_list()

#共同好友计数
print(len(contactlist1 & contactlist2 ))
```

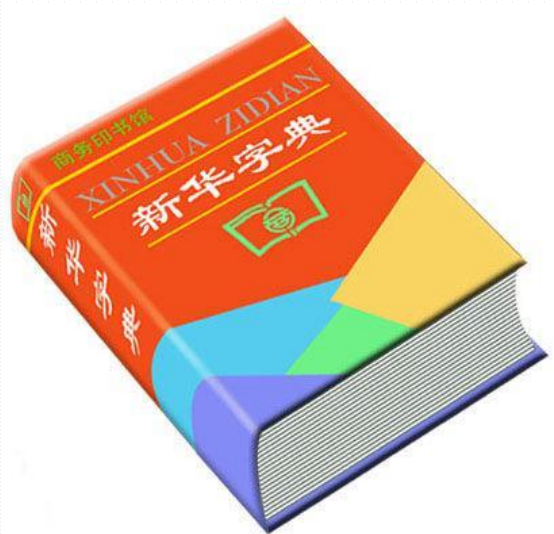


提升版： 想要获取共同好友的更多信息, 该如何解决呢?

```
Login successfully as import *  
LOG OUT!  
你们共有851位共同好友(若除性别外其它信息为空, 不在此记录), 他们分别是:  
好友昵称: 第五松, 陕西咸阳, 男  
好友昵称: Yichen, 陕西西安, 男  
好友昵称: 王雪艳, 陕西咸阳, 女
```

字典是另一种可变容器模型，且可存储任意类型对象。

键一般是唯一的，如果重复最后的一个键值对会替换前面的，值不需要唯一。



```
d = {key1 : value1, key2 : value2 }
```

```
d = {'Z' : '字', 'D' : '典' }
```

## 1). 简单字典创建

```
In [13]: dic = {"name":"fentiao", "age":5, "gender":"male"}
In [14]: dic['name']
Out[14]: 'fentiao'
In [15]: dic[0]
-----
KeyError                                Traceback (most recent call last)
<ipython-input-15-c3244007d783> in <module>()
----> 1 dic[0]
KeyError: 0
```

## 2). 内建方法:fromkeys

字典中的key有相同的value值，默认为None

```
In [16]: ddict = {}.fromkeys(['username','password'],'fentiao')
```

```
In [17]: ddict
```

```
Out[17]: {'password': 'fentiao', 'username': 'fentiao'}
```

```
In [18]: ddict = {}.fromkeys(['username','password'],)
```

```
In [19]: ddict
```

```
Out[19]: {'password': None, 'username': None}
```

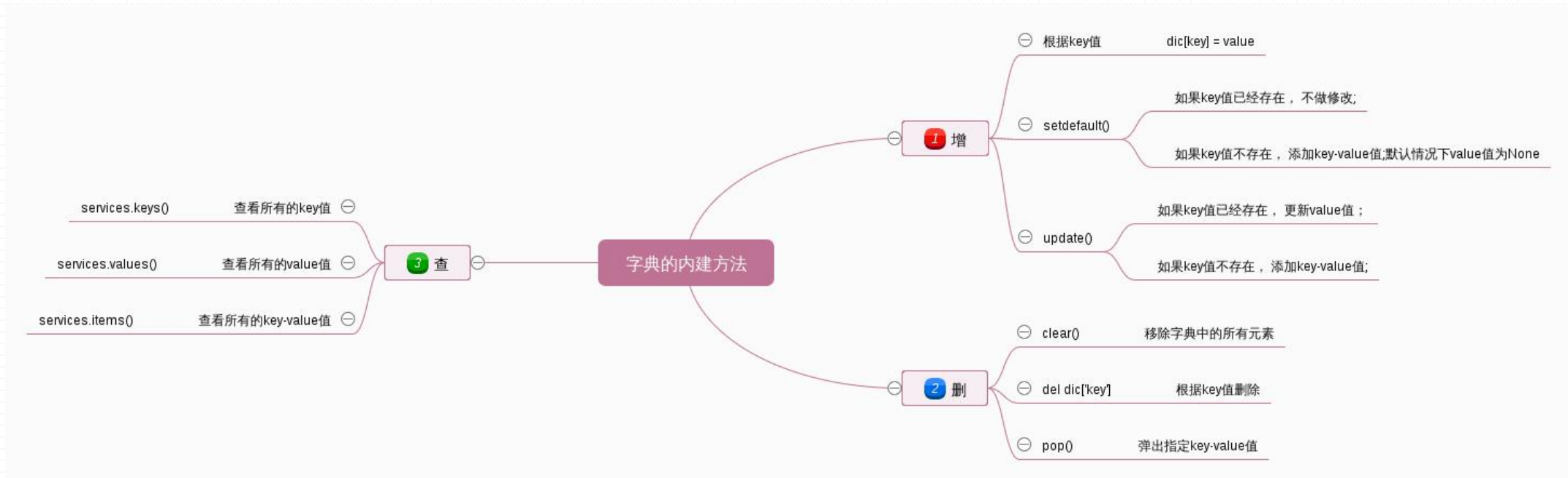


### 3). zip间接创建

```
>>> userInfo = zip(["name", "age"], ["fentiao", 10])  
>>> dict(userInfo)  
{'name': 'fentiao', 'age': 10}
```

### 3). zip间接创建

```
>>> userInfo = zip(["name", "age"], ["fentiao", 10])  
>>> dict(userInfo)  
{'name': 'fentiao', 'age': 10}
```





方法三: 通过字典的方式去重:, 因为字典的key值是不能重复的.

```
li = [1, 2, 3, 4, 65, 1, 2, 3]
```

```
print({}.fromkeys(li).keys())
```

```
void main()
{
int roll = 3 ;
switch ( roll )
{
case 1 :
printf("I am Pankaj");
break;
case 2 :
printf("I am Nikhil");
break;
case 3 :
printf("I am John");
break;
default :
printf("No student found");
break;
}
}
```

**注意：** python中没有switch语句， 如何间接实现？

```
grade = 'B'
if grade == 'A':
    print("优秀")
elif grade == 'B':
    print("良好")
elif grade == 'C':
    print("合格")
else:
    print('无效的成绩')
```

老办法

```
grade = 'D'
d = {
    'A': '优秀',
    'B': '良好',
    'C': "及格"
}
# if grade in d:
#     print(d[grade])
# else:
#     print("无效的成绩")

print(d.get(grade, "无效的成绩"))
```

新办法

python里面不支持switch语句;

C/C++/Java/Javascript:switch语句是用来简化if语句的.

**题目需求：** 假设已有若干用户名字及其喜欢的电影清单，现有某用户，已看过并喜欢一些电影，现在想找个新电影看看，又不知道看什么好。根据已有数据，查找与该用户爱好最相似的用户，也就是看过并喜欢的电影与该用户最接近，然后从那个用户喜欢的电影中选取一个当前用户还没看过的电影，进行推荐。

**技能要点：** Python内置函数以及内置字典与集合的用法

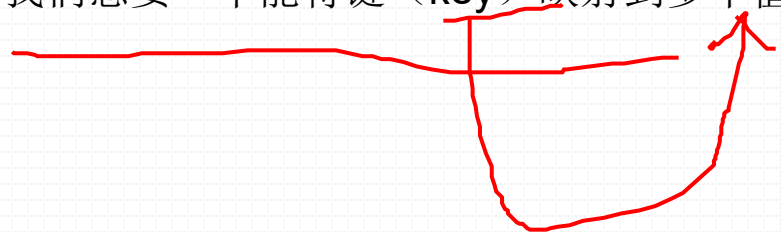


## 一键多值字典: *defaultdict*

collections.defaultdict类，本身提供了默认值的功能，默认值可以是整形，列表，集合等。

需求:

我们想要一个能将键 (**key**) 映射到多个值的字典 (即所谓的一键多值字典)



# 一键多值字典: defaultdict

## 解决方案:

- 1). 字典是一种关联容器，每个键都映射到一个单独的值上。如果想让键映射到多个值，需要将这些多个值保存到容器（列表或者集合）中。
- 2). 利用collections模块中的defaultdict类自动初始化第一个值，这样只需关注添加元素。

```

from collections import defaultdict ✓
d=defaultdict(list)
d['a'].append(1)
d['a'].append(2)
d['b'].append(4)

print(d)

d=defaultdict(set)
d['a'].add(1)
d['a'].add(2)
d['b'].add(4)

print(d)

```

Handwritten notes in red:

- $d['a'] = []$  with an arrow pointing to the `list` argument in the first code block.
- `.append()` written next to the arrow.
- `'a': [1, 2]` and `'b': [4]` written below the first code block.

## 总结

## 解决方案：

- 1). 字典是一种关联容器，每个键都映射到一个单独的值上。如果想让键映射到多个值，需要将这些多个值保存到容器（列表或者集合）中。
- 2). 利用collections模块中的defaultdict类自动初始化第一个值，这样只需关注添加元素。

```
from collections import defaultdict

d=defaultdict(list)
d['a'].append(1)
d['a'].append(2)
d['b'].append(4)

print(d)

d=defaultdict(set)
d['a'].add(1)
d['a'].add(2)
d['b'].add(4)

print(d)
```

**可变数据类型:可以增删改。** 可变数据类型, 允许变量的值发生变化, 即如果对变量进行`append`、`+=`等这种操作后, 只是改变了变量的值, 而不会新建一个对象, 变量引用的对象的地址也不会变化, 不过对于相同的值的不同对象, 在内存中则会存在不同的对象, 即每个对象都有自己的地址, 相当于内存中对于同值的对象保存了多份, 这里不存在引用计数, 是实实在在的对象。

**不可变数据类型:不可以增删改。** python中的不可变数据类型, 不允许变量的值发生变化, 如果改变了变量的值, 相当于是新建了一个对象, 而对于相同的值的对象, 在内存中则只有一个对象, 内部会有一个引用计数来记录有多少个变量引用这个对象。

**序列:** Python包含列表、元组、字符串、集合等内建的序列。所有序列类型都可以进行某些特定的操作。

有序序列: 这些操作包括: 索引 (**indexing**)、分片 (**slicing**)、连接操作符 (**adding**)、重复操作符 (**multiplying**) 以及成员操作符。

**非序列:**

# 06

## 内置数据结构总结

可以for循环:

不可以for循环:

**问题描述:**

有一个列表，其中包括 10 个元素，例如这个列表是[1,2,3,4,5,6,7,8,9,0],要求将列表中的每个元素一次向前移动一个位置，第一个元素到列表的最后，然后输出这个列表。最终样式是[2,3,4,5,6,7,8,9,0,1]

**问题描述：** 按照下面的要求实现对列表的操作：

- 1). 产生一个列表，其中有 40 个元素，每个元素是 0 到 100 的一个随机整数
- 2). 如果这个列表中的数据代表着某个班级 40 人的分数，请计算成绩低于平均分的学生人数
- 3). 对上面的列表元素从大到小排序并输出



**问题描述:**

如果将一句话作为一个字符串，那么这个字符串中必然会有空格（这里仅讨论英文），比如"**How are you.**"，有的时候，会在两个单词之间多出一个空格。现在的任务是，如果一个字符串中有连续的两个空格，请把它删除。

感谢聆听！

**THANK YOU!**