1. python没有char类型，全部都是str类型，且str类型不可被更改
2. 判断输入是不是空的：
3. 判断len(input)，如果是空的话，得到的值为0
4. input.isspace()，如果全部由空格组成，那么返回值为1
5. 判断input.strip()如果全是空格，返回‘’，是0
6. 输入一个数字，num=int(input())，省去一步
7. str.alpha()：判断字符串里是否全由字母组成。

str.isdigit()：判断字符串是否全由数字组成

1. str.islower()：判断是否全由小写字母组成，有其他字符如空格、数字，都不行；

str.isupper()：判断是否全由大写字母组成，有其他字符如空格、数字，都不行；

1. str.upper()：字符串中所有小写字母转换成大写

str.lower()：字符串中所有大写字母转换成小写

str.capitalize()：规范化str，第一个字母大写，**其余字母都小写**

str.title()：规范化单词，第一个字母大写，其余字母小写

1. 字符串的分割：

str.split(‘ ’)，以空格为分割，这样是有坑的，输入两个空格，会返回空字符——‘’，这是不希望看到的

str.split()，当不带任何参数时，是以空白符为分割标志，这才是最常用的操作

1. range(num)返回的是一个可迭代的对象，而不是list，所以，类似range(n).remove(i)的操作都是错误的。

正确的做法是range(a,b)

1. 格式化输出：

float保留小数位数：

print(‘%.2f’% (num))

将num保留两位小数输出

1. 字符与ASCII码的转换

这部分主要时想让python和C语言那样，可以**直接把int和字符做加法**

但是，python无char类型

Return the integer ordinal of a one-character string.

chr(ASCII码)

ord(one-character)



1. 记数：

可迭代对象（Iterable）有count()方法，可以统计从指定位置开始对指定的对象进行计数

list.count(object,start,end)，start和end也不是必须的

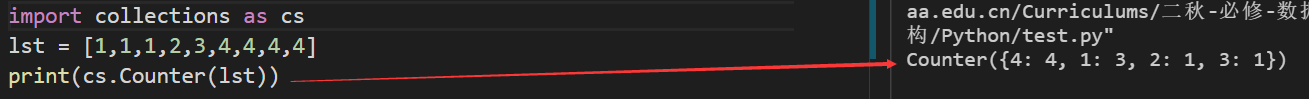
以统计list中元素出现个数并保存到字典中为例：

* 1. 利用collections中的Counter方法：

import collections

dic=dict(collections.Counter(list))

返回的类型是collections.Counter()



* 1. for i in list:

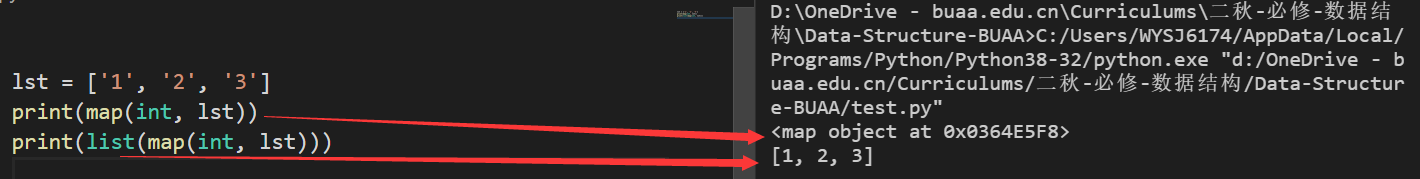
dic[i]=list.count(i)

这样的缺点是会有重复，当然，也可以set(list)去重后再count

1. 把一个str的list变成int的list，原来的列表还是原来的列表：

lst=list(map(int,lst))

如果不加上list函数，返回的是一个map object



1. **排序：**

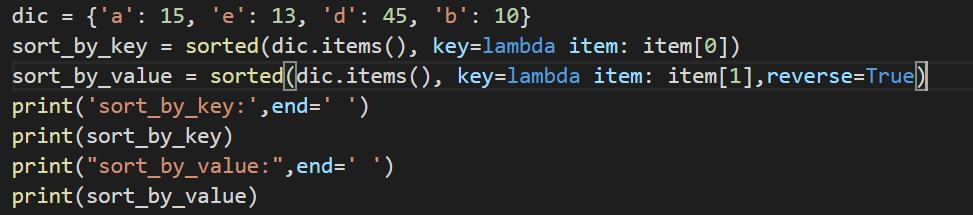
sorted(iterable,key,reverse)，返回列表

iterable表示可以迭代的对象，例如可以是dict.items()、dict.keys()等，

key是一个函数，用来评估参与比较的元素，

reverse则是用来指定排序是倒序还是顺序

e.g.



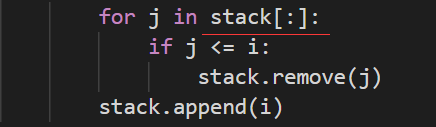


1. list中添加元素到末尾，按顺序添加，list.append(value)
2. list插入元素，list.insert(index,value)value将替代index之位
3. list中删除元素，

list.remove(i)

此处可能有坑，因为删除一个，列表就平移了，同时i也“指向”了后一个元素，所以for i in list: remove(i) 是错的

应当传入一个副本copy

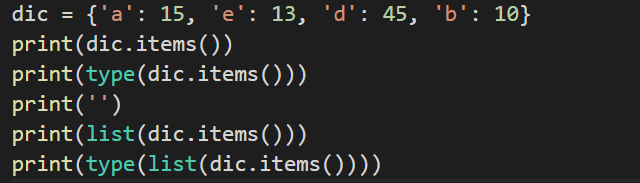


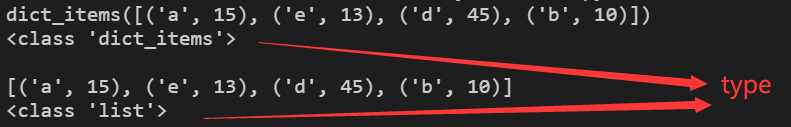
for i in list[:]:

if (condition): list.remove(i)

1. 获取list中元素的索引，list.index(value)，不过只能返回第一个的索引
2. 合并list，使用extend方法：Alist.extend(Blist)
3. dict新建键值对，直接dict[key]=value
4. 遍历键对、值对和键值对“

for k,v in dict.items():





类似的，for key in dict.keys():

1. **lambda函数 匿名函数：**

func=lambda argument\_list: expression

a=lambda x,y:x\*y

这样定义了一个叫做a的函数，a(3,4)=12

lambda函数可以当作一些以函数为参数的函数的参数

1. re模块，正则表达式(用得特别少)

findall()函数

在字符串中找到正则表达式所匹配的所有子串，并返回一个列表，如果没有找到匹配的，则返回空列表

re.findall(pattern, text)

在text里找到所有的pattern，并且保存到一个list里

在pattern中放括号，返回的就是括号里匹配上的字符串

re.findall(“ (a..) ”, text) 🡪[“air”, “are”]

re.findall(“ a.. ”,text) 🡪[“ air ”, “ are ”]

这二者的区别就是，前者过滤了后者的空格

注意“|”的使用，“|”分割了pattern，所以不能因为美观而在|左右加上空格，|左右的是要马上匹配的

1. 全局变量

这个主要是在递归中使用，在用之前加上一句，

global x

就可以了

[list的合并](https://www.cnblogs.com/lifangzheng/p/11379172.html)