一、二者区别

列表：

    1.可以增加列表内容     append

    2.可以统计某个列表段在整个列表中出现的次数 count

    3.可以插入一个字符串，并把整个字符串的每个字母拆分当作一个列表段追加到列表当中 extedn

    4.可以查询某个列表段在整个列表的位置 index

    5.可以在指定位置插入一个列表段 insert

    6.可以删除列表的最后一个列表段 pop

    7.可以删除指定列表中的某个列表段 remove

    8.可以正向反向排序 reverse

    9.可以按字母或数字排序 sort

    10.定义列表时候使用中括号"[]"

注意：在列表当中，假如某两个列表段相同，不管是使用index还是remove都是统计的最靠前的列表段

元组：

    1.可以统计某个元组段在整个元组中出现的次数    count

    2.可以查询某个元组段在整个元组中的元组号    index

    3.定义元组时候使用小括号"()"

二、二者的使用方法

列表

[复制代码](javascript:void(0);)

#定义列表

>>> name\_list = ['sean','tom','jack','Angelia','Daisy','jack']

#查看定义的列表

>>> name\_list

['sean', 'tom', 'jack', 'Angelia', 'Daisy', 'jack']

#增加david列表段

>>> name\_list.append('david')

>>> name\_list

['sean', 'tom', 'jack', 'Angelia', 'Daisy', 'jack', 'david']

#统计david列表段出现次数

>>> name\_list.count('david')

1

>>> name\_list.count('jack')

2

#使用extend向列表中增加列表段

>>> name\_list.extend('Hello,My name is sean')

>>> name\_list

['sean', 'tom', 'jack', 'Angelia', 'Daisy', 'jack', 'david', 'H', 'e', 'l', 'l', 'o', ',', 'M', 'y', ' ', 'n', 'a', 'm', 'e', ' ', 'i', 's', ' ', 's', 'e', 'a', 'n']

#查看列表段所在的索引号，注意这里统计的jack为第一个jack id号

>>> name\_list.index('jack')

2

>>> name\_list.index('tom')

1

#向索引号为2的地方插入Adam

>>> name\_list.insert(2,'Adam')

>>> name\_list

['sean', 'tom', 'Adam', 'jack', 'Angelia', 'Daisy', 'jack', 'david', 'H', 'e', 'l', 'l', 'o', ',', 'M', 'y', ' ', 'n', 'a', 'm', 'e', ' ', 'i', 's', ' ', 's', 'e', 'a', 'n']

#删除最后一个列表段

>>> name\_list.pop()

'n'

>>> name\_list

['sean', 'tom', 'Adam', 'jack', 'Angelia', 'Daisy', 'jack', 'david', 'H', 'e', 'l', 'l', 'o', ',', 'M', 'y', ' ', 'n', 'a', 'm', 'e', ' ', 'i', 's', ' ', 's', 'e', 'a']

#删除指定列表段，注意这里删除的是第一个jack

>>> name\_list.remove('jack')

>>> name\_list

['sean', 'tom', 'Adam', 'Angelia', 'Daisy', 'jack', 'david', 'H', 'e', 'l', 'l', 'o', ',', 'M', 'y', ' ', 'n', 'a', 'm', 'e', ' ', 'i', 's', ' ', 's', 'e', 'a']

#对整个列表进行倒序

>>> name\_list.reverse()

>>> name\_list

['a', 'e', 's', ' ', 's', 'i', ' ', 'e', 'm', 'a', 'n', ' ', 'y', 'M', ',', 'o', 'l', 'l', 'e', 'H', 'david', 'jack', 'Daisy', 'Angelia', 'Adam', 'tom', 'sean']

#对整个列表进行倒序

>>> name\_list.reverse()

>>> name\_list

['sean', 'tom', 'Adam', 'Angelia', 'Daisy', 'jack', 'david', 'H', 'e', 'l', 'l', 'o', ',', 'M', 'y', ' ', 'n', 'a', 'm', 'e', ' ', 'i', 's', ' ', 's', 'e', 'a']

#对整个列表进行列表段的首字母进行排序

>>> name\_list.sort()

>>> name\_list

[' ', ' ', ' ', ',', 'Adam', 'Angelia', 'Daisy', 'H', 'M', 'a', 'a', 'david', 'e', 'e', 'e', 'i', 'jack', 'l', 'l', 'm', 'n', 'o', 's', 's', 'sean', 'tom', 'y']

>>>

[复制代码](javascript:void(0);)

元组

[复制代码](javascript:void(0);)

#定义元组name\_tuple

>>> name\_tuple = ('xiaoming','xiaohong','xiaoli','xiaozhang','xiaoming')

>>> name\_tuple

('xiaoming', 'xiaohong', 'xiaoli', 'xiaozhang', 'xiaoming')

#统计xiaoming、xiaohong在元组内出现的次数

>>> name\_tuple.count('xiaoming')

2

>>> name\_tuple.count('xiaohong')

1

#查询xiaoming、xiaohong、xiaozhang在元组内的id号

>>> name\_tuple.index('xiaoming')

0

>>> name\_tuple.index('xiaohong')

1

>>> name\_tuple.index('xiaozhang')

3

>>>

#尝试增加一个元组单元

>>> name\_tuple.append('xiaowang')

Traceback (most recent call last):

File "<pyshell#49>", line 1, in <module>

name\_tuple.append('xiaowang')

AttributeError: 'tuple' object has no attribute 'append'

>>>

[复制代码](javascript:void(0);)

元组的元素是不可变的，元组的元素的元素是可变的

[复制代码](javascript:void(0);)

>>> tuple\_A = (1,2,{'k1':'v1'})

>>> for i in tuple\_A:

... print i

...

1

2

{'k1': 'v1'}

#更改元素

>>> tuple\_A[2]['k1'] = 'v2'

>>> for i in tuple\_A:

... print i

...

1

2

{'k1': 'v2'}

>>>

[复制代码](javascript:void(0);)

原文出处：

<http://www.cnblogs.com/xinzhiyu/p/5670219.html>