



链滴

Python 学习之列表 (list) 详解

作者: [alongblog](#)

原文链接: <https://ld246.com/article/1516586398359>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

Python中的list就是一个大杂烩 什么都可以放(我们知道的类型)

创建列表只需要用逗号(,)隔开来用方括号([])包含就可以了

Python有6个序列的内置类型,但最常见的是列表和元组。

序列都可以进行的操作包括索引,切片,加,乘,检查成员

列表索引从0开始。列表可以进行截取、组合

使用下标索引来访问列表中的值

list基本知识

[<a>列表创建](#)

[<a>List1 = \[\] 空列表](#)

[<a>List2 = \['Python',3.1415,121\] 混合列表](#)

向列表添加元素

- [end\(\)语法介绍](http://1fx.xin/articles/2018/01/22/1516586397873.html#b3_solo_h3_6) [append\(\)示例](http://1fx.xin/articles/2018/01/22/1516586397873.html#b3_solo_h4_7)
- [extend\(\)语法介绍](http://1fx.xin/articles/2018/01/22/1516586397873.html#b3_solo_h3_6) [extend\(\)示例](http://1fx.xin/articles/2018/01/22/1516586397873.html#b3_solo_h4_8)
- [insert\(\)语法介绍](http://1fx.xin/articles/2018/01/22/1516586397873.html#b3_solo_h3_6) [insert\(\)示例](http://1fx.xin/articles/2018/01/22/1516586397873.html#b3_solo_h4_9)
- [append \(\) 和extend \(\) 区别](http://1fx.xin/articles/2018/01/22/1516586397873.html#b3_solo_h4_10)

从列表删除元素

- [x] [remove\(\)语法介绍](http://1fx.xin/articles/2018/01/22/1516586397873.html#b3_solo_h3_6) [remove\(\)示例](http://1fx.xin/articles/2018/01/22/1516586397873.html#b3_solo_h4_11)
- [x] [pop\(\)语法介绍](http://1fx.xin/articles/2018/01/22/1516586397873.html#b3_solo_h3_6) [pop\(\)示例](http://1fx.xin/articles/2018/01/22/1516586397873.html#b3_solo_h4_12)
- [x] [del语法介绍](http://1fx.xin/articles/2018/01/22/1516586397873.html#b3_solo_h3_6) [del 示例](http://1fx.xin/articles/2018/01/22/1516586397873.html#b3_solo_h4_13)
- [x] [del , remove,pop三者区别](http://1fx.xin/articles/2018/01/22/1516586397873.html#b3_solo_h4_14)

列表分片 (Slice)

语法

- [x] [介绍以及示例讲解点击我](http://1fx.xin/articles/2018/01/22/1516586397873.html#b3_solo_h3_15)
- [x] [sequence\[indexStart : indexEnd : stride\]](#)

Python列表函数

<table> <tr>	
序号</td>	函数</td>
</tr>	
<tr><td>1</td><td><a>cmp(list1, list2) 比较两个列表的元素</td></tr>	
<tr><td>2</td><td><a>len(list) 列表元素个数</td></tr>	
<tr><td>3</td><td><a>max(list) 返回列表元素最大值</td></tr>	
<tr><td>4</td><td><a>min(list) 返回列表元素最小值</td></tr>	
<tr><td>5</td><td><a>list(seq) 将元组转换为列表</td></tr>	
</table>	

Python列表方法

<table> <tr>	
序号</td>	方法</td>
参数</td>	返回值</td>
</tr>	
<tr><td>1</td><td><a>list.append(obj) 在列表末尾添加新的象</td><td>obj -- 列表对象</td><td>该方法无返回值，但是会修改原来的列表</td></tr>	
<tr><td>2</td><td><a>list.count(obj) 统计某个元素在列表中出现的次数</td><td>obj -- 列表对象</td><td></td></tr>	
<tr><td>3</td><td><a>list.extend(seq) 在列表末尾一次性追加另一个序列中的多个值（用新列表扩展原来的列表</td><td></td><td>该方法没有返回值，但会在已存在的列表中添加新的列表内容</td></tr>	
<tr><td>4</td><td><a>list.index(obj) 从列表中找出某个值第一个匹配项的索引位置</td><td>obj -- 列表对象</td><td></td></tr>	
<tr><td>5</td><td><a>list.insert(index, obj) 将对象插入列表</td><td>index -- 要插入的索引位置 obj -- 列表对象</td><td>该方法没有返回值，但会在表指定位置插入对象</td></tr>	
<tr><td>6</td><td><a>list.pop(index=-1) 移除列表中的一个元素（默认最后一个元素），并且返回该元素的值</td><td>index=-1--删除对象list的索引 如果为空默认为-1最后一项</td><td>返回要删除元素的值</td></tr>	
<tr><td>7</td><td><a>list.remove(obj) 移除列表中某个值的一个匹配项</td><td>obj -- 列表对象</td><td>此方法不返回任何值，但从列表中删除给定的对</td></tr>	
<tr><td>8</td><td><a>list.reverse() 反向列表中元素</td><td></td><td></td></tr>	
<tr><td>9</td><td><a>list.sort([func]) 对原列表进行排序</td><td></td><td></td></tr>	
</table>	

append()方法示例

```
list.append(obj)
在尾部添加元素，如下所示：
list = [] ## 创建空列表
list.append('学习List') ## 使用 append() 添加元素
list.append('append示例')
print list
输出结果：
['学习List','append示例']
```

```

#### #### extend()方法示例

```

list.extend(seq)

可以在尾部添加一个或多个对象(元素) 如下所示:

List1 = [along, 'zzz', 'wuwu', 'abc', 222]

List2 = [2018, '1月']

List1.extend(List2)

print ("Extended List : ", List1)

输出结果:

[along,'zzz','wuwu','abc',222,2018,'1月']

insert()方法示例

list.insert(index, obj)

用于将对象(元素)插入列表的指定位置。

List1 = [1314,'Love','you','you love me ?']

List1.insert(1,'I')

print ("Final List:", List1)

输出结果:

Final List: [1314, 'I', 'Love', 'you', 'you love me ?']

append () 和extend () 区别

如果append和extend的参数都是一个的话, 这两功能类似,

但是, 要是参数为多个 extend将它的参数视为 list, 并将该参数的每个元素都添加到原有的列表中

append是将它的参数视为element (元素), 作为一个整体添加上去的。比如在List1中添加List2

List2本身是一个列表(可用type查看) append会把这个列表添加进去

通过实机操作我们发现一个小技巧 append可以将一维列表变成二维列表

其他就不一一测试了

实验append () 和extend () 区别

append()

List1 = [1314,'Love','you','you love me ?']

List2 = ['YES',123]

List1.insert(1,'I')

List1.append(List2)

print("append List:",List1)

输出结果:

append List: [1314,'I' 'Love', 'you', 'you love me ?', ['YES', 123]] ##列表中还有一个列表

extend()

List1 = [1314,'Love','you','you love me ?']

List2 = ['YES',123]

List1.insert(1,'I')

List1.extend(List2)

print ("extend List:", List1)

输出结果:

```
extend List: [1314, 'I', 'Love', 'you', 'you love me ?', 'YES', 123]
```

通过比较我们发现

```
append List: [1314, 'I', 'Love', 'you', 'you love me ?', ['YES', 123]]
```

```
extend List: [1314, 'I', 'Love', 'you', 'you love me ?', 'YES', 123]
```

如果append参数传的是多个 那个append会把TA当作列表加入进去
而extend会把列表的每一个元素依次加入到列表尾部

remove()方法示例

```
list.remove(obj)
```

从列表中删除第一个obj(如果多个元素一样, 默认删除第一个)

```
List1 = [1314, 'I', 'Love', 'you', 'you love me ?']
```

```
List1.remove(1314)
```

```
print ("remove List:", List1)
```

输出结果:

```
remove List: ['I', 'Love', 'you', 'you love me ?']
```

pop()方法示例

```
list.pop(index=-1)
```

用于取出并移除列表中的一个元素 (默认最后一个元素) 还可以pop(索引值)

```
List1 = [1,2,3,4,5,6]
```

```
pop1=List1.pop() #默认删除最后一个元素
```

```
pop2=List1.pop(2) #删除索引为2的元素 (3)
```

```
print ("pop:", pop1) #打印pop后的返回值
```

```
print ("pop1:", pop2) #
```

```
print ("pop List:", List1)
```

输出结果:

```
pop: 6
```

```
pop1: 3
```

```
pop List: [1, 2, 4, 5]
```

del (BIF)

```
del [引用],[对象]
```

del的主要作用是删除引用而不是删除对象 但是有些时候我们也可以用于删除list元素

如果del要删除list元素 他是通过索引进行的

```
List1 = [1314, 'I', 'Love', 'you', 'you love me ?']
```

```
del List1[0] #删除1314 下标是0
```

```
print ("del List:", List1)
```

输出结果:

remove List: ['I', 'Love', 'you', 'you love me ?']

这个就是大概说一下 del的主要作用不在这里 我们后面会讲到

del ,remove,pop三者区别

- > 首先, remove 是删除首个符合条件的元素。并不是删除特定的索引
- > 而对于 del 来说, 它是根据索引(元素所在位置)来删除的
- > 其次pop返回的是你弹出的那个数值 在根据索引(元素所在位置)来删除的
- > 所以使用时要根据你的具体需求选用合适的方法(在一些复杂的计算里 很容易出错)

本章篇幅有限 点击我
续学习Python碎片知识