



链滴

# Python 列表和元组 - 学习笔记 (持续更新)

作者: [angels](#)

原文链接: <https://ld246.com/article/1502339511324>

来源网站: 链滴

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

<h3 id="注意事项-Python内建了6种序列-列表和元组为其中常用的两种-">注意事项 (Python 内  
了 6 种序列, 列表和元组为其中常用的两种) </h3>

<ul>

<li>列表可修改, 元组不可修改</li>

<li>几乎所有情况下都可以用列表来代替元组</li>

<li>在创建字典时, key 的存储之所以使用元组而不使用列表, 是因为列表可修改, 元组不可修改</li>

</ul>

<h3 id="创建形式">创建形式</h3>

<ul>

<li>列表里可存储任意类型的值<br>

1、lis = ['value1','value2',value3]</li>

<li>序列可以包含序列<br>

2、list = ['value1',[value2,'value3']]</li>

</ul>

<h3 id="序列的通用操作">序列的通用操作</h3>

<ul>

<li>

<p>所有的序列类型都可以进行某些特定的操作</p>

</li>

<li>

<p>1、索引-indexing<br>

序列中的所有元素都是有编号的 (从 0 开始递增), 也就是我们的索引号, 可以通过编号进行访问</p>

<pre><code class="language-mysql highlight-chroma"><span class="highlight-line"><span class="highlight-cl"><span class="highlight-w"> </span><span class="highlight-err">如:  
/span><span class="highlight-n">str</span><span class="highlight-w"> </span><span class="highlight-o">=</span><span class="highlight-w"> </span><span class="highlight-s2">  
hello</span><span class="highlight-w">

</span></span></span><span class="highlight-line"><span class="highlight-cl"><span class="highlight-w"> </span><span class="highlight-err">获取</span><span class="highlight-  
>`</span><span class="highlight-n">o</span><span class="highlight-o">`</span><span class="highlight-err">:</span><span class="highlight-n">str</span><span class="highligh  
-p">[</span><span class="highlight-mi">4</span><span class="highlight-p">]</span><span class="highlight-w">

</span></span></span><span class="highlight-line"><span class="highlight-cl"><span class="highlight-w"> </span><span class="highlight-err">我们也可以倒着获取</span><span class="highlight-  
s="highlight-p">:</span><span class="highlight-n">str</span><span class="highlight-p"></span><span class="highlight-o">-</span><span class="highlight-mi">1</span><span class="highlight-p">]</span><span class="highlight-w">

</span></span></span></code></pre>

</li>

<li>

<p>2、分片-slicing<br>

分片就是提取序列的一部分, 跟通过索引号访问字符串中的某个字符类型, 我们这里是获取某个范围  
字符串</p>

<pre><code class="language-mysql highlight-chroma"><span class="highlight-line"><span class="highlight-cl"><span class="highlight-w"> </span><span class="highlight-err">如:  
/span><span class="highlight-n">str</span><span class="highlight-o">=</span><span class="highlight-s2">"hello"</span><span class="highlight-w">

</span></span></span><span class="highlight-line"><span class="highlight-cl"><span class="highlight-w"> </span><span class="highlight-err">获取</span><span class="highlight-  
>`</span><span class="highlight-n">ll</span><span class="highlight-o">`</span><span class="highlight-err">,<span class="highlight-err">当然你可以这样:</span><span class="highlight-n">str</span><span class="highlight-p">

```

class="highlight-p">[/span><span class="highlight-mi">2[/span><span class="highlight-
">[/span><span class="highlight-o">+[/span><span class="highlight-n">str[/span><sp
n class="highlight-p">[/span><span class="highlight-mi">3[/span><span class="highligh
-p">[/span><span class="highlight-err">或者[/span><span class="highlight-n">str[/spa
><span class="highlight-p">[/span><span class="highlight-o">-[/span><span class="high
light-mi">3[/span><span class="highlight-p">[/span><span class="highlight-o">+[/spa
><span class="highlight-n">str[/span><span class="highlight-p">[/span><span class="h
ghlight-o">-[/span><span class="highlight-mi">2[/span><span class="highlight-p">[/sp
n><span class="highlight-w">
[/span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w">[/span><span class="highlight-err">分片获取:  [/span><span class="highl
ight-n">str[/span><span class="highlight-p">[/span><span class="highlight-mi">2[/spa
><span class="highlight-p">:[/span><span class="highlight-mi">4[/span><span class="hi
hlight-p">],[/span><span class="highlight-err">也就是从索引号[/span><span class="highli
ht-mi">2[/span><span class="highlight-err">开始取, 一直取到索引号为[/span><span class
"highlight-mi">4[/span><span class="highlight-err">之前的元素[/span><span class="highl
ght-w">
[/span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w">[/span><span class="highlight-err">也可以倒着来哦:  [/span><span clas
="highlight-n">str[/span><span class="highlight-p">[/span><span class="highlight-o">-
[/span><span class="highlight-mi">3[/span><span class="highlight-p">:[/span><span class
="highlight-o">-[/span><span class="highlight-mi">1[/span><span class="highlight-p">]
[/span><span class="highlight-err">注意取的顺序还是要从左到右哦[/span><span class="high
light-w">
[/span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w">[/span><span class="highlight-err">如果要取[/span><span class="highli
ht-o">`[/span><span class="highlight-n">llo[/span><span class="highlight-o">`[/span><
pan class="highlight-p">:[/span><span class="highlight-n">str[/span><span class="highli
ht-p">[/span><span class="highlight-mi">2[/span><span class="highlight-p">:[/span><
pan class="highlight-mi">5[/span><span class="highlight-p">[/span><span class="highli
ht-w">
[/span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w">[/span><span class="highlight-err">如果一直取到末尾, 我们也可以这样哦
[/span><span class="highlight-n">str[/span><span class="highlight-p">[/span><span cl
ss="highlight-mi">2[/span><span class="highlight-p">:]/[/span><span class="highlight-err
>、[/span><span class="highlight-n">str[/span><span class="highlight-p">[/span><span
class="highlight-o">-[/span><span class="highlight-mi">3[/span><span class="highlight
p">:]/[/span><span class="highlight-w">
[/span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w">[/span><span class="highlight-err">如果取整个序列:  [/span><span clas
="highlight-n">str[/span><span class="highlight-p">[:],[/span><span class="highlight-err
>没错, 都用空的就可以了哦[/span><span class="highlight-w">
[/span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w">[/span><span class="highlight-err">总结: 用数学开闭区间来理解[/span>
span class="highlight-w">[/span><span class="highlight-p">[/span><span class="highli
ht-n">oneIndex[/span><span class="highlight-p">:[/span><span class="highlight-n">two
ndex[/span><span class="highlight-p">)/[/span><span class="highlight-w">
[/span></span></span></span></code></pre>
</li>
<li>
<p>分片除了设置起始和终止索引号外, 我们还可以设置步长哦, 其实之前也是有步长的, 默认的步骤为 1</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl"> 同样, 我们取str= "hello" 重的ll: str[2:4:1]

```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> 我要取e和o呢?
</span></span><span class="highlight-line"><span class="highlight-cl"> 当然你可以这样str
1]+str[4]或者str[-4]+str[-1]
</span></span><span class="highlight-line"><span class="highlight-cl"> 但是我们同样可
用分片: str[1::2],也就是取一个元素后隔两个再取
</span></span></code></pre>
```

</li>

<li>

<p>3、序列相加-adding<br>

Python 中的序列可以用 <code>+</code> 直接相加哦，但是要注意，类型必须要一致哦</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl"> lst1 = [1,2,3];
</span></span><span class="highlight-line"><span class="highlight-cl"> lst2 = [4,5,6];
</span></span><span class="highlight-line"><span class="highlight-cl"> lst3 = lst1 + lst2;
</span></span><span class="highlight-line"><span class="highlight-cl"> 那么lst此时为[1,2,
,4,5,6]
</span></span></code></pre>
```

</li>

<li>

<p>4、序列相乘-multiplying<br>

Python 中用：新的序列 = 数字\*序列</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl"> lst1 = [1];
</span></span><span class="highlight-line"><span class="highlight-cl"> lst2 = lst1*5;
</span></span><span class="highlight-line"><span class="highlight-cl"> lst2:[1,1,1,1,1];
</span></span></code></pre>
```

</li>

<li>

<p>5、检查某个元素是否属于序列<br>

Python 中判断某个值是否在序列中很简单，使用 <code>in</code> 运算符即可解决。返回的是 boolean 值，true 为存在。false 不存在</p>

</li>

<li>

<p>6、计算序列长度</p>

</li>

<li>

<p>7、找出最大最小元素</p>

</li>

</ul>

## <ul> <li>1、list 函数<br> list 函数适用于所有类型的序列<br> 字符串是不可变的，但是我们可以用字符串来创建字符串列表，这样就可以使用列表的可变能力来实现我们的目的了：lst = list("hello") lst 为['h','e','l','l','o'];</li> </ul> <ul> <li> <p>1、元素赋值</p> ``` <pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl"> lst=[1,2,3]; </span></span><span class="highlight-line"><span class="highlight-cl"> lst[1] = 4; </span></span><span class="highlight-line"><span class="highlight-cl"> lst : [1,4,3] </span></span></code></pre> ``` 原文链接: [Python 列表和元组 - 学习笔记 \(持续更新\)](#)

</span></span><span class="highlight-line"><span class="highlight-cl"> 注：不能给不存  
的位置赋值，如lst[3]=5就会报错了

```
</span></span></code></pre>
```

</li>

<li>

<p>2、删除元素<br>

删除元素我们使用 del 语句实现</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl"> lst=[1,2,3];
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> del lst[1];
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> lst: [1,3]
```

```
</span></span></code></pre>
```

</li>

<li>

<p>3、分片赋值<br>

我们之前讲了分片读值，我们也可以分片赋值哦</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl"> str = list("hello");
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> str[2::2]=list('xx');
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> str: ['h', 'e', 'x', 'l', 'x']
```

</span></span><span class="highlight-line"><span class="highlight-cl"> 分片赋值如果插  
的位置不存在，会创建新的值，但是步长必须为1哦

```
</span></span></code></pre>
```

</li>

</ul>

<h3 id="列表的方法">列表的方法</h3>

<ul>

<li>

<p>1、append：在列表末尾添加新的对象</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl"> lst = [1,2];
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> lst.append(3);
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> lst: [1,2,3]
```

```
</span></span></code></pre>
```

</li>

<li>

<p>2、count：统计某个元素在列表中出现的次数</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl"> lst = [1,1,[1,2],3];
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> ct = lst.count(1);
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> ct:2
```

```
</span></span></code></pre>
```

</li>

<li>

<p>3、extend：在列表末尾追加另一个列表的值</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl"> lst1 = [1,2];
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> lst2 = [3,4];
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> lst1.extend(lst2);
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> lst1: [1,2,3,4]
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> lst2: [3,4]
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> 用+号也可以，
```

是并不是在原来列表追加，而是产生新的列表

```
</span></span><span class="highlight-line"><span class="highlight-cl"> lst1 = [1,2];
</span></span><span class="highlight-line"><span class="highlight-cl"> lst2 = [3,4];
</span></span><span class="highlight-line"><span class="highlight-cl"> lst3 = lst1+lst2;
</span></span><span class="highlight-line"><span class="highlight-cl"> lst1 :[1,2]
</span></span><span class="highlight-line"><span class="highlight-cl"> lst2 :[3,4]
</span></span><span class="highlight-line"><span class="highlight-cl"> lst3 :[1,2,3,4]
</span></span></code></pre>
```

</li>
<li>

<p>4、index: 从列表中找到某个值第一个匹配项的索引位置</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl"> lst = ['h','e','l','l','o',' ','h','i']
</span></span><span class="highlight-line"><span class="highlight-cl"> ix = lst.index('h');
</span></span><span class="highlight-line"><span class="highlight-cl"> ix : 1
</span></span></code></pre>
```

</li>
<li>

<p>5、insert: 将对象插入列表中，插入的对象可以是不同类型</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl"> lst = [1,2,3];
</span></span><span class="highlight-line"><span class="highlight-cl"> lst.insert(1,'hello')
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> lst : [1,'hello',2,3]
</span></span><span class="highlight-line"><span class="highlight-cl"> 使用分片来实现,
读性不好:
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> lst = [1,2,3];
</span></span><span class="highlight-line"><span class="highlight-cl"> lst[1:1]=['hello'];
</span></span><span class="highlight-line"><span class="highlight-cl"> lst : [1,'hello',2,3]
</span></span></code></pre>
```

</li>
<li>

<p>6、pop: 移除列表中的一个元素，默认是最后一个(pop 方法是唯一一个既能返回元素值又能改列表的方法)</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl"> lst = [1,2,3];
</span></span><span class="highlight-line"><span class="highlight-cl"> v = lst.pop();
</span></span><span class="highlight-line"><span class="highlight-cl"> v : 3
</span></span><span class="highlight-line"><span class="highlight-cl"> lst : [1,2]
</span></span><span class="highlight-line"><span class="highlight-cl"> 当然也可以移除
定的
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> lst = [1,2,3];
</span></span><span class="highlight-line"><span class="highlight-cl"> v = lst.pop();
</span></span><span class="highlight-line"><span class="highlight-cl"> v : 1
</span></span><span class="highlight-line"><span class="highlight-cl"> lst : [2,3]
</span></span></code></pre>
```

</li>
<li>

<p>7、remove: 移除列表中某个值的第一个匹配项</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl"> lst=[1,3,[1,2],3];
</span></span><span class="highlight-line"><span class="highlight-cl"> lst.remove(3);
</span></span><span class="highlight-line"><span class="highlight-cl"> lst : [1,[1,2],3]
</span></span></code></pre>
```

</li>

</li>

<p>8、reverse: 将列表元素反转</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl"> lst = [1,2,3];  
</span></span><span class="highlight-line"><span class="highlight-cl"> lst.reverse();  
</span></span><span class="highlight-line"><span class="highlight-cl"> lst : [3,2,1]  
</span></span></code></pre>
```

</li>

</li>

<p>9、sort: 在原列表中对数据进行排序, 不会产生副本</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl"> lst = [3,1,2];  
</span></span><span class="highlight-line"><span class="highlight-cl"> lst.sort();  
</span></span><span class="highlight-line"><span class="highlight-cl"> lst : [1,2,3]  
</span></span></code></pre>
```

</li>

</li>

<p>10、sorted(object): 生成一份新的列表副本, 对副本进行排序, 返回值为创建的副本<br>

```
lst1 = [3,1,2];<br>
```

```
lst2 = sorted(lst1);<br>
```

```
lst1 : [3,1,2]<br>
```

```
lst2 : [1,2,3]<br>
```

另类实现<br>

```
lst1 = [3,1,2];<br>
```

```
lst2 = lst1[:];<br>
```

```
lst1 : [3,1,2]<br>
```

```
lst2 : [1,2,3]</pre>
```

</li>

</li>

<p>11、高级排序 <code>key</code>,<code>reverse</code>,</p>

</li>

</li>

<p>reverse 使用: <br>

```
lst = ['1','3','2'];<br>
```

```
lst.sort(reverse=False);<br>
```

```
lst : ['1', '2', '3']<br>
```

```
lst = ['1','3','2'];<br>
```

```
lst.sort(reverse=True)<br>
```

```
['3', '2', '1']</pre>
```

</li>

</li>

<p>key 使用 : </p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl"> lst = [3,1,2];  
</span></span><span class="highlight-line"><span class="highlight-cl"> lst.sort(key=lam  
da x:(x));  
</span></span><span class="highlight-line"><span class="highlight-cl"> [1, 2, 3]  
</span></span><span class="highlight-line"><span class="highlight-cl"> lst = [3,1,2];  
</span></span><span class="highlight-line"><span class="highlight-cl"> lst.sort(key=lam  
da x:(-x));  
</span></span><span class="highlight-line"><span class="highlight-cl"> [3, 2, 1]  
</span></span></code></pre>
```

</li>

```
</ul>
<h2 id="元组-不可变序列">元组：不可变序列</h2>
<ul>
<li>创建形式 1：用(.)。注意，最后要加多一个.</li>
<li>创建形式 2：tuple()</li>
<li>元组操作比较简单，记住其是不可变的</li>
</ul>
```