

切片

[2.7旧版教程](#)

阅读: 231781

取一个list或tuple的部分元素是非常常见的操作。比如，一个list如下：

```
>>> L = ['Michael', 'Sarah', 'Tracy', 'Bob', 'Jack']
```

取前3个元素，应该怎么做？

笨办法：

```
>>> [L[0], L[1], L[2]]
['Michael', 'Sarah', 'Tracy']
```

之所以是笨办法是因为扩展一下，取前N个元素就没辙了。

取前N个元素，也就是索引为0-(N-1)的元素，可以用循环：

```
>>> r = []
>>> n = 3
>>> for i in range(n):
...     r.append(L[i])
...
>>> r
['Michael', 'Sarah', 'Tracy']
```

对这种经常取指定索引范围的操作，用循环十分繁琐，因此，Python提供了切片（Slice）操作符，能大大简化这种操作。

对应上面的问题，取前3个元素，用一行代码就可以完成切片：

```
>>> L[0:3]
['Michael', 'Sarah', 'Tracy']
```

`L[0:3]`表示，从索引0开始取，直到索引3为止，但不包括索引3。即索引0，1，2，正好是3个元素。

如果第一个索引是0，还可以省略：

```
>>> L[:3]
['Michael', 'Sarah', 'Tracy']
```

也可以从索引1开始，取出2个元素出来：

```
>>> L[1:3]
['Sarah', 'Tracy']
```

类似的，既然Python支持`L[-1]`取倒数第一个元素，那么它同样支持倒数切片，试试：



```
>>> L[-2:]  
['Bob', 'Jack']  
>>> L[-2:-1]  
['Bob']
```

记住倒数第一个元素的索引是 `-1`。

切片操作十分有用。我们先创建一个0-99的数列：

```
>>> L = list(range(100))  
>>> L  
[0, 1, 2, 3, ..., 99]
```

可以通过切片轻松取出某一段数列。比如前10个数：

```
>>> L[:10]  
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

后10个数：

```
>>> L[-10:]  
[90, 91, 92, 93, 94, 95, 96, 97, 98, 99]
```

前11-20个数：

```
>>> L[10:20]  
[10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
```

前10个数，每两个取一个：

```
>>> L[:10:2]  
[0, 2, 4, 6, 8]
```

所有数，每5个取一个：

```
>>> L[::5]  
[0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95]
```

甚至什么都不写，只写 `[:]` 就可以原样复制一个list：

```
>>> L[:]  
[0, 1, 2, 3, ..., 99]
```

tuple也是一种**list**，唯一区别是**tuple**不可变。因此，**tuple**也可以用切片操作，只是操作的结果仍是**tuple**：

```
>>> (0, 1, 2, 3, 4, 5)[:3]  
(0, 1, 2)
```

字符串 `'xxx'` 也可以看成是一种**list**，每个元素就是一个字符。因此，字符串也可以用切片操作，只是操作结果仍是字符串：

```
>>> 'ABCDEFGH'[:3]
'ABC'
>>> 'ABCDEFGH'[:2]
'AC'
>>> 'ABCDEFGH'[:2:2]
'ACEG'
```

在很多编程语言中，针对字符串提供了很多各种截取函数（例如，`substring`），其实目的就是对于字符串切片。
`Python`没有针对字符串的截取函数，只需要切片一个操作就可以完成，非常简单。

小结

有了切片操作，很多地方循环就不再需要了。`Python`的切片非常灵活，一行代码就可以实现很多行循环才能完成的操作。

参考源码

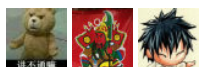
[do_slice.py](#)

感觉本站内容不错，读后有收获？

¥ 我要小额赞助，鼓励作者写出更好的教程

还可以分享给朋友

分享 [你听云杉...](#) , [bbbbbb2019](#) 等5人分享过



珠峰NODE.JS全栈开发

Angular

React

Gulp

Meteor

Vue

微信开发

技术陪伴成长社区

珠峰培训

麦子学院
www.maiziedu.com

百万级python导师亲身指导

保你120天
变身python大牛

有时候，你需要的只是一句点拨

立即咨询

掘金

一个只有高手分享
的技术社区

深度学习在线课程

通向无人驾驶的必经之路



[高级特性](#)

[迭代](#)

评论

发表评论

Sign In to Make a Comment



友情链接: [中华诗词](#) - [阿里云](#) - [SICP](#) - [4closure](#)