



链滴

Python 学习之切片 (Slice) 操作

作者: [alongblog](#)

原文链接: <https://ld246.com/article/1516605571200>

来源网站: [链滴](#)

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

列表分片

Python中的分片

Python中使用索引来访问序列中的某一个元素，如果要访问序列中某一定范围内的元素，可以使用分片来完成。

分片操作是通过由冒号相隔的两个索引号来实现的：`sequence[indexStart : indexEnd]`

分片操作后返回一个新的list列表 我们可以通过变量来接收这个列表值

示例如下：

```
numbers = [5, 1, -2, 67, 10, 23, 100]
print('numbers[2 : 4] :',numbers[2 : 4]) ##等同于list2 = numbers[2 : 4] print('numbers[2 : 4] :',lis
2)
print('numbers[2: -3] :',numbers[2: -3])
print('numbers[0:6:2] :',numbers[0:6:2]) #可以加上步长下面会讲到
输出结果：
numbers[2 : 4] : [-2, 67]
numbers[2: -3] : [-2, 67]
numbers[0:6:2] : [5, -2, 10]
```

分片操作可以表示为：

```
sequence[indexStart : indexEnd : stride]
```

按步长stride从序列sequence中取出从indexStart开始到indexEnd终止范围内的元素组成一个新的列表

indexStart: 起始索引号，它的值可以是正的也可以是负的，它所对应的元素会加入到新的序列中；索引号指的是原始序列中的索引号，而不是新序列中的索引号

indexEnd: 终止索引号，它的值可以是正的也可以是负的，它所对应的元素不会加入到新的序列中；索引号指的是原始序列中的索引号，而不是新序列中的索引号

stride: 步长，它的值可以是正的也可以是负的，可以省略，省略时使用默认步长，默认步长为正1

indexStart 可以省略，省略时，若stride为正的，则表示序列的首个元素；若stride为负的，则表示序列的最后一个元素

indexEnd 也可以省略，省略时，若stride为正的，则表示序列的最后一个元素；若stride为负的，则表示序列的首个元素

分片操作其实就是按照一定的顺序访问序列中某一范围内的元

记住一点就行了，那就是在一定的访问顺序中，indexStart所对应的元素必须先于indexEnd所对应的元素被访问到，否则，分片操作返回的就是空的序列；

具体来说，就是先看访问顺序，其实就是先看步长，因访问顺序是由步长来决定的，

当步长为正时：按从原始序列sequence的头部(从左)到尾部(到右)的顺序访问序列中的元素，indexStart在原始序列sequence中所对应的元素必须出现在indexEnd所对应的元素的左边

当步长为负时：按从原始序列sequence的尾部(从右)到头部(到左)的顺序访问序列中的元素，indexStart在原始序列sequence中所对应的元素必须出现在indexEnd所对应的元素的右边

示例如下：

```
numbers = [5, 1, -2, 67, 10, 23, 100]
print(numbers[5:2:-1])
```

```
print(numbers[0:5:2])  
print(numbers[3:3])  
print(numbers[-2:4:-1])
```