

# NumPy 切片和索引

ndarray对象的内容可以通过索引或切片来访问和修改，与 Python 中 list 的切片操作一样。

ndarray 数组可以基于 0 - n 的下标进行索引，切片对象可以通过内置的 slice 函数，并设置 start, stop 及 step 参数进行，从原数组中切割出一个新数组。

## 实例

```
import numpy as np
a = np.arange(10)
s = slice(2,7,2) # 从索引 2 开始到索引 7 停止，间隔为2
print (a[s])
```

输出结果为：

```
[2  4  6]
```

以上实例中，我们首先通过 arange() 函数创建 ndarray 对象。然后，分别设置起始，终止和步长的参数为 2，7 和 2。

我们也可以通过冒号分隔切片参数 **start:stop:step** 来进行切片操作：

## 实例

```
import numpy as np
a = np.arange(10)
b = a[2:7:2] # 从索引 2 开始到索引 7 停止，间隔为 2
print(b)
```

输出结果为：

```
[2  4  6]
```

冒号 `:` 的解释：如果只放置一个参数，如 `[2]`，将返回与该索引相对应的单个元素。如果为 `[2:]`，表示从该索引开始以后的所有项都将被提取。如果使用了两个参数，如 `[2:7]`，那么则提取两个索引(不包括停止索引)之间的项。

## 实例

```
import numpy as np
a = np.arange(10) # [0 1 2 3 4 5 6 7 8 9]
b = a[5]
print(b)
```

输出结果为：

```
5
```

### 实例

```
import numpy as np
a = np.arange(10)
print(a[2:])
```

输出结果为：

```
[2 3 4 5 6 7 8 9]
```

### 实例

```
import numpy as np
a = np.arange(10) # [0 1 2 3 4 5 6 7 8 9]
print(a[2:5])
```

输出结果为：

```
[2 3 4]
```

多维数组同样适用上述索引提取方法：

### 实例

```
import numpy as np
a = np.array([[1,2,3],[3,4,5],[4,5,6]])
print(a)
# 从某个索引处开始切割
print('从数组索引 a[1:] 处开始切割')
print(a[1:])
```

输出结果为：

```
[[1 2 3]
 [3 4 5]
 [4 5 6]]
从数组索引 a[1:] 处开始切割
[[3 4 5]
 [4 5 6]]
```

切片还可以包括省略号 `...`，来使选择元组的长度与数组的维度相同。如果在行位置使用省略号，它将返回包含行中元素的 `ndarray`。

### 实例

```
import numpy as np
a = np.array([[1,2,3],[3,4,5],[4,5,6]])
print(a[... ,1]) # 第2列元素
```

```
print (a[1,...]) # 第2行元素  
print (a[...,1:]) # 第2列及剩下的所有元素
```

输出结果为：

```
[2 4 5]  
[3 4 5]  
[[2 3]  
 [4 5]  
 [5 6]]
```

[← NumPy 从数值范围创建数组](#)

[NumPy 高级索引 →](#)

[✎ 点我分享笔记](#)