

Python科学计算库Numpy之

04-索引和切片







知识结构图

索引和切片

布尔索引

通过一个布尔数组来索引
找出与布尔数组中值为True的对应值

数组名[行切片, 列切片]

	Expression	Shape
	<code>arr[:2, 1:]</code>	<code>(2, 2)</code>
	<code>arr[2]</code> <code>arr[2, :]</code> <code>arr[2:, :]</code>	<code>(3,)</code> <code>(3,)</code> <code>(1, 3)</code>
	<code>arr[:, :2]</code>	<code>(3, 2)</code>
	<code>arr[1, :2]</code> <code>arr[1:2, :2]</code>	<code>(2,)</code> <code>(1, 2)</code>

多维数组切片

一维数组

正向索引: 正向索引从0开始, 向右依次递增
反向索引: 反向索引从-1开始, 向左依次递减。

数组名[start: end: step]

start: 起始位置, 默认0
stop: 终止位置 (不包括)
step: 步长 (默认为1)

多维数组索引

数组名[行索引, 列索引]

		axis 1		
		0	1	2
axis 0	0	0,0	0,1	0,2
	1	1,0	1,1	1,2
	2	2,0	2,1	2,2

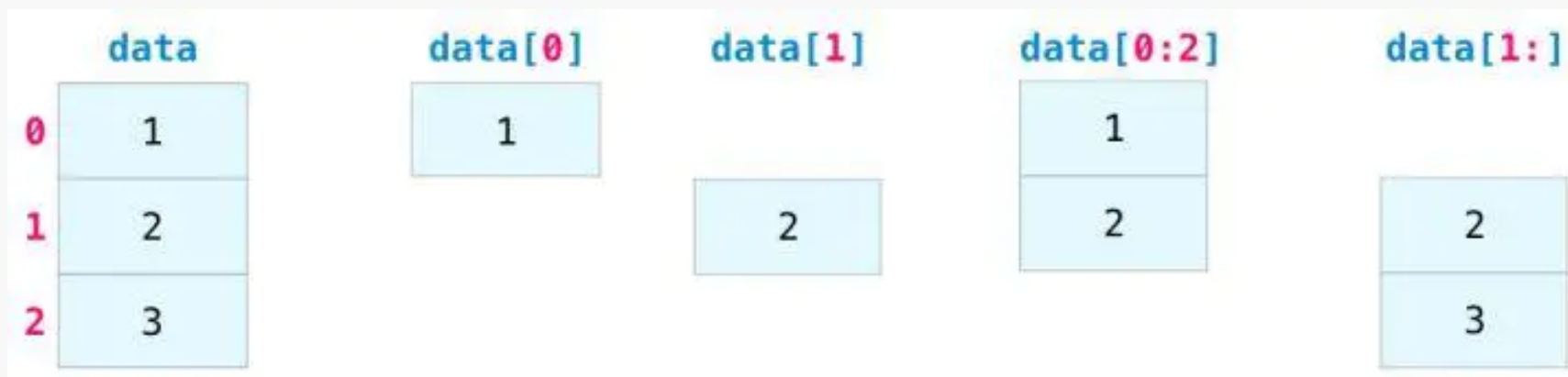
取一行 ⊖ 数组名[行索引]

取一列 ⊖ 数组名[:, 列索引]

切片+索引结合

主要内容

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。



住这个规则 **arr_name[start: end: step]** 就可以了。

特殊例子：[:]表示复制源列表

二维数组索引

二维数组中，各索引位置上的元素不再是标量，而是一维数组 **[行索引, 列索引]**

```
a = np.arange(12).reshape(3, 4)
a
```

```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11]])
```

使用单个索引来索引一整行内容

```
a[1, 3] = -1
a
```

```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6, -1],
       [ 8,  9, 10, 11]])
```

```
# 返回第二行组成的array
a[1]
```

```
array([ 4,  5,  6, -1])
```

使用切片+索引来获取一整列内容

```
a[:, 1]
```

```
array([1, 5, 9])
```

		axis 1		
		0	1	2
axis 0	0	0,0	0,1	0,2
	1	1,0	1,1	1,2
	2	2,0	2,1	2,2

多维数组切片

```
arr2d=np.arange(1,10).reshape(3,3)
arr2d
```

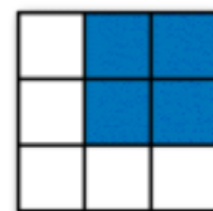
```
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
```

冒号表示从哪个位置到哪个位置，
留空表示开头或者结尾；
逗号表示行和列。

```
arr2d[2]
arr2d[:, 2]
```

```
arr2d[:2]
```

```
>>> arr2d[:2,1:]
array([[2, 3],
       [5, 6]])
>>> arr2d[1,:2]
array([4, 5])
>>> arr2d[:2,2]
array([3, 6])
>>> arr2d[:, :1]
array([[1],
       [4],
       [7]])
```



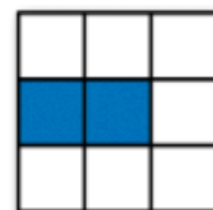
Expression	Shape
<code>arr[:2, 1:]</code>	<code>(2, 2)</code>



<code>arr[2]</code>	<code>(3,)</code>
<code>arr[2, :]</code>	<code>(3,)</code>
<code>arr[2:, :]</code>	<code>(1, 3)</code>



<code>arr[:, :2]</code>	<code>(3, 2)</code>
-------------------------	---------------------



<code>arr[1, :2]</code>	<code>(2,)</code>
<code>arr[1:2, :2]</code>	<code>(1, 2)</code>

可以通过一个**布尔数组**来索引目标数组，找出与布尔数组中**值为True**的**对应的目标**数组中的数据。
需要注意的是，布尔数组的长度必须与目标数组对应的轴的长度一致。

```
In [24]: arr = np.arange(7)
```

```
In [25]: booling1 =  
np.array([True,False,False,True,True,False,False])
```

```
In [26]: arr[booling1]  
Out[26]: array([0, 3, 4])
```

```
In [139]: arr
```

```
Out[139]:  
array([[ 0,  1,  2,  3,  4,  5,  6],  
       [ 7,  8,  9, 10, 11, 12, 13],  
       [14, 15, 16, 17, 18, 19, 20],  
       [21, 22, 23, 24, 25, 26, 27]])
```

```
In [140]: arr[arr<=15]=0
```

```
In [141]: arr
```

```
Out[141]:  
array([[ 0,  0,  0,  0,  0,  0,  0],  
       [ 0,  0,  0,  0,  0,  0,  0],  
       [ 0,  0, 16, 17, 18, 19, 20],  
       [21, 22, 23, 24, 25, 26, 27]])
```

谢谢!

