Fancy indexing 与 numpy 中的比较运算

Fancy indexing

Fancy indexing：通过索引数组，一次性地得到多个元素

```
In [1]: import numpy as np
```

```
In [2]: x = np.array(list("ABCDEFG"))
        x
```

```
Out[2]: array(['A', 'B', 'C', 'D', 'E', 'F', 'G'], dtype='<U1')
```

```
In [3]: x[1]
```

```
Out[3]: 'B'
```

```
In [4]: x[1:5]
```

```
Out[4]: array(['B', 'C', 'D', 'E'], dtype='<U1')
```

```
In [5]: x[1:5:2]
```

```
Out[5]: array(['B', 'D'], dtype='<U1')
```

```
In [6]: [x[1],x[2],x[4]]
```

```
Out[6]: ['B', 'C', 'E']
```

```
In [7]: ind = [1, 2, 4]
```

```
In [8]: x[ind]
```

```
Out[8]: array(['B', 'C', 'E'], dtype='<U1')
```

```
In [9]: x
```

```
Out[9]: array(['A', 'B', 'C', 'D', 'E', 'F', 'G'], dtype='<U1')
```

```
In [11]: ind = np.arange(1, 5).reshape(2, -1)
         ind
```

```
Out[11]: array([[1, 2],
                [3, 4]])
```

```
In [12]: x[ind]

Out[12]: array([['B', 'C'],
                ['D', 'E']], dtype='<U1')
```

```
In [13]: X = np.arange(16).reshape(4,-1)
```

```
In [14]: X

Out[14]: array([[ 0,  1,  2,  3],
                [ 4,  5,  6,  7],
                [ 8,  9, 10, 11],
                [12, 13, 14, 15]])
```

```
In [16]: row = [0, 1, 3]
         col = [1, 2, 3]
         X[row, col]

Out[16]: array([ 1,  6, 15])
```

```
In [17]: X[1, col]

Out[17]: array([5, 6, 7])
```

```
In [18]: X[1:3, col]

Out[18]: array([[ 5,  6,  7],
                [ 9, 10, 11]])
```

```
In [20]: col = [True, True, False, False]
         X[1:3, col]

Out[20]: array([[4, 5],
                [8, 9]])
```

比较

```
In [21]: x = np.arange(10)
         x

Out[21]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
In [22]: x < 5

Out[22]: array([ True,  True,  True,  True,  True, False, False, False, Fal
         se,
                False])
```

```
In [23]: x[x<5]

Out[23]: array([0, 1, 2, 3, 4])
```

```
In [24]: x >= 5
```

```
Out[24]: array([False, False, False, False, False,  True,  True,  True,  Tr
         ue,
                 True])
```

```
In [25]: x != 5
```

```
Out[25]: array([ True,  True,  True,  True,  True, False,  True,  True,  Tr
         ue,
                 True])
```

```
In [26]: x == 5
```

```
Out[26]: array([False, False, False, False, False,  True, False, False, Fal
         se,
                False])
```

```
In [27]: x-1 != x // 2
```

```
Out[27]: array([ True, False, False,  True,  True,  True,  True,  True,  Tr
         ue,
                 True])
```

```
In [28]: x+2 == x//2 + 3
```

```
Out[28]: array([False,  True,  True, False, False, False, False, False, Fal
         se,
                False])
```

```
In [29]: x[x+2 == x//2 + 3]
```

```
Out[29]: array([1, 2])
```

```
In [30]: X
```

```
Out[30]: array([[ 0,  1,  2,  3],
                [ 4,  5,  6,  7],
                [ 8,  9, 10, 11],
                [12, 13, 14, 15]])
```

```
In [31]: X < 5
```

```
Out[31]: array([[ True,  True,  True,  True],
                [ True, False, False, False],
                [False, False, False, False],
                [False, False, False, False]])
```

```
In [32]: x = x + 2
```

```
In [33]: x
```

```
Out[33]: array([ 2,  3,  4,  5,  6,  7,  8,  9, 10, 11])
```

```
In [34]: np.sum(x<8)

Out[34]: 6

In [35]: np.count_nonzero(x<8)

Out[35]: 6

In [36]: np.any(x<2)        # any  只要其中一个满足条件就为True

Out[36]: False

In [37]: np.any(x<=2)

Out[37]: True

In [38]: np.all(x>2)       # all  所有的都满足条件才为True

Out[38]: False

In [39]: np.all(x>=2)

Out[39]: True

In [40]: X

Out[40]: array([[ 0,  1,  2,  3],
               [ 4,  5,  6,  7],
               [ 8,  9, 10, 11],
               [12, 13, 14, 15]])

In [41]: np.sum(X%2 == 0)

Out[41]: 8

In [42]: np.sum(X%2 == 0, axis=0)

Out[42]: array([4, 0, 4, 0])

In [43]: np.sum(X%2 == 0, axis=1)

Out[43]: array([2, 2, 2, 2])

In [44]: X

Out[44]: array([[ 0,  1,  2,  3],
               [ 4,  5,  6,  7],
               [ 8,  9, 10, 11],
               [12, 13, 14, 15]])

In [45]: np.sum((X>2) & (X<10))

Out[45]: 7
```

```
In [48]: np.sum((X==2) | (X==10))

Out[48]: 2
```

```
In [49]: np.sum(~(X==2))

Out[49]: 15
```

```
In [50]: X[(X>2) & (X<10)]

Out[50]: array([3, 4, 5, 6, 7, 8, 9])
```

```
In [51]: X[(X==2) | (X==10)]

Out[51]: array([ 2, 10])
```

```
In [52]: X

Out[52]: array([[ 0,  1,  2,  3],
                [ 4,  5,  6,  7],
                [ 8,  9, 10, 11],
                [12, 13, 14, 15]])
```

运用案例：输出最后一列是3的倍数的所有行

```
In [54]: X[X[:, -1]%3==0, :]

Out[54]: array([[ 0,  1,  2,  3],
                [12, 13, 14, 15]])
```