

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

ndim shape size

```
In [2]: a = np.arange(10)
```

```
In [3]: a
```

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

```
In [4]: a.shape
```

```
Out[4]: (10,)
```

```
In [5]: a.size
```

```
Out[5]: 10
```

```
In [6]: a.ndim
```

```
Out[6]: 1
```

```
In [8]: A = np.ones(shape = (2, 3))
```

```
In [9]: A
```

```
Out[9]: array([[1., 1., 1.],  
               [1., 1., 1.]])
```

```
In [10]: A.ndim
```

```
Out[10]: 2
```

```
In [11]: A.shape
```

```
Out[11]: (2, 3)
```

```
In [12]: A.size
```

```
Out[12]: 6
```

```
In [13]: X = np.array([[1, 2, 3],  
                       [4, 5, 6]])
```

```
In [14]: X
```

```
Out[14]: array([[1, 2, 3],  
               [4, 5, 6]])
```

```
In [15]: X.ndim
```

```
Out[15]: 2
```

```
In [16]: X.shape
```

```
Out[16]: (2, 3)
```

```
In [17]: X.size
```

```
Out[17]: 6
```

```
In [18]: X2 = np.array([[1, 2, 3],  
                        [4, 5]])
```

```
In [19]: X2
```

```
Out[19]: array([list([1, 2, 3]), list([4, 5])], dtype=object)
```

```
In [20]: X2.ndim
```

```
Out[20]: 1
```

```
In [21]: X2.shape
```

```
Out[21]: (2,)
```

```
In [23]: X2.size    # 不是5
```

```
Out[23]: 2
```

reshape

```
In [25]: a = np.arange(10)  
a
```

```
Out[25]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
In [30]: s = a.reshape(2, 5)  
s
```

```
Out[30]: array([[0, 1, 2, 3, 4],  
               [5, 6, 7, 8, 9]])
```

```
In [28]: a    # reshape并不会改变原来的数组维度
```

```
Out[28]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
In [31]: s.ndim
```

```
Out[31]: 2
```

```
In [32]: s.shape
```

```
Out[32]: (2, 5)
```

```
In [33]: a.reshape(2, -1)      # 填 -1 numpy会自动算出需要变为多少列或多少行
```

```
Out[33]: array([[0, 1, 2, 3, 4],
                [5, 6, 7, 8, 9]])
```

```
In [34]: a.reshape(-1, 2)
```

```
Out[34]: array([[0, 1],
                [2, 3],
                [4, 5],
                [6, 7],
                [8, 9]])
```

```
In [35]: a.reshape(-1, 3)      # 可以让numpy自动算行或列，但一定要保证能够成功换算，
    否则报错
```

```
-----
-----
ValueError                                Traceback (most recent c
all last)
<ipython-input-35-8c2722caef64> in <module>()
----> 1 a.reshape(-1, 3)
```

```
ValueError: cannot reshape array of size 10 into shape (3)
```

取值

```
In [36]: m = np.arange(15).reshape(3, 5)
```

```
In [37]: m
```

```
Out[37]: array([[ 0,  1,  2,  3,  4],
                [ 5,  6,  7,  8,  9],
                [10, 11, 12, 13, 14]])
```

```
In [38]: m[0]
```

```
Out[38]: array([0, 1, 2, 3, 4])
```

```
In [40]: m[-1]
```

```
Out[40]: array([10, 11, 12, 13, 14])
```

```
In [41]: m[0][1]
```

```
Out[41]: 1
```

```
In [42]: m[(0, 1)]
```

```
Out[42]: 1
```

```
In [43]: m[0, 1]    # 一般采用这种方式
```

```
Out[43]: 1
```

切片

```
In [44]: b = np.arange(10)
```

```
In [45]: b
```

```
Out[45]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
In [46]: b[0:5]
```

```
Out[46]: array([0, 1, 2, 3, 4])
```

```
In [47]: b[:5]
```

```
Out[47]: array([0, 1, 2, 3, 4])
```

```
In [48]: b[5:]
```

```
Out[48]: array([5, 6, 7, 8, 9])
```

```
In [49]: b[0:8:2]
```

```
Out[49]: array([0, 2, 4, 6])
```

```
In [50]: b[::2]
```

```
Out[50]: array([0, 2, 4, 6, 8])
```

```
In [51]: b[::-1]
```

```
Out[51]: array([9, 8, 7, 6, 5, 4, 3, 2, 1, 0])
```

二维数组切片

```
In [53]: m
```

```
Out[53]: array([[ 0,  1,  2,  3,  4],
                 [ 5,  6,  7,  8,  9],
                 [10, 11, 12, 13, 14]])
```

```
In [54]: m[0:2, 0:3]
```

```
Out[54]: array([[0, 1, 2],
               [5, 6, 7]])
```

```
In [55]: m[:2, :3]
```

```
Out[55]: array([[0, 1, 2],
               [5, 6, 7]])
```

```
In [57]: m[:2][:3]    # 等效于 x2=m[:2]  x2[:3]  即先切出[0:2],再在前一次的结果上,
                再做[0:3]
```

```
Out[57]: array([[0, 1, 2, 3, 4],
               [5, 6, 7, 8, 9]])
```

```
In [58]: m
```

```
Out[58]: array([[ 0,  1,  2,  3,  4],
               [ 5,  6,  7,  8,  9],
               [10, 11, 12, 13, 14]])
```

```
In [59]: m[:, :2]
```

```
Out[59]: array([[ 0,  2,  4],
               [10, 12, 14]])
```

```
In [60]: m[:, :-1]
```

```
Out[60]: array([[14, 13, 12, 11, 10],
               [ 9,  8,  7,  6,  5],
               [ 4,  3,  2,  1,  0]])
```

引用关系

1. python list中的引用关系

```
In [61]: a = list(range(10))
```

```
In [62]: a
```

```
Out[62]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
In [63]: b = a[:2]
```

```
In [64]: b
```

```
Out[64]: [0, 1]
```

```
In [65]: b[1] = 99  
b
```

```
Out[65]: [0, 99]
```

```
In [66]: a      # b引用a, b的值改变不影响a
```

```
Out[66]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

2. numpy.ndarray中的引用关系

```
In [67]: m
```

```
Out[67]: array([[ 0,  1,  2,  3,  4],  
               [ 5,  6,  7,  8,  9],  
               [10, 11, 12, 13, 14]])
```

```
In [69]: type(m)
```

```
Out[69]: numpy.ndarray
```

```
In [70]: m2 = m[:2, :2]
```

```
In [71]: m2
```

```
Out[71]: array([[0, 1],  
               [5, 6]])
```

```
In [72]: m2[0, 0] = 99
```

```
In [73]: m2
```

```
Out[73]: array([[99,  1],  
               [ 5,  6]])
```

```
In [75]: m      # numpy中为了提高效率, 并没有在内存中拷贝一份, m2的值改变, m的值也就改变了
```

```
Out[75]: array([[99,  1,  2,  3,  4],  
               [ 5,  6,  7,  8,  9],  
               [10, 11, 12, 13, 14]])
```

```
In [76]: m3 = m[:3, :3].copy()  
m3
```

```
Out[76]: array([[99,  1,  2],  
               [ 5,  6,  7],  
               [10, 11, 12]])
```

```
In [77]: m3[0, 0] = 100  
m3
```

```
Out[77]: array([[100,  1,  2],  
               [  5,  6,  7],  
               [ 10, 11, 12]])
```

```
In [78]: m      # 切片时使用copy()函数, 将拷贝一份出来
```

```
Out[78]: array([[99,  1,  2,  3,  4],  
               [  5,  6,  7,  8,  9],  
               [10, 11, 12, 13, 14]])
```