

In [1]:

```
import numpy as np
```

In [2]:

```
np.arange(5, dtype=int)
```

Out[2]:

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

In [3]:

```
np.array([1, 3, 2, 5])
```

Out[3]:

```
array([1, 3, 2, 5])
```

In [4]:

```
a=np.array([[1, 3, 2, 5], [1, 2, 2, 7]])
```

In [5]:

```
type(a)
```

Out[5]:

```
numpy.ndarray
```

In [6]:

```
np.shape(a)
```

Out[6]:

```
(2, 4)
```

In [7]:

```
a.reshape((8, 1)) #a未被修改
```

Out[7]:

```
array([[1],  
       [3],  
       [2],  
       [5],  
       [1],  
       [2],  
       [2],  
       [7]])
```

In [8]:

```
a.dtype
```

Out[8]:

```
dtype('int32')
```

In [9]:

```
a.astype("bool")
```

Out[9]:

```
array([[ True,  True,  True,  True],
       [ True,  True,  True,  True]])
```

In [10]:

```
a.flatten()
```

Out[10]:

```
array([1, 3, 2, 5, 1, 2, 2, 7])
```

In [11]:

```
a+2#广播运算
```

Out[11]:

```
array([[3, 5, 4, 7],
       [3, 4, 4, 9]])
```

In [12]:

```
b=np.arange(4)
```

In [13]:

```
print(a) # (2, 4)
print(b) # (1, 4)
a+b
```

```
[[1 3 2 5]
 [1 2 2 7]]
[0 1 2 3]
```

Out[13]:

```
array([[ 1,  4,  4,  8],
       [ 1,  3,  4, 10]])
```

In [14]:

```
a.transpose() #转置
```

Out[14]:

```
array([[1, 1],
       [3, 2],
       [2, 2],
       [5, 7]])
```

In [15]:

```
a.swapaxes(1, 0)
```

Out[15]:

```
array([[1, 1],
       [3, 2],
       [2, 2],
       [5, 7]])
```

In [16]:

```
#加载数据
x=np.loadtxt('iris_data.txt', dtype=np.float, delimiter=",", skiprows=0, usecols=[0, 1, 2, 3])
y=np.loadtxt('iris_data.txt', dtype=str, delimiter=",", skiprows=0, usecols=[4])
```

In [17]:

```
c=np.arange(50)
```

In [18]:

```
d=c.reshape(5, 10)
np.shape(d)
```

Out[18]:

```
(5, 10)
```

In [19]:

```
#索引与切片
d[2] #第二行
```

Out[19]:

```
array([20, 21, 22, 23, 24, 25, 26, 27, 28, 29])
```

In [20]:

```
d[3:] #第3行及以后
```

Out[20]:

```
array([[30, 31, 32, 33, 34, 35, 36, 37, 38, 39],
       [40, 41, 42, 43, 44, 45, 46, 47, 48, 49]])
```

In [21]:

```
d[[1, 2, 4]] #第124行
```

Out[21]:

```
array([[10, 11, 12, 13, 14, 15, 16, 17, 18, 19],
       [20, 21, 22, 23, 24, 25, 26, 27, 28, 29],
       [40, 41, 42, 43, 44, 45, 46, 47, 48, 49]])
```

In [22]:

```
d[1, 3] #第1行3列
```

Out[22]:

```
13
```

In [23]:

```
d[2:, 3] #第2行及以后, 第3列
```

Out[23]:

```
array([23, 33, 43])
```

In [24]:

```
d[2, :] #第2行
```

Out[24]:

```
array([20, 21, 22, 23, 24, 25, 26, 27, 28, 29])
```

In [25]:

```
d[:, 0] #第0列
```

Out[25]:

```
array([ 0, 10, 20, 30, 40])
```

In [26]:

```
d[[1, 2], [3, 4]] #第1行3列, 第2行4列
```

Out[26]:

```
array([13, 24])
```

In [27]:

```
d[[1, 2], 2:] #第12行, 2及以后列
```

Out[27]:

```
array([[12, 13, 14, 15, 16, 17, 18, 19],
       [22, 23, 24, 25, 26, 27, 28, 29]])
```

In [28]:

```
d[:, :-1] #所有行，列到最后一列
```

Out[28]:

```
array([[ 0,  1,  2,  3,  4,  5,  6,  7,  8],
       [10, 11, 12, 13, 14, 15, 16, 17, 18],
       [20, 21, 22, 23, 24, 25, 26, 27, 28],
       [30, 31, 32, 33, 34, 35, 36, 37, 38],
       [40, 41, 42, 43, 44, 45, 46, 47, 48]])
```

In [29]:

```
#布尔索引
d>20
```

Out[29]:

```
array([[False, False, False, False, False, False, False, False, False,
        False],
       [False, False, False, False, False, False, False, False, False,
        False],
       [False,  True,  True,  True,  True,  True,  True,  True,  True,
        True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
        True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
        True]])
```

In [30]:

```
d[d>45]=100
d
```

Out[30]:

```
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, 28, 29],
       [30, 31, 32, 33, 34, 35, 36, 37, 38, 39],
       [40, 41, 42, 43, 44, 45, 100, 100, 100, 100]])
```

In [31]:

```
e=np.where(d>20, 1, 0) #三目运算符
e
```

Out[31]:

```
array([[0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
       [0, 1, 1, 1, 1, 1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]])
```

In [32]:

```
d.clip(10, 40) #裁剪
```

Out[32]:

```
array([[10, 10, 10, 10, 10, 10, 10, 10, 10, 10],
       [10, 11, 12, 13, 14, 15, 16, 17, 18, 19],
       [20, 21, 22, 23, 24, 25, 26, 27, 28, 29],
       [30, 31, 32, 33, 34, 35, 36, 37, 38, 39],
       [40, 40, 40, 40, 40, 40, 40, 40, 40, 40]])
```

In [33]:

```
f=np.array([[1, 2, 3, 4], [5, 6, 7, 8]])
e=np.array([[5, 6, 7, 8], [1, 2, 3, 4]])
```

In [34]:

```
np.vstack((f, e))
```

Out[34]:

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

In [35]:

```
np.hstack((f, e))
```

Out[35]:

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

In [36]:

```
np.zeros((2, 3))
```

Out[36]:

```
array([[0., 0., 0.],
       [0., 0., 0.]])
```

In [37]:

```
np.ones(3)
```

Out[37]:

```
array([1., 1., 1.]])
```

In [38]:

```
np.eye(3)
```

Out[38]:

```
array([[1., 0., 0.],
       [0., 1., 0.],
       [0., 0., 1.]])
```

In [39]:

```
np.random.rand(2, 3, 4)
```

Out[39]:

```
array([[[0.21470055, 0.73638781, 0.45908525, 0.51865533],
        [0.4549272 , 0.24772337, 0.38358932, 0.40412502],
        [0.18506535, 0.27905103, 0.66731935, 0.04118808]],

       [[0.74298553, 0.31831288, 0.61208098, 0.88013046],
        [0.233292  , 0.45673042, 0.61231121, 0.33440572],
        [0.65446996, 0.10392549, 0.49809605, 0.03597793]])])
```

In [40]:

```
np.random.randn(2, 3)
```

Out[40]:

```
array([[ -0.12548891, -0.24288363, -1.31729537],
       [ 0.40002965, -0.33521624, -1.20413723]])
```

In [41]:

```
np.random.randint(1, 10, (3, 4))
```

Out[41]:

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

In [42]:

```
np.random.uniform(1, 5, 10)
```

Out[42]:

```
array([2.53012265, 2.94346061, 2.46888173, 4.86603138, 4.45974962,
       4.04588147, 3.04414488, 4.29680628, 3.28262115, 1.04004688])
```

In [43]:

```
np.random.normal(0, 1, 10)
```

Out[43]:

```
array([-1.43778609,  0.43287258, -0.78821465,  0.67471583, -1.9791805 ,
       -1.77065211, -0.63740745,  0.71321314, -0.49253985, -0.52849214])
```

In [44]:

```
np.random.seed(10)
np.random.randint(1, 5, (2, 4))
```

Out[44]:

```
array([[2, 2, 1, 4],
       [1, 2, 4, 1]])
```

In [45]:

```
np.nan==np.nan
```

Out[45]:

False

In [46]:

```
x=np.array([1, np.nan, 2, np.nan, np.nan])
np.count_nonzero(x!=x)
```

Out[46]:

3

In [47]:

```
y=np.nan
np.isnan(y)
```

Out[47]:

True

In [48]:

```
np.sum(d, axis=0)
```

Out[48]:

```
array([100, 105, 110, 115, 120, 125, 184, 188, 192, 196])
```

In [49]:

```
d.sum(axis=0)
```

Out[49]:

```
array([100, 105, 110, 115, 120, 125, 184, 188, 192, 196])
```

In [50]:

```
data=np.random.randint(1, 10, (2, 10))
data
```

Out[50]:

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


In [51]:

```
data.sum(axis=0)
```

Out[51]:

```
array([ 8, 11, 11, 10, 14,  9,  9, 11,  7,  9])
```

In [52]:

```
data.sum()
```

Out[52]:

```
99
```

In [53]:

```
data.mean(axis=0)
```

Out[53]:

```
array([4. , 5.5, 5.5, 5. , 7. , 4.5, 4.5, 5.5, 3.5, 4.5])
```

In [54]:

```
np.median(data, axis=0)
```

Out[54]:

```
array([4. , 5.5, 5.5, 5. , 7. , 4.5, 4.5, 5.5, 3.5, 4.5])
```

In [55]:

```
data.max()
```

Out[55]:

```
9
```

In [56]:

```
data.min()
```

Out[56]:

```
1
```

In [57]:

```
np.ptp(data)
```

Out[57]:

```
8
```

In [58]:

```
data.std()
```

Out[58]:

2.783433131943356