

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

```
In [3]: A = np.array([[1,2,3],  
                      [4,5,6],  
                      [7,8,9]])
```

```
In [4]: A[0]
```

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

```
In [5]: A[1]
```

```
Out[5]: array([4, 5, 6])
```

```
In [6]: A[2]
```

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

```
In [7]: A[:,0]
```

```
Out[7]: array([1, 4, 7])
```

```
In [8]: A[:,1]
```

```
Out[8]: array([2, 5, 8])
```

```
In [9]: A[:,2]
```

```
Out[9]: array([3, 6, 9])
```

```
In [10]: A[0,0]
```

```
Out[10]: 1
```

```
In [13]: A[0,1]
```

```
Out[13]: 2
```

```
In [14]: A[0,2]
```

```
Out[14]: 3
```

```
In [15]: A[1,0]
```

```
Out[15]: 4
```

```
In [16]: A[1,1]
```

```
Out[16]: 5
```

```
In [17]: A[2,2]
```

```
Out[17]: 9
```

```
In [18]: A[2,1]
```

```
Out[18]: 8
```

```
In [19]: A.shape
```

```
Out[19]: (3, 3)
```

```
In [20]: np.shape(A)
```

```
Out[20]: (3, 3)
```

```
In [21]: A[1:2,0:2]
```

```
Out[21]: array([[4, 5]])
```

```
In [22]: A[1:,1:]
```

```
Out[22]: array([[5, 6],  
                [8, 9]])
```

```
In [23]: A[1:3,1:3]
```

```
Out[23]: array([[5, 6],  
               [8, 9]])
```

```
In [24]: A>5
```

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

```
In [26]: A<=5
```

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

```
In [27]: A[A>5]
```

```
Out[27]: array([6, 7, 8, 9])
```

```
In [28]: A[A<5]
```

```
Out[28]: array([1, 2, 3, 4])
```

```
In [30]: A[A<=5]
```

```
Out[30]: array([1, 2, 3, 4, 5])
```

```
In [31]: A[A>=5]
```

```
Out[31]: array([5, 6, 7, 8, 9])
```

```
In [32]: np.zeros((4,5))
```

```
Out[32]: array([[0., 0., 0., 0., 0.],  
               [0., 0., 0., 0., 0.],  
               [0., 0., 0., 0., 0.],  
               [0., 0., 0., 0., 0.]])
```

```
In [44]: np.ones((6,6))
```

```
Out[44]: array([[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., 1.],
                [1., 1., 1., 1., 1., 1.]])
```

```
In [39]: np.diag([1,2,3])
```

```
Out[39]: array([[1, 0, 0],
                [0, 2, 0],
                [0, 0, 3]])
```

```
In [45]: np.empty((3,3))
```

```
Out[45]: array([[0.00000000e+000, 0.00000000e+000, 0.00000000e+000],
                [0.00000000e+000, 0.00000000e+000, 7.78647458e-321],
                [8.70018274e-313, 8.70018274e-313, 3.11510341e-307]])
```

```
In [48]: np.random.random((4,2))
```

```
Out[48]: array([[0.26859125, 0.18632705],
                [0.43658263, 0.11516762],
                [0.51768315, 0.74339491],
                [0.93019433, 0.29884355]])
```

```
In [49]: np.eye(3,3)
```

```
Out[49]: array([[1., 0., 0.],
                [0., 1., 0.],
                [0., 0., 1.]])
```

```
In [50]: np.zeros((3,3))
```

```
Out[50]: array([[0., 0., 0.],
                [0., 0., 0.],
                [0., 0., 0.]])
```

```
In [52]: B = np.zeros((3,3))  
np.vstack((A,B))
```

```
Out[52]: array([[1., 2., 3.],  
                [4., 5., 6.],  
                [7., 8., 9.],  
                [0., 0., 0.],  
                [0., 0., 0.],  
                [0., 0., 0.]])
```

```
In [53]: np.hstack((A,B))
```

```
Out[53]: array([[1., 2., 3., 0., 0., 0.],  
                [4., 5., 6., 0., 0., 0.],  
                [7., 8., 9., 0., 0., 0.]])
```

```
In [54]: np.append(A,B)
```

```
Out[54]: array([1., 2., 3., 4., 5., 6., 7., 8., 9., 0., 0., 0., 0., 0., 0., 0., 0.,  
                0.])
```

```
In [55]: #numpy append is A and B axis with equal to with zero  
np.append(A,B,axis =0)
```

```
Out[55]: array([[1., 2., 3.],  
                [4., 5., 6.],  
                [7., 8., 9.],  
                [0., 0., 0.],  
                [0., 0., 0.],  
                [0., 0., 0.]])
```

```
In [59]: # append axis is equal too with one  
np.append(A,B,axis =1)
```

```
Out[59]: array([[1., 2., 3., 0., 0., 0.],  
                [4., 5., 6., 0., 0., 0.],  
                [7., 8., 9., 0., 0., 0.]])
```

```
In [57]: #horizontal split  
np.hsplit(A,3)
```

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

```
In [58]: # vertical split  
np.vsplit(A,3)
```

```
Out[58]: [array([[1, 2, 3]]), array([[4, 5, 6]]), array([[7, 8, 9]])]
```

```
In [60]: np.transpose(A)
```

```
Out[60]: array([[1, 4, 7],  
                [2, 5, 8],  
                [3, 6, 9]])
```

```
In [62]: A.T
```

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

```
In [63]: np.dot(A,A)
```

```
Out[63]: array([[ 30,  36,  42],  
                [ 66,  81,  96],  
                [102, 126, 150]])
```

```
In [64]: A@A
```

```
Out[64]: array([[ 30,  36,  42],  
                [ 66,  81,  96],  
                [102, 126, 150]])
```

```
In [65]: 5*A
```

```
Out[65]: array([[ 5, 10, 15],  
               [20, 25, 30],  
               [35, 40, 45]])
```

```
In [66]: A/2
```

```
Out[66]: array([[0.5, 1. , 1.5],  
               [2. , 2.5, 3. ],  
               [3.5, 4. , 4.5]])
```

```
In [68]: A+3
```

```
Out[68]: array([[ 4,  5,  6],  
               [ 7,  8,  9],  
               [10, 11, 12]])
```

```
In [69]: np.mean(A)
```

```
Out[69]: 5.0
```

```
In [70]: np.max(A)
```

```
Out[70]: 9
```

```
In [71]: np.mean(A,axis=0)
```

```
Out[71]: array([4., 5., 6.])
```

```
In [72]: np.mean(A,axis=1)
```

```
Out[72]: array([2., 5., 8.])
```

```
In [73]: np.std(A)
```

```
Out[73]: 2.581988897471611
```

```
In [74]: np.std(A,axis=0)
```

```
Out[74]: array([2.44948974, 2.44948974, 2.44948974])
```

```
In [75]: np.std(A,axis=1)
```

```
Out[75]: array([0.81649658, 0.81649658, 0.81649658])
```

```
In [76]: np.linalg.eig(A)
```

```
Out[76]: (array([ 1.61168440e+01, -1.11684397e+00, -3.38433605e-16]),
          array([[ -0.23197069, -0.78583024,  0.40824829],
                 [-0.52532209, -0.08675134, -0.81649658],
                 [-0.8186735 ,  0.61232756,  0.40824829]]))
```

```
In [77]: np.linalg.qr(A)
```

```
Out[77]: (array([[ -0.12309149,  0.90453403,  0.40824829],
                 [-0.49236596,  0.30151134, -0.81649658],
                 [-0.86164044, -0.30151134,  0.40824829]]),
          array([[ -8.12403840e+00, -9.60113630e+00, -1.10782342e+01],
                 [ 0.00000000e+00,  9.04534034e-01,  1.80906807e+00],
                 [ 0.00000000e+00,  0.00000000e+00, -1.11164740e-15]]))
```

```
In [78]: np.linalg.svd(A)
```

```
Out[78]: (array([[ -0.21483724,  0.88723069,  0.40824829],
                 [-0.52058739,  0.24964395, -0.81649658],
                 [-0.82633754, -0.38794278,  0.40824829]]),
          array([1.68481034e+01, 1.06836951e+00, 3.33475287e-16]),
          array([[ -0.47967118, -0.57236779, -0.66506441],
                 [-0.77669099, -0.07568647,  0.62531805],
                 [-0.40824829,  0.81649658, -0.40824829]]))
```

```
In [ ]:
```