Python Lab Batch S13 Date 10/02/2022, Batch Prof. Dr. Shachi Natu ¶

In [1]:

```
#!pip install numpy
```

In [2]:

```
import numpy as np
```

In [3]:

```
a= np.array([1,2,3,4],dtype=float)
```

In [4]:

```
print(a)
```

```
[1. 2. 3. 4.]
```

In [5]:

```
print(type(a))
```

```
<class 'numpy.ndarray'>
```

In [6]:

```
for element in a:
    print(element)
```

```
1.0
2.0
3.0
4.0
```

In [7]:

```
for i in range(len(a)):
    print(a[i])
```

```
1.0
2.0
3.0
4.0
```

In [8]:

```
print(a.ndim)
```

```
1
```

In [9]:

```
1 print(a.shape)
```

```
(4,)
```

In [10]:

```
1 print(a.size)
```

```
4
```

In [11]:

```
1 print(a.itemsize)
```

```
8
```

In [12]:

```
1 print(a.dtype)
```

```
float64
```

In [13]:

```
1 print(a.nbytes)
```

```
32
```

Accessing specific element aray

In [14]:

```
1 a=np.array([[1,2,3,4],[5,6,7,8],[9,10,11,12]])
```

In [15]:

```
1 a
```

Out[15]:

```
array([[ 1,  2,  3,  4],
       [ 5,  6,  7,  8],
       [ 9, 10, 11, 12]])
```

In [16]:

```
1 print(a[1,2])
```

```
7
```

In [17]:

```
1 print(a[0,:])
```

```
[1 2 3 4]
```

In [18]:

```
1 print(a[:,2])
```

```
[ 3  7 11]
```

In [20]:

```
1 print(a[:,0:2])
```

```
[[ 1  2]
 [ 5  6]
 [ 9 10]]
```

In [46]:

```
1 a=np.arange(0,15,dtype=float)
```

In [47]:

```
1 a
```

Out[47]:

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

In [48]:

```
1 a.reshape(3,5)
```

Out[48]:

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

In [24]:

```
1 a.shape
```

Out[24]:

```
(15,)
```

In [25]:

```
1 a.size
```

Out[25]:

```
15
```

In [49]:

```
a.resize((5,3))
```

In [50]:

```
a
```

Out[50]:

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

In [28]:

```
a.shape
```

Out[28]:

```
(5, 3)
```

In [31]:

```
lin_arr = np.linspace(2,10,10,endpoint=True)
```

In [32]:

```
lin_arr
```

Out[32]:

```
array([ 2.        ,  2.88888889,  3.77777778,  4.66666667,  5.55555556,
        6.44444444,  7.33333333,  8.22222222,  9.11111111, 10.        ])
```

In [33]:

```
log_arr=np.logspace(2,10,10,endpoint=True)
```

In [34]:

```
log_arr
```

Out[34]:

```
array([1.00000000e+02, 7.74263683e+02, 5.99484250e+03, 4.64158883e+04,
       3.59381366e+05, 2.78255940e+06, 2.15443469e+07, 1.66810054e+08,
       1.29154967e+09, 1.00000000e+10])
```

In [35]:

```
a = np.zeros((3,5), dtype=float)
```

In [36]:

```
1 a
```

Out[36]:

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

In [37]:

```
1 a = np.ones((3,5), dtype=float)
```

In [38]:

```
1 a
```

Out[38]:

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

In [40]:

```
1 b=np.zeros_like(a,dtype=float)
```

In [41]:

```
1 b
```

Out[41]:

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

In [42]:

```
1 c = np.full((2,2),50)
```

In [43]:

```
1 c
```

Out[43]:

```
array([[50, 50],
       [50, 50]])
```

In [44]:

```
1 np.identity(4)
```

Out[44]:

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

In [45]:

```
1 np.eye(4)
```

Out[45]:

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

In [55]:

```
1 f = a.flatten('F')
```

In [56]:

```
1 f
```

Out[56]:

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

In [57]:

```
1 a
```

Out[57]:

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

In [59]:

```
1 np.sum(a,axis=0)
```

Out[59]:

```
array([30., 35., 40.])
```

In [60]:

```
1 np.sum(a,axis=1)
```

Out[60]:

```
array([ 3., 12., 21., 30., 39.])
```

In [64]:

```
1 np.max(a,axis=1)
```

Out[64]:

```
array([ 2.,  5.,  8., 11., 14.])
```

In [65]:

```
1 np.abs(a)
```

Out[65]:

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

In [66]:

```
1 np.argmax(a)
```

Out[66]:

```
14
```

In [67]:

```
1 np.argmin(a)
```

Out[67]:

```
0
```

In [68]:

```
1 np.argmax(a,axis=0)
```

Out[68]:

```
array([4, 4, 4], dtype=int64)
```

In [69]:

```
1 np.argmax(a,axis=1)
```

Out[69]:

```
array([2, 2, 2, 2, 2], dtype=int64)
```

broadcasting in numpy array

In [71]:

```
a = np.array([[0,0,0],[1,2,3],[4,5,6]])
```

In [72]:

```
a
```

Out[72]:

```
array([[0, 0, 0],
       [1, 2, 3],
       [4, 5, 6]])
```

In [73]:

```
b=np.array([10,11,12])
```

In [74]:

```
b
```

Out[74]:

```
array([10, 11, 12])
```

In [75]:

```
a+b
```

Out[75]:

```
array([[10, 11, 12],
       [11, 13, 15],
       [14, 16, 18]])
```

In [76]:

```
c=np.array([[6],[7],[8]])
```

In [77]:

```
c
```

Out[77]:

```
array([[6],
       [7],
       [8]])
```

In [78]:

```
1 a+c
```

Out[78]:

```
array([[ 6,  6,  6],
       [ 8,  9, 10],
       [12, 13, 14]])
```

Vector Stacking

In [79]:

```
1 a=np.arange(2,14,2)
```

In [80]:

```
1 a
```

Out[80]:

```
array([ 2,  4,  6,  8, 10, 12])
```

In [81]:

```
1 b = np.array([1,2,3,4,5,6])
```

In [82]:

```
1 b
```

Out[82]:

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

In [85]:

```
1 c=np.vstack((b,a))
```

In [86]:

```
1 c
```

Out[86]:

```
array([[ 1,  2,  3,  4,  5,  6],
       [ 2,  4,  6,  8, 10, 12]])
```

In [87]:

```
1 c=np.hstack((a,b))
```

In [88]:

```
1 c
```

Out[88]:

```
array([ 2,  4,  6,  8, 10, 12,  1,  2,  3,  4,  5,  6])
```

In [89]:

```
1 a = np.array([[1,-2,3],[4,-5,6]])
```

In [90]:

```
1 b= np.array([[-1,2,-3],[-4,5,-6]])
```

In [91]:

```
1 a
```

Out[91]:

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

In [92]:

```
1 b
```

Out[92]:

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

In [93]:

```
1 c = np.vstack((a,b))
```

In [94]:

```
1 c
```

Out[94]:

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

In [95]:

```
1 c = np.hstack((a,b))
```

In [96]:

```
1 c
```

Out[96]:

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

In [98]:

```
1 c = a+b
```

In [99]:

```
1 c
```

Out[99]:

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

In [100]:

```
1 print(a-b)
```

```
[[  2  -4   6]
 [  8 -10  12]]
```

In [101]:

```
1 print(a*b)
```

```
[[ -1  -4  -9]
 [-16 -25 -36]]
```

In [102]:

```
1 print(a/b)
```

```
[[-1. -1. -1.]
 [-1. -1. -1.]]
```

where()

In [103]:

```
1 a = np.array([0,-5,-6,10])
```

In [104]:

```
1 b = np.array([3,7,-5,-9])
```

In [105]:

```
1 c = np.where(a<b,a,b)
```

In [106]:

```
1 c
```

Out[106]:

```
array([ 0, -5, -6, -9])
```

In [ ]:

```
1
```