

Sahil Saini Salaria

Reg No : 180905048

Roll No 11 C

Lab3

```
In [6]: import numpy as np
```

Array creation

```
In [14]: A=np.array([1,2,3,4,5])  
A
```

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

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

```
Out[11]: dtype('int64')
```

```
In [12]: B=np.array([1.3,2,3,4,5])  
B
```

```
Out[12]: array([1.3, 2. , 3. , 4. , 5. ])
```

```
In [13]: B.dtype
```

```
Out[13]: dtype('float64')
```

Creating sequence of sequence will create 2-dimensional array.

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

```
In [16]: A
```

```
Out[16]: array([[3, 4, 5],  
               [1, 2, 3]])
```

```
In [19]: z
```

```
Out[19]: array([[0., 0., 0., 0.],  
               [0., 0., 0., 0.]])
```

Instead of step-size, we can specify total number of elements in the array using

```
In [21]: S=np.linspace(0,2,9)
S
```

```
Out[21]: array([0. , 0.25, 0.5 , 0.75, 1. , 1.25, 1.5 , 1.75, 2.  ])
```

usage of Random Number functions

```
In [22]: import random
```

```
In [25]: random.choice([1,2,3,4,5])
```

```
Out[25]: 3
```

```
In [26]: random.choice("Sahil")
```

```
Out[26]: 'h'
```

```
In [27]: random.randrange(2,10)
```

```
Out[27]: 8
```

```
In [29]: random.randrange(2,100,3)
```

```
Out[29]: 83
```

```
In [30]: random.random()
```

```
Out[30]: 0.009318908043415286
```

```
In [32]: random.uniform(2,10)
```

```
Out[32]: 5.493144175984303
```

```
In [36]: a=[1,2,3,4,5]
random.shuffle(a)
```

```
In [37]: a
```

```
Out[37]: [2, 3, 4, 5, 1]
```

```
In [38]: random.seed(10)
```

```
In [41]: a=np.arange(15).reshape(3,5)
a
```

```
Out[41]: array([[ 0,  1,  2,  3,  4],
               [ 5,  6,  7,  8,  9],
               [10, 11, 12, 13, 14]])
```

```
In [43]: a.shape
```

Out[43]: (3, 5)

In [44]: `a.T`

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

In [45]: `np.arange(24).reshape(2,3,4)`

Out[45]: `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]])`

Array operations

In [46]: `a=np.array([1,2,3,4,])
b=np.arange(4)`

In [47]: `a`

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

In [48]: `b`

Out[48]: `array([0, 1, 2, 3])`

In [49]: `a-b`

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

In [50]: `a**2`

Out[50]: `array([1, 4, 9, 16])`

In [51]: `np.sin(a)`

Out[51]: `array([0.84147098, 0.90929743, 0.14112001, -0.7568025])`

In [52]: `10*np.sin(a)`

Out[52]: `array([8.41470985, 9.09297427, 1.41120008, -7.56802495])`

In [53]: `a<3`

Out[53]: `array([True, True, False, False])`

Matrix operations

```
In [54]: A = np.array( [[1,1],[0,1]] )  
        B = np.array( [[2,0],[3,4]] )
```

```
In [55]: A*B
```

```
Out[55]: array([[2, 0],  
               [0, 4]])
```

```
In [56]: A.dot(B)
```

```
Out[56]: array([[5, 4],  
               [3, 4]])
```

```
In [57]: np.dot(A,B)
```

```
Out[57]: array([[5, 4],  
               [3, 4]])
```

```
In [61]: b=np.arange(12).reshape(3,4)  
        b
```

```
Out[61]: array([[ 0,  1,  2,  3],  
               [ 4,  5,  6,  7],  
               [ 8,  9, 10, 11]])
```

```
In [63]: b.sum(axis=1) #Row
```

```
Out[63]: array([ 6, 22, 38])
```

```
In [64]: b.sum(axis=0)
```

```
Out[64]: array([12, 15, 18, 21])
```

```
In [65]: b.sum()
```

```
Out[65]: 66
```

Indexing, Slicing & Iterating Array

```
In [66]: a=np.arange(10)*3
```

```
In [67]: a
```

```
Out[67]: array([ 0,  3,  6,  9, 12, 15, 18, 21, 24, 27])
```

```
In [68]: a[2:5]
```

```
Out[68]: array([ 6,  9, 12])
```

```
In [69]: a[1:6:2]
```

```
Out[69]: array([ 3,  9, 15])
```

```
In [70]: b=np.arange(20).reshape(5,4)
b
```

```
Out[70]: array([[ 0,  1,  2,  3],
                [ 4,  5,  6,  7],
                [ 8,  9, 10, 11],
                [12, 13, 14, 15],
                [16, 17, 18, 19]])
```

```
In [71]: b[2,3]
```

```
Out[71]: 11
```

```
In [73]: for ele in b.flat:
          print(ele)
```

```
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
```

```
In [74]: b.ravel()
```

```
Out[74]: array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
                17, 18, 19])
```

```
In [76]: b.reshape(4,5)
```

```
Out[76]: array([[ 0,  1,  2,  3,  4],
                [ 5,  6,  7,  8,  9],
                [10, 11, 12, 13, 14],
                [15, 16, 17, 18, 19]])
```

Stacking together different arrays

```
In [77]: A1=np.arange(6).reshape(2,3)
A2=np.arange(6).reshape(2,3)
```

```
In [78]: A1
```

```
Out[78]: array([[0, 1, 2],
                [3, 4, 5]])
```

```
In [79]: A2
```

```
Out[79]: array([[0, 1, 2],
               [3, 4, 5]])
```

```
In [82]: D1=np.vstack((A1,A2))
D1
```

```
Out[82]: array([[0, 1, 2],
               [3, 4, 5],
               [0, 1, 2],
               [3, 4, 5]])
```

```
In [85]: D2=np.hstack((A1,A2))
D2
```

```
Out[85]: array([[0, 1, 2, 0, 1, 2],
               [3, 4, 5, 3, 4, 5]])
```

```
In [89]: a=np.array([2,3])
b=np.array([4,5])
np.column_stack((a,b))
```

```
Out[89]: array([[2, 4],
               [3, 5]])
```

```
In [90]: np.vstack((a,b))
```

```
Out[90]: array([[2, 3],
               [4, 5]])
```

```
In [ ]:
```

Indexing with array of indices

```
In [98]: a=np.arange(20)
i=np.array([1,1,3,8,5])
```

```
In [99]: a
```

```
Out[99]: array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
               17, 18, 19])
```

```
In [101]: a[i]
```

```
Out[101]: array([1, 1, 3, 8, 5])
```

```
In [102]: j = np.array([ [ 3, 4], [ 9, 7 ] ] )
```

```
In [103]: a[j]
```

```
Out[103]: array([[3, 4],
               [9, 7]])
```

Usage of for-loop (Mapping by Value)

```
In [104_ a=np.arange(20).reshape(4,5)
a
```

```
Out[104_ array([[ 0,  1,  2,  3,  4],
                [ 5,  6,  7,  8,  9],
                [10, 11, 12, 13, 14],
                [15, 16, 17, 18, 19]])
```

```
In [106_ s1=0
for row in a:
    for col in row:
        s1=s1+col

s1
```

```
Out[106_ 190
```

Usage of for-loop (Mapping by Index)

```
In [107_ a=np.arange(20).reshape(4,5)
a
```

```
Out[107_ array([[ 0,  1,  2,  3,  4],
                [ 5,  6,  7,  8,  9],
                [10, 11, 12, 13, 14],
                [15, 16, 17, 18, 19]])
```

```
In [108_ s1=0
for i in range(0,a.shape[0]):
    for j in range(0,a.shape[1]):
        s1=s1+a[i][j]

s1
```

```
Out[108_ 190
```

Ends

```
In [ ]:
```