

```
In [22]: !pip install numpy
```

Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: numpy in c:\programdata\anaconda3\lib\site-packages (1.24.3)

```
In [3]: #importing
import numpy as np
```

```
In [26]: a= np.array([1,2,3])
print(a)
```

```
[1 2 3]
```

```
In [66]: #2d
a=np.array([[1,3,6],[11,22,33]])
print(a)
```

```
[[ 1  3  6]
 [11 22 33]]
```

```
In [68]: #3d
b=np.array([[[1,2,3,4],[2,3,4,5],[5,6,7,8]]])
print(b)
```

```
[[[1 2 3 4]
  [2 3 4 5]
  [5 6 7 8]]]
```

```
In [75]: #4d
c=np.array([[[[33,44,55,66],[22,54,65,76],[83,77,43,22],[94,32,51,76]]]])
print(c)
```

```
[[[[33 44 55 66]
  [22 54 65 76]
  [83 77 43 22]
  [94 32 51 76]]]]]
```

```
In [76]: #import numpy as np
a=np.zeros((3,4))
print(a)
```

```
[[0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]]
```

```
In [2]: #linspace
b=np.linspace(2,8,10)
print(b)
```

```
[2.          2.66666667 3.33333333 4.          4.66666667 5.33333333
 6.          6.66666667 7.33333333 8.          ]
```

```
In [15]: #create n array of evenly spaced values (step value)
a = np.arange(2,22,3)
print(a)
```

```
[ 2  5  8 11 14 17 20]
```

```
In [20]: #samples
b = np.linspace(5,11,12)
print(b)
```

```
[ 5.          5.54545455  6.09090909  6.63636364  7.18181818  7.72727273
 8.27272727  8.81818182  9.36363636  9.90909091 10.45454545 11.          ]
```

```
In [23]: #create a dim
a = np.full((2),7)
print(a)
print(a.ndim)
```

```
[7 7]
1
```

```
In [25]: #create a 2d
b = np.full((3,4),6)
print(b)
print (b.ndim)
```

```
[[6 6 6 6]
 [6 6 6 6]
 [6 6 6 6]]
2
```

```
In [30]: #create a 3d
c = np.full((3,2,3),2)
print(c)
print(c.ndim)
```

```
[[[2 2 2]
   [2 2 2]]

  [[2 2 2]
   [2 2 2]]

  [[2 2 2]
   [2 2 2]]]
3
```

```
In [31]: #create a 4d
d =np.full((2,3,4,5),6)
print(d)
print(d.ndim)
```

```
[[[6 6 6 6 6]
  [6 6 6 6 6]
  [6 6 6 6 6]
  [6 6 6 6 6]]
```

```
[[6 6 6 6 6]
 [6 6 6 6 6]
 [6 6 6 6 6]
 [6 6 6 6 6]]
```

```
[[6 6 6 6 6]
 [6 6 6 6 6]
 [6 6 6 6 6]
 [6 6 6 6 6]]]
```

```
[[[6 6 6 6 6]
  [6 6 6 6 6]
  [6 6 6 6 6]
  [6 6 6 6 6]]
```

```
[[6 6 6 6 6]
 [6 6 6 6 6]
 [6 6 6 6 6]
 [6 6 6 6 6]]
```

```
[[6 6 6 6 6]
 [6 6 6 6 6]
 [6 6 6 6 6]
 [6 6 6 6 6]]]]
```

4

```
In [33]: #create a 2x2 identity matrix its print 0 ands 1 's
x = np.eye(5)
print(x)
```

```
[[1. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0.]
 [0. 0. 1. 0. 0.]
 [0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 1.]]
```

```
In [34]: #create an array with random values
y = np.random.random((6,6))
print(y)
```

```
[[0.48914892 0.48323483 0.76009428 0.60913521 0.83568327 0.06001858]
 [0.02837414 0.24609822 0.46947201 0.21796671 0.96410021 0.75061081]
 [0.35935275 0.17110568 0.27465464 0.7268585 0.85036651 0.62399154]
 [0.83360392 0.46605288 0.48141716 0.78292422 0.20429755 0.38899717]
 [0.40328211 0.97170349 0.67697829 0.45511356 0.39105843 0.38029658]
 [0.76122984 0.96829137 0.57436596 0.83933866 0.93726477 0.74666784]]
```

```
In [39]: #to find shape an array
xyz = np.array([[2,3,4,5],
                [5,6,7,8],
                [6,7,8,9]])
print(xyz.shape)

#3 dontes number of rows
#4 dontes number oh columns
```

(3, 4)

```
In [41]: #size to find the number of elements in a array
xyz = np.array([[2,3,4,5],
                [5,6,7,8],
                [6,7,8,9]])

print(xyz.size)
```

12

```
In [42]: #to find len of array
xyz = np.array([[2,3,4,5],
                [5,6,7,8],
                [6,7,8,9]])

print(len(xyz)) #it only counts no of rows.
```

3

```
In [51]: #access array elements
xy = np.array ([1,2,3,4,5,6])
print(xy[5])
```

6

```
In [59]: #adding two index
game = np.array ([3,4,5,6,7,8,9,10,11])
print(game[6] + game[5]) # addition
print(game[6] * game[5]) #multip
print(game[6] / game[5]) # did
print(game[6] ** game[5]) #exponents
```

17
72
1.125
43046721

```
In [61]: #access 2d arrays
ab =np.array ([[1,2,3,4],
               [4,5,6,7],
               [7,8,9,10],
               [11,12,13,14]])

print(ab[2,3])
print(ab[3,2])
```

10
13

```
In [77]: #access 3d arrays
xyz = np.array ([[22,33,44,55], [33,44,55,66]],
                 [[44,55,66,77], [55,66,77,88]],
                 [[44,55,66,77], [44,55,66,77]])

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

66

```
In [1]: # addition of two arrays
abc = [10,20,30]
xyz = [40,50,60]
print (abc + xyz)
```

[10, 20, 30, 40, 50, 60]

data types

```
In [5]: #string
x = np.array([22,33,44,55])
print(x)
print(x.dtype)

[22 33 44 55]
int32
```

```
In [9]: ab = np.array([2.1,3.5,7.6,8.3])
print(ab)
print(ab.dtype)

[2.1 3.5 7.6 8.3]
float64
```

```
In [15]: data = np.array ([2.1,3.5,7.6,8.3], dtype = "S")
print(data)
print(data.dtype)

[b'2.1' b'3.5' b'7.6' b'8.3']
|S3
```

```
In [16]: data = np.array([2.1,3.5,7.6,8.3], dtype = "i")
print(data)
print(data.dtype)

[2 3 7 8]
int32
```

```
In [17]: data = np.array ([1,2,3,4])
science = data.astype(bool)
print(science)
print(science.dtype)

[ True  True  True  True]
bool
```

```
In [18]: abc = np.array([2.1,3.5,7.6,8.3])
xyz = abc.astype("i")
print(xyz)
print(xyz.dtype)

[2 3 7 8]
int32
```

```
In [29]: #copy method
w = np.array ([11,22,33,44,55,66])
x = w.copy()
w[1]=333          # replace the element
print(x)
print(w)

[11 22 33 44 55 66]
[ 11 333  33  44  55  66]
```

```
In [30]: #view method
w = np.array ([11,22,33,44,55,66])
x = w.view()
w[1]=67          # replace the element
print(x)
print(w)

[11 67 33 44 55 66]
[11 67 33 44 55 66]
```

```
In [32]: #array shape
a = [1,2,3,4]
b = [5,6,7,8]
abc = np.array([[1,2,3,4], [5,6,7,8]])
print(abc.shape)
```

```
(2, 4)
```

reshaping array

```
In [33]: #2D
xyz = np.array ([1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20])
abc = xyz.reshape(4,5)
print(abc)
```

```
[[ 1  2  3  4  5]
 [ 6  7  8  9 10]
 [11 12 13 14 15]
 [16 17 18 19 20]]
```

```
In [34]: #3D
abc = np.array ([[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18]])
xyz = abc.reshape (2,3,3)
print(xyz)
```

```
[[[ 1  2  3]
   [ 4  5  6]
   [ 7  8  9]]

 [[10 11 12]
  [13 14 15]
  [16 17 18]]]
```

slicing

```
In [38]: #positive slice
abc = np.array([2,3,4,5,6,7,8,9])
print(abc[2:4])      #n-1
print(abc[:3])       #beginning to index 3
print(abc[2:])       #elements from 1 to end no of the array
```

```
[4 5]
[2 3 4]
[4 5 6 7 8 9]
```

```
In [41]: #negative slice
xyz = np.array([2,3,4,5,6,7,8,9])
print(xyz[2:-1])
```

```
[4 5 6 7 8]
```

```
In [48]: x= np.array ([11,222,34,56,87,95])
print(x[-3:-2])
```

```
[56]
```

```
In [50]: #slicing using step
y = np.array ([1,2,3,4,5,6,7,8])
print(y[1:7:2])
```

```
[2 4 6]
```

```
In [51]: #return entire elements from the entire array  
q = np.array ([99,88,77,66,55,44,33,22,11])  
print(q[::4])
```

```
[99 55 11]
```

```
In [52]: #slicing 2d array  
west = np.array ([[23,34,45,56],[67,78,89,91]])  
print(west[0,1:4])
```

```
[34 45 56]
```

```
In [64]: #slicing 3d array  
east = np.array ([[[23,34,45,56],[67,78,89,91],[22,34,45,67]]])  
print(east[0:1,1])
```

```
[[67 78 89 91]]
```

```
In [ ]:
```