

31/10/25

1. Array creation functions

```
In [101... import numpy as np
```

```
In [103... #create an array from a list
a=np.array([1,2,3])
print("array a:",a)
```

```
array a: [1 2 3]
```

```
In [105... #create an array with evenly spaced values
b=np.arange(0,10,2)
print("array b:",b)
```

```
array b: [0 2 4 6 8]
```

```
In [107... b1=np.arange(0,10,3)
print("array b1:",b1)
```

```
array b1: [0 3 6 9]
```

```
In [109... #create an array filled with zeros
c=np.zeros((2,3))
print("array c:",c)
```

```
array c: [[0. 0. 0.]
[0. 0. 0.]]
```

```
In [111... c=np.zeros((2,3),dtype=int)
print("array c:",c)
```

```
array c: [[0 0 0]
[0 0 0]]
```

```
In [113... #create an array filled with ones
d=np.ones((3,2))
print("array d:",d)
```

```
array d: [[1. 1.]
[1. 1.]
[1. 1.]]
```

```
In [115... d=np.ones((3,2),dtype=int)
print("array d:",d)
```

```
array d: [[1 1]
[1 1]
[1 1]]
```

```
In [117... #create an identity matrix
e=np.eye(4)
```

```
print("identity matrix:",e)

identity matrix: [[1. 0. 0. 0.]
 [0. 1. 0. 0.]
 [0. 0. 1. 0.]
 [0. 0. 0. 1.]]
```

In [119...]:

```
i=np.eye(10)
print("Identity matrix:",i)
```

```
Identity matrix: [[1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 1. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 1. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 1. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 1. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 1.]]
```

2. Array manipulation functions

In [122...]:

```
#reshape an array
a1=np.array([1,2,3])
reshaped=np.reshape(a1,(1,3)) # reshape to 1x3
print("Reshaped array:",reshaped)
```

```
Reshaped array: [[1 2 3]]
```

In [124...]:

```
#reshape an array
a1=np.array([1,2,3])
reshaped=np.reshape(a1,(3,1)) # reshape to 1x3
print("Reshaped array:",reshaped)
```

```
Reshaped array: [[1]
 [2]
 [3]]
```

In [126...]:

```
#flatten an array
f1=np.array([[1,2],[3,4]])
flatten=f1.flatten()
print("flattened array:",flatten)
```

```
flattened array: [1 2 3 4]
```

In [128...]:

```
#flatten an array
f1=np.array([[1,2],[3,4]])
ravel=f1.ravel()
print("flattened array:",ravel)
```

```
flattened array: [1 2 3 4]
```

In [130...]:

```
#flatten an array
f1=np.array([[1,2],[3,4],[5,6],[7,8]])
```

```
ravel=np.ravel(f1)
print("flattened array:",ravel)
```

flattened array: [1 2 3 4 5 6 7 8]

In [132...]

```
#transpose an array
e1=np.array([[1,2],[3,4]])
transposed=np.transpose(e1)
print("transposed array:",transposed)
```

transposed array: [[1 3]
[2 4]]

In [134...]

```
#stack arrays vertically
a2=np.array([1,2])
b2=np.array([3,4])
stacked=np.vstack([a2,b2]) #stack a2 and b2 vertically
print("stacked array:",stacked)
```

stacked array: [[1 2]
[3 4]]

In [136...]

```
#stack arrays horizontally
a2=np.array([1,2])
b2=np.array([3,4])
stacked=np.hstack([a2,b2]) #stack a2 and b2 horizontally
print("stacked array:",stacked)
```

stacked array: [1 2 3 4]

advanced indexing and slicing

In [139...]

```
import numpy as np
a=np.random.randint(0,100,(10,10))
a
```

Out[139...]

```
array([[36, 59, 37, 83, 63, 44, 51, 82, 31, 71],
       [84, 10, 51, 44, 27, 9, 87, 31, 29, 22],
       [78, 88, 2, 21, 68, 7, 47, 86, 61, 26],
       [60, 72, 52, 20, 64, 88, 67, 37, 65, 21],
       [58, 40, 25, 4, 18, 47, 26, 83, 25, 13],
       [58, 3, 71, 40, 38, 92, 27, 77, 33, 59],
       [77, 50, 46, 56, 14, 52, 72, 60, 19, 37],
       [54, 21, 75, 14, 87, 10, 18, 54, 23, 4],
       [36, 84, 2, 39, 91, 1, 98, 9, 24, 52],
       [76, 15, 36, 29, 24, 59, 39, 37, 66, 63]])
```

In [141...]

```
a[::-1]
```

```
Out[141... array([[76, 15, 36, 29, 24, 59, 39, 37, 66, 63],  
                   [36, 84, 2, 39, 91, 1, 98, 9, 24, 52],  
                   [54, 21, 75, 14, 87, 10, 18, 54, 23, 4],  
                   [77, 50, 46, 56, 14, 52, 72, 60, 19, 37],  
                   [58, 3, 71, 40, 38, 92, 27, 77, 33, 59],  
                   [58, 40, 25, 4, 18, 47, 26, 83, 25, 13],  
                   [60, 72, 52, 20, 64, 88, 67, 37, 65, 21],  
                   [78, 88, 2, 21, 68, 7, 47, 86, 61, 26],  
                   [84, 10, 51, 44, 27, 9, 87, 31, 29, 22],  
                   [36, 59, 37, 83, 63, 44, 51, 82, 31, 71]])
```

```
In [143... a[:::-2]
```

```
Out[143... array([[76, 15, 36, 29, 24, 59, 39, 37, 66, 63],  
                   [54, 21, 75, 14, 87, 10, 18, 54, 23, 4],  
                   [58, 3, 71, 40, 38, 92, 27, 77, 33, 59],  
                   [60, 72, 52, 20, 64, 88, 67, 37, 65, 21],  
                   [84, 10, 51, 44, 27, 9, 87, 31, 29, 22]])
```

```
In [145... a[:::-3]
```

```
Out[145... array([[76, 15, 36, 29, 24, 59, 39, 37, 66, 63],  
                   [77, 50, 46, 56, 14, 52, 72, 60, 19, 37],  
                   [60, 72, 52, 20, 64, 88, 67, 37, 65, 21],  
                   [36, 59, 37, 83, 63, 44, 51, 82, 31, 71]])
```

```
In [147... row=4  
col=5
```

```
In [149... col
```

```
Out[149... 5
```

```
In [151... row
```

```
Out[151... 4
```

```
In [153... a
```

```
Out[153... array([[36, 59, 37, 83, 63, 44, 51, 82, 31, 71],  
                   [84, 10, 51, 44, 27, 9, 87, 31, 29, 22],  
                   [78, 88, 2, 21, 68, 7, 47, 86, 61, 26],  
                   [60, 72, 52, 20, 64, 88, 67, 37, 65, 21],  
                   [58, 40, 25, 4, 18, 47, 26, 83, 25, 13],  
                   [58, 3, 71, 40, 38, 92, 27, 77, 33, 59],  
                   [77, 50, 46, 56, 14, 52, 72, 60, 19, 37],  
                   [54, 21, 75, 14, 87, 10, 18, 54, 23, 4],  
                   [36, 84, 2, 39, 91, 1, 98, 9, 24, 52],  
                   [76, 15, 36, 29, 24, 59, 39, 37, 66, 63]])
```

```
In [155... a[row,col]
```

```
Out[155... 47
```

In [157...]: col=6

In [159...]: a[row,col]

Out[159...]: 26

In [161...]: a[:,col]

Out[161...]: array([51, 87, 47, 67, 26, 27, 72, 18, 98, 39])

In [163...]: a[:, -3]

Out[163...]: array([82, 31, 86, 37, 83, 77, 60, 54, 9, 37])

In [165...]: a[:, 8]

Out[165...]: array([31, 29, 61, 65, 25, 33, 19, 23, 24, 66])

In [167...]: a[5:7]

Out[167...]: array([[58, 3, 71, 40, 38, 92, 27, 77, 33, 59],
[77, 50, 46, 56, 14, 52, 72, 60, 19, 37]])

In [169...]: a[0:10]

Out[169...]: array([[36, 59, 37, 83, 63, 44, 51, 82, 31, 71],
[84, 10, 51, 44, 27, 9, 87, 31, 29, 22],
[78, 88, 2, 21, 68, 7, 47, 86, 61, 26],
[60, 72, 52, 20, 64, 88, 67, 37, 65, 21],
[58, 40, 25, 4, 18, 47, 26, 83, 25, 13],
[58, 3, 71, 40, 38, 92, 27, 77, 33, 59],
[77, 50, 46, 56, 14, 52, 72, 60, 19, 37],
[54, 21, 75, 14, 87, 10, 18, 54, 23, 4],
[36, 84, 2, 39, 91, 1, 98, 9, 24, 52],
[76, 15, 36, 29, 24, 59, 39, 37, 66, 63]])

In [171...]: a[2:6, 2:4]

Out[171...]: array([[2, 21],
[52, 20],
[25, 4],
[71, 40]])

In [173...]: a[1:2, 2:4]

Out[173...]: array([[51, 44]])

In [175...]: a[2:3, 2:3]

Out[175...]: array([[2]])

In [177...]: a[2:4, 3:5]

```
Out[177... array([[21, 68],  
                  [20, 64]]))
```

```
In [179... a[3:5,3:4]
```

```
Out[179... array([[20],  
                  [ 4]]))
```

```
In [181... b=np.arange(27).reshape(3,3,3)  
b
```

```
Out[181... 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]]]))
```

```
In [183... b[0]
```

```
Out[183... array([[0, 1, 2],  
                  [3, 4, 5],  
                  [6, 7, 8]]))
```

```
In [185... b[1]
```

```
Out[185... array([[ 9, 10, 11],  
                  [12, 13, 14],  
                  [15, 16, 17]]))
```

```
In [187... b[2]
```

```
Out[187... array([[18, 19, 20],  
                  [21, 22, 23],  
                  [24, 25, 26]]))
```

```
In [189... b[:]
```

```
Out[189... 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]]]))
```

```
In [191... b[::2]
```

```
Out[191... array([[ [ 0,  1,  2],  
                   [ 3,  4,  5],  
                   [ 6,  7,  8]],  
                   [[18, 19, 20],  
                    [21, 22, 23],  
                    [24, 25, 26]]])
```

```
In [193... b[::-1]
```

```
Out[193... array([[ [18, 19, 20],  
                   [21, 22, 23],  
                   [24, 25, 26]],  
                   [[ 9, 10, 11],  
                    [12, 13, 14],  
                    [15, 16, 17]],  
                   [[ 0,  1,  2],  
                    [ 3,  4,  5],  
                    [ 6,  7,  8]]])
```

```
In [195... b[::-2]
```

```
Out[195... array([[ [18, 19, 20],  
                   [21, 22, 23],  
                   [24, 25, 26]],  
                   [[ 0,  1,  2],  
                    [ 3,  4,  5],  
                    [ 6,  7,  8]]])
```

```
In [197... b[:]
```

```
Out[197... 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]]])
```

```
In [199... b[0,1]
```

```
Out[199... array([3, 4, 5])
```

```
In [201... b[1,2]
```

```
Out[201... array([15, 16, 17])
```

```
In [203... b[2,2]
```

```
Out[203... array([24, 25, 26])
```

```
In [205... b[0,1,:]
```

```
Out[205... array([3, 4, 5])
```

```
In [207... b[1,1,:]
```

```
Out[207... array([12, 13, 14])
```

```
In [209... b[1,2,0]
```

```
Out[209... 15
```

```
In [211... b[0,1,1]
```

```
Out[211... 4
```

```
In [213... b[0,:,:1]
```

```
Out[213... array([1, 4, 7])
```

```
In [215... b[:]
```

```
Out[215... 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]]])
```

```
In [217... b[:,2,1]
```

```
Out[217... array([ 7, 16, 25])
```

```
In [219... b[1,:,:1]
```

```
Out[219... array([10, 13, 16])
```

```
In [221... b[2,1:]
```

```
Out[221... array([[21, 22, 23],
                   [24, 25, 26]])
```

```
In [223... b[2,1:,1:]
```

```
Out[223... array([[22, 23],
                   [25, 26]])
```

In [225...]: `b[0::2]`

```
Out[225...]: array([[[ 0,  1,  2],
       [ 3,  4,  5],
       [ 6,  7,  8]],
      [[18, 19, 20],
       [21, 22, 23],
       [24, 25, 26]]])
```

In [227...]: `b[0::2,0]`

```
Out[227...]: array([[ 0,  1,  2],
       [18, 19, 20]])
```

In [229...]: `b[0::2,0,:,:2]`

```
Out[229...]: array([[ 0,  2],
       [18, 20]])
```

numpy masking

In [232...]: `mat=np.arange(0,100)`
mat

```
Out[232...]: 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, 46, 47, 48, 49, 50,
       51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67,
       68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84,
       85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99])
```

In [234...]: `mat>50`

```
Out[234...]: array([False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False,
       True,  True,  True,  True,  True,  True,  True,  True,
       True])
```

In [236...]: `mat[mat>50]`

```
Out[236...]: array([51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67,
       68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84,
       85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99])
```

```
In [238]: mat[mat>=50]
```

```
Out[238...]: array([50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66,
   67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83,
   84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99])
```

```
In [240]: mat[mat<=50]
```

```
Out[240... 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, 46, 47, 48, 49, 50])
```

```
In [242]: mat[mat==50]
```

```
Out[242... array([50])
```

```
In [244]: mat[mat!=50]
```

```
Out[244...]: 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, 46, 47, 48, 49, 51,
   52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68,
   69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85,
   86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99])
```

In [246... mat>0

In [248... mat[mat>0]

```
Out[248... array([ 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, 46, 47, 48, 49, 50, 51,
   52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68,
   69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85,
   86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99])
```

In [250... mat>=0

In [252]: mat<0

In [254]: mat<=0

```
In [256]: mat[mat<=0]
```

Out[256... array([0])

```
In [258]: mat[mat>=0]
```

```
Out[258...]: 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, 46, 47, 48, 49, 50,
       51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67,
       68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84,
       85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99])
```

```
In [260... mat[mat==20]
```

```
Out[260... array([20])
```

Numpy operations

```
In [263... arr1=np.array([1,2,3,4,5])  
arr1
```

```
Out[263... array([1, 2, 3, 4, 5])
```

```
In [265... arr2=np.ones(5,dtype=int)  
arr2
```

```
Out[265... array([1, 1, 1, 1, 1])
```

```
In [267... arr1+arr2
```

```
Out[267... array([2, 3, 4, 5, 6])
```

```
In [269... arr1-arr2
```

```
Out[269... array([0, 1, 2, 3, 4])
```

```
In [271... arr1*arr2
```

```
Out[271... array([1, 2, 3, 4, 5])
```

```
In [273... arr1/arr2
```

```
Out[273... array([1., 2., 3., 4., 5.])
```

```
In [275... arr1//arr2
```

```
Out[275... array([1, 2, 3, 4, 5])
```

```
In [277... arr1%arr2
```

```
Out[277... array([0, 0, 0, 0, 0])
```

```
In [279... arr1**arr2
```

```
Out[279... array([1, 2, 3, 4, 5])
```

```
In [281... arr1**arr1
```

```
Out[281... array([ 1, 4, 27, 256, 3125])
```

```
In [283... #sum of the numbers  
arr1.sum()
```

Out[283... 15

In [285... *#max of numbers*
arr1.max()

Out[285... 5

In [287... *#min of numbers*
arr1.min()

Out[287... 1

In [289... *#mean*
arr1.mean()

Out[289... 3.0

In [291... *#median*
arr1.median()

```
-----  
AttributeError  
Cell In[291], line 2  
      1 #median  
----> 2 arr1.median()
```

Traceback (most recent call last)

```
AttributeError: 'numpy.ndarray' object has no attribute 'median'
```

In [293... *from numpy import**
#median
median(arr1)

Out[293... 3.0

In [295... *#standard deviation*
arr1.std()

Out[295... 1.4142135623730951

In [297... *#variance*
arr1.var()

Out[297... 2.0

In [299... *#sin fuction*
np.sin(arr1)

Out[299... array([0.84147098, 0.90929743, 0.14112001, -0.7568025 , -0.95892427])

In [301... *#cos fuction*
np.cos(arr1)

Out[301... array([0.54030231, -0.41614684, -0.9899925 , -0.65364362, 0.28366219])

```
In [303... #tan function
np.tan(arr1)
```

```
Out[303... array([ 1.55740772, -2.18503986, -0.14254654,  1.15782128, -3.38051501])
```

vstack(),hstack(),vsplit(),hsplit(),view() and copy()

```
In [306... #creating 2 arrays
a1=np.array([[1,2,3],[4,5,6]])
print(a1)
b1=np.array([[7,8,9],[5,6,7]])
print(b1)
```

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

```
In [308... vertical_stack=np.vstack((a1,b1))
print(vertical_stack)
```

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

```
In [310... #horizontally stacked
horizontal_stack=np.hstack((a1,b1))
print(horizontal_stack)
```

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

```
In [312... print(a1)
```

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

```
In [314... splitting=np.split(a1,2)
print(splitting)
```

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

```
In [316... splitting=np.vsplit(a1,2)
print(splitting)
```

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

```
In [318... splitting=np.hsplit(a1,3)
print(splitting)
```

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

```
In [320... arr=np.array([[5,4,4,7],[7,5,0,6],[4,3,7,6]])
print(arr)
```

```
[[5 4 4 7]
 [7 5 0 6]
 [4 3 7 6]]
```

```
In [322... #vertical splitting
splitting=np.vsplit(arr,3)
print(splitting)
```

```
[array([[5, 4, 4, 7]]), array([[7, 5, 0, 6]]), array([[4, 3, 7, 6]])]
```

```
In [324... #horizontal splitting
splitting=np.hsplit(arr,4)
print(splitting)
```

```
[array([[5],
       [7],
       [4]]), array([[4],
       [5],
       [3]]), array([[4],
       [0],
       [7]]), array([[7],
       [6],
       [6]])]
```

```
In [326... #view and copy fuctions
a = np.array([[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]])
print(a)
```

```
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]]
```

```
In [328... b=a[0,:]
print(b)
```

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

```
In [330... b[0]=90
print(b)
```

```
[90  2  3  4]
```

```
In [332... print(a)
```

```
[[90  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]]
```

```
In [334... b1=a.copy()
print(b1)
```

```
[[90  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]]
```

```
In [336... b[1]=100
      print(b)
```

```
[ 90 100  3  4]
```

```
In [338... print(b1)
```

```
[[90  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]]
```

```
In [311... print(a)
```

```
[[ 90 100  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]]
```

Unique function

```
In [209... u=np.array([1,2,3,4,2,4,5,6,7,8,4,9,10,11,10,11,8,9,11,2,1,12,12,13])
      print(u)
```

```
[ 1  2  3  4  2  4  5  6  7  8  4  9 10 11 10 11  8  9 11  2  1 12 12 13]
```

```
In [213... unique_items=np.unique(u)
      print(unique_items)
```

```
[ 1  2  3  4  5  6  7  8  9 10 11 12 13]
```

```
In [223... #to get indexes of unique values
      index_values=np.unique(u,return_index=True)
      print(index_values)
```

```
(array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13]), array([ 0,  1,  2,  3,  6,  7,  8,  9, 11, 12, 13, 21, 23], dtype=int64))
```

```
In [225... #frequency count of unique values
      occurrence_count=np.unique(u,return_counts=True)
      print(occurrence_count)
```

```
(array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13]), array([2, 3, 1, 3, 1, 1, 1, 2, 2, 3, 2, 1], dtype=int64))
```

```
In [229... unique_items2=np.unique(u,return_index=True,return_counts=True)
      print(unique_items3)
```

```
(array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13]), array([ 0,  1,  2,  3,  6,  7,  8,  9, 11, 12, 13, 21, 23], dtype=int64), array([2, 3, 1, 3, 1, 1, 1, 2, 2, 3, 2, 1], dtype=int64))
```

```
In [233... u2d=np.array([[1,2,3,4],[5,6,7,8],[9,10,11,12],[1,2,3,4]])
      print(u2d)
```

```
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]
 [ 1  2  3  4]]
```

```
In [235... uniq=np.unique(u2d)
uniq
```

```
Out[235... array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12])
```

```
In [237... uniq2=np.unique(u2d,return_index=True,return_counts=True)
uniq2
```

```
Out[237... (array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12]),
 array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11], dtype=int64),
 array([2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1], dtype=int64))
```

flip fuction (reverse)

```
#1D array
f1=np.array([1,2,3,4,5,6])
flip1=np.flip(f1)
print(flip1)
```

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

```
#2D array
f2=np.array([[1,2,3],[3,4,5],[5,6,7]])
flip2=np.flip(f2)
print(flip2)
```

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

```
#reverse only rows
flip3=np.flip(f2, axis=0)
print(flip3)
```

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

```
#reverse only columns
flip4=np.flip(f2, axis=1)
print(flip4)
```

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

```
#reversing single row at index position
flip5=f2[0]=np.flip(f2[0])
print(flip5)
```

[1 2 3]

```
In [26]: flip6=f2[2]=np.flip(f2[2])
print(flip6)
```

[5 6 7]

```
In [27]: #reversing single column at index position
flip7=f2[:,1]=np.flip(f2[:,1])
print(flip7)
```

[6 4 2]

In []:

In []: