

```
In [5]: #To install Numpy package and do the basic functions
import numpy as np
a=np.array([[1,2,4],[5,8,7]])
print("Array created using passed list:\n",a)
b=np.zeros((3,4))
print("\nAn Array initialized with all zeros:\n",b)
g=b.astype('i')
print(g)
c=np.full((3,3),6)
print(c)
d=np.random.random((1,2))
print(d)
e=np.arange(1,45,7)
print(e)
f=np.array([[1,3,4,5],[4,3,4,6],[54,5,0,76]])
new=f.reshape(6,2)
print(f)
flarr=f.flatten()
print(flarr)
print(f.ndim)
print(f.shape)
print(f.size)
print(f.dtype)
h=f.astype('f')
```

Array created using passed list:

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

An Array initialized with all zeros:

```
[[0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]]
[[0 0 0 0]
 [0 0 0 0]
 [0 0 0 0]]
[[6 6 6]
 [6 6 6]
 [6 6 6]]
[[0.51862929 0.90548946]]
[ 1  8 15 22 29 36 43]
[[ 1  3  4  5]
 [ 4  3  4  6]
 [54  5  0 76]]
[ 1  3  4  5  4  3  4  6 54  5  0 76]
2
(3, 4)
12
int32
```

```
In [6]: import numpy as np
arr=np.array([[ -1,2,0,4],[4,-0.5,6,0],[2.6,0,7,8],[3,-7,4,2.0]])
print(arr)
print(arr[0:3:1])
temp=arr[:,2:3]
print(temp)
temp=arr[[0,1,2,3],[2,3,1,0]]
print(temp)
c=arr>2
temp=arr[c]
print(temp)
arr=np.array([1,2,3,4,5,6,7])
print(arr)
```

```

print(arr[:,2])
import numpy as np
arr1=np.array([1,2,3])
arr2=np.array([4,5,6])
arr=np.concatenate((arr1,arr2))
print(arr1,arr2)
print(arr)
arr=np.hstack((arr1,arr2))
print(arr)
arr=np.vstack((arr1,arr2))
print(arr)
arr=np.dstack((arr1,arr2))
print(arr)
arr=np.array([1,2,3,4,5,6])
newarr=np.array_split(arr,3)
print(arr)
print(newarr)
print(newarr[0])
print(newarr[1])
print(newarr[2])

```

```

[[-1.  2.  0.  4. ]
 [ 4. -0.5 6.  0. ]
 [ 2.6  0.  7.  8. ]
 [ 3. -7.  4.  2. ]]
[[-1.  2.  0.  4. ]
 [ 4. -0.5 6.  0. ]
 [ 2.6  0.  7.  8. ]]
[[-1.  2.  0. ]
 [ 4. -0.5 6. ]]
[0. 0. 0. 3.]
[4. 4. 6. 2.6 7. 8. 3. 4. ]
[1 2 3 4 5 6 7]
[1 3 5 7]
[1 2 3] [4 5 6]
[1 2 3 4 5 6]
[1 2 3 4 5 6]
[[1 2 3]
 [4 5 6]]
[[[1 4]
 [2 5]
 [3 6]]]
[1 2 3 4 5 6]
[array([1, 2]), array([3, 4]), array([5, 6])]
[1 2]
[3 4]
[5 6]

```

```

In [7]: import numpy as np
arr=np.array([1,2,3,4,5,4,4])
print("original array:",arr)
x=np.where(arr==4)
print("\nindexes where the value is 4:",x)
arr=np.array([1,2,3,4,5,6,7,8])
x=np.where(arr%2==0)
print("\noriginal array:",arr)
print("\nindexes where the values are even",x)
x=np.searchsorted(arr,3,side='left')
print("\nindexes where the value 3 should be inserted starting from the right:",x)
arr=np.array([3,2,0,1])
print("\noriginalarray:",arr)
print("\nsorted array",np.sort(arr))
arr=np.array([[3,2,4],[5,0,1]])
print("\noriginal array:",arr)

```

```

print("\nsorted array:",np.sort(arr))
arr=np.array([41,42,43,44])
x=[True,False,True,False]
newarr=arr[x]
print("\noriginal array:",arr)
print("\nfilter index",x)
print("\nfilter array",newarr)
arr=np.array([41,42,43,44])
filter_arr=arr>42
newarr=arr[filter_arr]
print("\noriginal array",arr)
print("\nfilter array:condition>42",filter_arr)
print("\nnew array",newarr)

```

original array: [1 2 3 4 5 4 4]

indexes where the value is 4: (array([3, 5, 6], dtype=int64),)

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

indexes where the values are even (array([1, 3, 5, 7], dtype=int64),)

indexes where the value 3 should be inserted starting from the right: 2

originalarray: [3 2 0 1]

sorted array [0 1 2 3]

original array: [[3 2 4]
[5 0 1]]

sorted array: [[2 3 4]
[0 1 5]]

original array: [41 42 43 44]

filter index [True, False, True, False]

filter array [41 43]

original array [41 42 43 44]

filter array:condition>42 [False False True True]

new array [43 44]

```

In [8]: import numpy as np
a=np.array([10,20,30,40,50])
b=np.array([2,4,5,8,10])
print("original arrays")
print(a)
print(b)
print("\nvector addition")
print(a+b)
print("\nvector subtraction")
print(a-b)
print("\nvector multiplication")
print(a*b)
print("\nvector division")
print(a/b)
print("\nvector dot product")
print(a.dot(b))
print("\nscalar multiplication")
sclr=5

```

```
print("scalar value=",sclr)
print("array=",a)
print("result=",a*sclr)
```

original arrays
[10 20 30 40 50]
[2 4 5 8 10]

vector addition
[12 24 35 48 60]

vector subtraction
[8 16 25 32 40]

vector multiplication
[20 80 150 320 500]

vector division
[5. 5. 6. 5. 5.]

vector dot product
1070

scalar multiplication
scalar value= 5
array= [10 20 30 40 50]
result= [50 100 150 200 250]

In []: pip install nbconvert

In []:

In []: