



numpy

www.hbpatel.in

```
import numpy as np
import sys

a = np.array([1, 2, 3], dtype='int8')
b = np.array([[1.0, 2.0, 3.0], [4.0, 5.0, 6.0]])

print(a)

print("Dimension - a : ", a.ndim)
print("Dimension - b : ", b.ndim)

print("a - Type: ", a.dtype, "Size: ", a.itemsize, "bytes")
print("b - Type: ", b.dtype, "Size: ", b.itemsize, "bytes")
```

```
[1 2 3]
Dimension - a : 1
Dimension - b : 2
a - Type: int8 Size: 1 bytes
b - Type: float64 Size: 8 bytes
```

```
In [1]: import numpy as np
import sys

In [3]: a = np.array([[11,22,33,44,55],[10,20,30,40,50]])
print(a)

[[11 22 33 44 55]
 [10 20 30 40 50]]

In [5]: a.shape
Out[5]: (2, 5)

In [6]: a[0, 2]
Out[6]: 33

In [7]: a[1,-2]
Out[7]: 40

In [8]: a[0,:]
Out[8]: array([11, 22, 33, 44, 55])

In [9]: a[:, 0]
Out[9]: array([11, 10])
```

```

In [10]: a[1,4]=99
          print(a)

[[11 22 33 44 55]
 [10 20 30 40 99]]

In [11]: a[:, 2]=101
          print(a)

[[ 11  22 101  44  55]
 [ 10  20 101  40  99]]

In [12]: np.zeros(5)
Out[12]: array([0., 0., 0., 0., 0.])

In [14]: np.zeros((2, 3))
Out[14]: array([[0., 0., 0.],
                [0., 0., 0.]])

In [15]: np.ones(5)
Out[15]: array([1., 1., 1., 1., 1.])

In [16]: np.ones((2, 3))
Out[16]: array([[1., 1., 1.],
                [1., 1., 1.]])

In [18]: np.full((2,4),99)
Out[18]: array([[99, 99, 99, 99],
                [99, 99, 99, 99]])

```

```

In [19]: np.full((2,4),99, dtype='float32')
Out[19]: array([[99., 99., 99., 99.],
                [99., 99., 99., 99.]], dtype=float32)

In [20]: np.random.rand(4,2)
Out[20]: array([[0.59161756, 0.23972281],
                [0.33114222, 0.47488643],
                [0.12111486, 0.71373998],
                [0.75847543, 0.78653875]])

In [21]: np.random.random_sample(a.shape)
Out[21]: array([[0.58526244, 0.51710963, 0.4514024 , 0.30242931, 0.2918369 ],
                [0.86825353, 0.43792201, 0.47336005, 0.41503633, 0.55981663]])

In [23]: np.random.randint(10,size=(2,4))
Out[23]: array([[8, 3, 4, 2],
                [9, 1, 5, 8]])

In [24]: np.random.randint(10,20,size=(2,4))
Out[24]: array([[18, 10, 13, 12],
                [19, 15, 15, 17]])

In [25]: np.identity(3)
Out[25]: array([[1., 0., 0.],
                [0., 1., 0.],
                [0., 0., 1.]])

```

```

In [25]: np.identity(3)
Out[25]: array([[1., 0., 0.],
                [0., 1., 0.],
                [0., 0., 1.]])

In [27]: arr = np.array([1,2,3])
          rp = np.repeat(arr, 4)
          print(rp)

[1 1 1 1 2 2 2 2 3 3 3 3]

In [28]: a = np.array([1,2,3,4])
          print(a)

[1 2 3 4]

In [29]: a+2
Out[29]: array([3, 4, 5, 6])

In [30]: a-2
Out[30]: array([-1,  0,  1,  2])

In [31]: a*2
Out[31]: array([2, 4, 6, 8])

In [32]: a/2
Out[32]: array([0.5, 1. , 1.5, 2. ])

```

numpy

```

Out[33]: array([2, 2, 4, 4])

In [34]: a ** 2
Out[34]: array([ 1,  4,  9, 16], dtype=int32)

In [35]: np.sin(a)
Out[35]: array([ 0.84147098,  0.90929743,  0.14112001, -0.7568025 ])

In [36]: np.cos(a)
Out[36]: array([ 0.54030231, -0.41614684, -0.9899925 , -0.65364362])

```

numpy

```
In [37]: np.sin(a)
Out[37]: array([ 0.84147098,  0.90929743,  0.14112001, -0.7568025 ])
```

```
In [44]: a = np.ones((2, 3))
print (a)

b = np.full((3, 2), 4)
print (b)

c = np.matmul(a, b)
print (c)

[[1.  1.  1.]
 [1.  1.  1.]]
[[4 4]
 [4 4]
 [4 4]]
[[12. 12.]
 [12. 12.]
```

```
In [47]: # Find determinant
c = np.identity(3)
np.linalg.det(c)

Out[47]: 1.0
```

```
In [50]: stats = np.array ([[1,2,3],[4,5,6]])
stats

Out[50]: array([[1, 2, 3],
               [4, 5, 6]])
```

```
In [52]: np.min(stats)
Out[52]: 1

In [53]: np.max(stats)
Out[53]: 6

In [56]: np.max(stats, axis=1)
Out[56]: array([3, 6])

In [57]: np.max(stats, axis=0)
Out[57]: array([4, 5, 6])

In [58]: np.sum(stats)
Out[58]: 21

In [59]: np.sum(stats, axis=0)
Out[59]: array([5, 7, 9])

In [60]: np.sum(stats, axis=1)
Out[60]: array([ 6, 15])
```

```
In [64]: pre = np.array([[1,2,3,4],[5,6,7,8]])
print(pre)

post = pre.reshape((1,8))
print(post)

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

```
In [62]: pre = np.array([[1,2,3,4],[5,6,7,8]])
print(pre)

post = pre.reshape((2,4))
print(post)

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

```
In [63]: pre = np.array([[1,2,3,4],[5,6,7,8]])
print(pre)

post = pre.reshape((2,2,2))
print(post)

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

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

```
In [67]: v1 = np.array([1,2,3,4])
v2 = np.array([5,6,7,8])

np.vstack([v1, v2])
np.vstack([v1, v2, v2, v2])

Out[67]: array([[1, 2, 3, 4],
               [5, 6, 7, 8],
               [5, 6, 7, 8],
               [5, 6, 7, 8]])
```

```
In [70]: h1 = np.ones((2,4))
h2 = np.zeros((2,2))
np.hstack([h1, h2])

Out[70]: array([[1., 1., 1., 1., 0., 0.],
               [1., 1., 1., 1., 0., 0.]])
```



numpy

www.hbpatel.in

Home Page - Select or create a notebook x Untitled - Jupyter Notebook x Home Page - Select or create a notebook x +

localhost:8888/tree

jupyter

Quit Logout

Files Running Clusters

Select items to perform actions on them.

0 /

- anaconda3
- Contacts
- Desktop
- Documents
- Downloads

Upload New

Name

Notebook:

Python 3 (ipykernel)

Other:

Text File

Folder

Terminal

2 hours ago

Home Page - Select or create a notebook x Untitled - Jupyter Notebook x Home Page - Select or create a notebook x data.txt - Jupyter Text Editor x +

localhost:8888/edit/data.txt

jupyter data.txt a few seconds ago

Logout

File Edit View Language Plain Text

```
1 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
2 11, 22, 33, 44, 55, 66, 77, 88, 99, 111
3 10, 20, 30, 40, 50, 60, 70, 80, 90, 100
```



numpy

www.hbpatel.in

```
In [74]: np.genfromtxt('data.txt',delimiter=',')
```

```
Out[74]: array([[ 1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10.],
 [11., 22., 33., 44., 55., 66., 77., 88., 99., 111.],
 [10., 20., 30., 40., 50., 60., 70., 80., 90., 100.]])
```

```
In [76]: filedata = np.genfromtxt('data.txt',delimiter=',')
filedata
```

```
Out[76]: array([[ 1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10.],
 [11., 22., 33., 44., 55., 66., 77., 88., 99., 111.],
 [10., 20., 30., 40., 50., 60., 70., 80., 90., 100.]])
```

```
In [78]: filedata = np.genfromtxt('data.txt',delimiter=',')
filedata = filedata.astype('int32')
filedata
```

```
Out[78]: array([[ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10],
 [11, 22, 33, 44, 55, 66, 77, 88, 99, 111],
 [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]])
```

```
In [79]: filedata > 50
```

```
Out[79]: array([[False, False, False, False, False, False, False, False, False,
False],
 [False, False, False, False, True, True, True, True, True,
True],
 [False, False, False, False, False, True, True, True, True,
True]])
```

```
In [82]: filedata[filedata > 50]
```

```
Out[82]: array([ 55,  66,  77,  88,  99, 111,  60,  70,  80,  90, 100])
```

```
In [83]: np.any(filedata > 50, axis=0)
```

```
Out[83]: array([False, False, False, False, True, True, True, True, True,
True])
```

```
In [84]: np.all(filedata > 50, axis=0)
```

```
Out[84]: array([False, False, False, False, False, False, False, False, False,
False])
```

```
In [86]: ((filedata > 50) & (filedata < 100))
```

```
Out[86]: array([[False, False, False, False, False, False, False, False, False,
False],
 [False, False, False, False, True, True, True, True, True,
False],
 [False, False, False, False, False, True, True, True, True,
False]])
```

```
In [87]: ~((filedata > 50) & (filedata < 100))
```

```
Out[87]: array([[ True,  True,  True,  True,  True,  True,  True,  True,  True,
 True],
 [ True,  True,  True,  True, False, False, False, False, False,
 True],
 [ True,  True,  True,  True,  True, False, False, False, False,
 True]])
```