

numpy

In [1]: `import numpy as np`

In [2]: `a=np.array([[1,2,8],[4,5,7]],dtype=np.int32)`
`print(a)`

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

In [3]: `a.ndim` *#dimensions so it is 2 dimensional array*

Out[3]: 2

In [4]: `a.shape` *#it gives the shape(row,column)*

Out[4]: (2, 3)

In [5]: `print(a.dtype)` *#tells the type of elements inside the array*
`print(type(a))` *#it tells the data type of a which is numpy array*

```
int32
<class 'numpy.ndarray'>
```

In [6]: `np.ones((3,4),dtype=np.int32)` *#makes all elements of array as one*

Out[6]: `array([[1, 1, 1, 1],`
 `[1, 1, 1, 1],`
 `[1, 1, 1, 1]])`

In [7]: `np.arange(1,15)`

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

In [8]: `b=np.array([[1,3,4],[3,3,5]])`
`np.concatenate([a,b])`

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

In [9]: `print(a.astype(np.float32))` *#conversion of array*

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

In [10]: `a[0,0]=5`
`print(a)`
`print(a.astype(np.float32))`

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

In [11]:

```
c=np.arange(0,6)
print(c)
print(c.shape)
c=c.reshape(2,3)#reshaped the matrix from (6,1)to (2,3)
print(c)
c=c.ravel()#again backs to normal
print(c)
```

```
[0 1 2 3 4 5]
(6,)
[[0 1 2]
 [3 4 5]]
[0 1 2 3 4 5]
```

In [12]:

```
c=np.arange(0,6).reshape(2,3)
print (c)
c=c.T#transposes the matrix 1st way
print(c)
c=c.transpose((1,0))#again transposes 2nd way
print(c)
c=c.transpose((1,0))#again transposes
print(c)
```

```
[[0 1 2]
 [3 4 5]]
[[0 3]
 [1 4]
 [2 5]]
[[0 1 2]
 [3 4 5]]
[[0 3]
 [1 4]
 [2 5]]
```

In [13]:

```
np.savez('data.npz',c=c)
data = np.load('data.npz')
c= data['c']
print(c)
```

```
[[0 3]
 [1 4]
 [2 5]]
```

mathematical operations

In [29]:

```
print(a)
print(b)
a*b
print(a*b)
a=a/b
a
```

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

```
[[ 5  6 32]
 [12 15 35]]
```

```
Out[29]: array([[5.          , 0.66666667, 2.          ],
                [1.33333333, 1.66666667, 1.4          ]])
```

```
In [ ]: b1=np.array([[2,6,8],[6,6,10]],dtype=np.int32)
        b1
        b2=np.array([[1,3,9],[3,3,5]],dtype=np.int32)
        b2
        b1 = b1/b2
        b1
```

```
In [ ]: a[0,:]
```

```
In [ ]: a[:,::-1]
```

```
In [ ]: # a[:, :, ::-1]error
```

```
In [ ]: # a[a<10]=0 setting condition for all elements to 0
        # # set a =0
```

```
In [23]: e=list(range(10))
        print(e)
        e[4::-1]# reversing the list logic yaad rakhna
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
Out[23]: []
```

```
In [28]: print(a)
        a.sum()
        a.sum(axis=1,keepdims=True)
```

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

```
Out[28]: array([[15],
                [16]])
```

```
In [36]: np.ones((3,4),dtype=np.int32)
```

```
Out[36]: array([[1, 1, 1, 1],
                [1, 1, 1, 1],
                [1, 1, 1, 1]])
```

```
In [ ]:
```