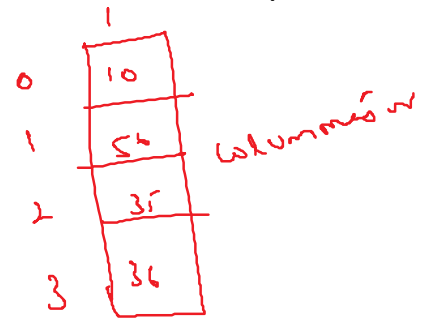


Day22_Numpy_Package

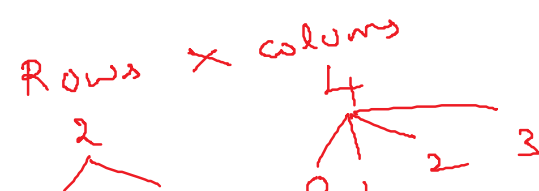
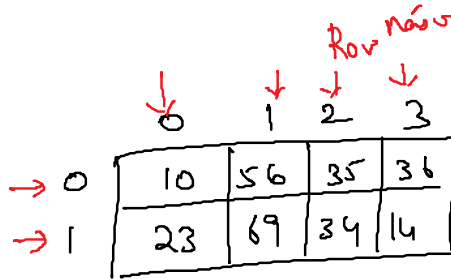
numpy stands for numerical python. It contains the components that work on the mathematical structures which are put into various dimensions

```
import numpy as np
arr1d = np.array([10,56,35,36])
print(arr1d)
#[10 56 35 36]
```

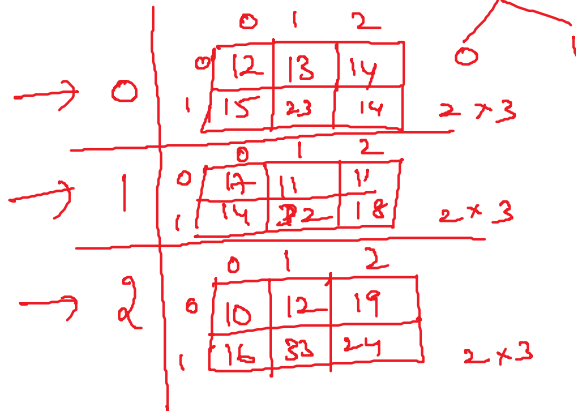


```
type(arr1d)
#numpy.ndarray
```

```
arr2d = np.array([[10,56,35,36],
                 [23,69,34,14]])
print(arr2d)
```



```
arr3d = np.array([[[12,13,14],
                  [15,23,14]],
                 [[17,11,11],
                  [14,32,18]],
                 [[10,12,19],
                  [16,33,24]]])
```



```
arr3d
```

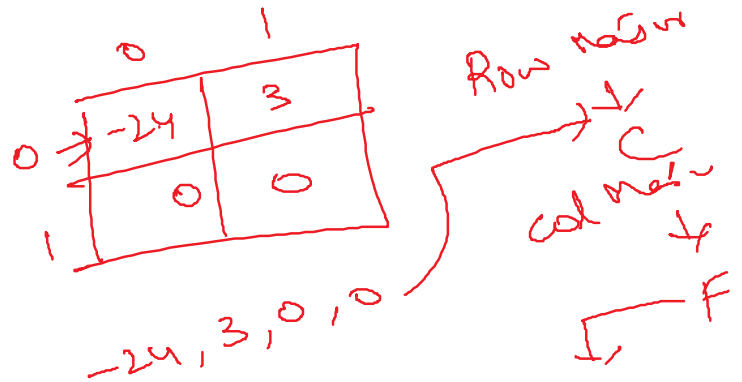
```
arr3d.ndim
#3
```

```
arr3d.shape
#(3, 2, 3)
```

```
#3 Arrays of 2R and 3 C
```

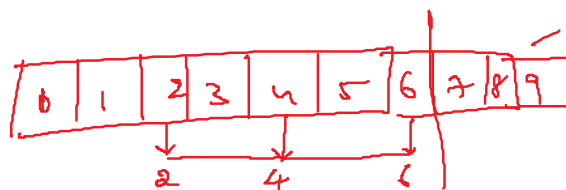
```
np.empty((2,2), dtype = np.int8, order = 'C')
```

```
array([[ -24,  3],
       [  0,  0]], dtype=int8)
```



Slicing

```
a = np.arange(10)
print(a)
#[0 1 2 3 4 5 6 7 8 9]
```



```
s = slice(2,7,2)
print(a[s])
#[2 4 6]
```

$a[2:7:2]$
 $slice(2,7,2)$

```
a = np.arange(10)
```

```
b = a[2:7:2]
print(b)
```

```
a = np.array([[1,2,3],
              [3,4,5],
              [4,5,6]])
```

```
print(a)
```

#All the rows of first column

```
print(a[:,1])
#[2 4 5]
```

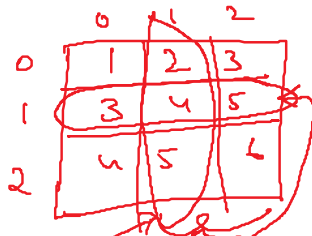
#All the columns of the first row

```
print(a[1,...])
#[3 4 5]
```

#All rows from 1 column to end

```
print(a[:,1:])
```

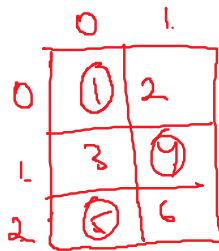
```
[[2 3]
 [4 5]
 [5 6]]
```



```
x = np.array([[1, 2],
              [3, 4],
              [5, 6]])
```

```
y = x[[0,1,2],
       [0,1,0]]
```

```
print(y)
#[1 4 5]
```



(0,0), (1,1) and (2,0) from the array

```
x = np.array([[0, 1, 2],
              [3, 4, 5],
              [6, 7, 8],
              [9, 10, 11]])
```

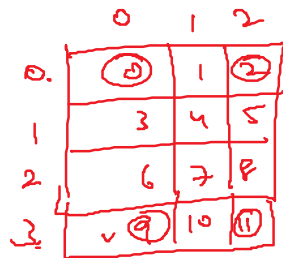
```
rows = np.array([[0,0],
                 [0,3]])
```

```
cols = np.array([[0,2],
                 [0,2]])
```

```
y = x[rows,cols]
```

```
print(y)
```

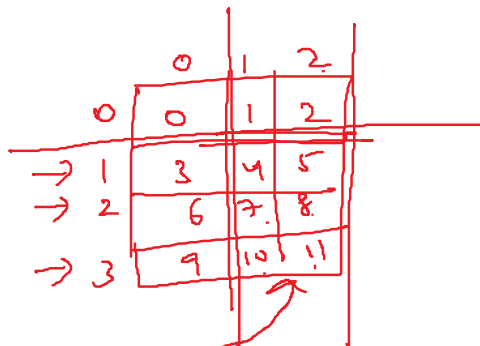
```
[[0 2]
 [9 11]]
```



```
x = np.array([[0, 1, 2],
              [3, 4, 5],
              [6, 7, 8],
              [9, 10, 11]])
```

```
z = x[1:4,1:3]
print(z)
```

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



```
[[ 4 5]
 [ 7 8]
 [10 11]]
```

```
y = x[1:4,[1,2]]
```

```
print( y)
[[ 4 5]
 [ 7 8]
 [10 11]]
```

```
x = np.array([[ 0, 1, 2],
              [ 3, 4, 5],
              [ 6, 7, 8],
              [ 9, 10, 11]])
print (x[x > 5])
```

```
#[ 6 7 8 9 10 11]
```

```
print (x[x >= 5])
#[ 5 6 7 8 9 10 11]
```

```
a = np.array([np.nan, 1,2,np.nan,3,4,5])
print(a)
#[nan 1. 2. nan 3. 4. 5.]
#Retrieve non-nan elements
print(a[~np.isnan(a)])
#[1. 2. 3. 4. 5.]
```

Reshape

```
numpy.reshape(arr, newshape, order')
```

```
a = np.arange(8)
print(a)
#[0 1 2 3 4 5 6 7]
```

```
b = a.reshape(4,2)
print(b)
```

*4 Rows
2 Cols
Row Major Arrangement*

```
→ [[0 1]
→ [2 3]
→ [4 5]
→ [6 7]]
```

```
np.reshape(a,(2,4))
array([[0, 1, 2, 3],
       [4, 5, 6, 7]])
```

```
np.reshape(a,(3,2))
#ValueError: cannot reshape array of size 8 into shape (3,2)
```

Row Major

```
arr = np.reshape(a,(4,2),'C')
print(arr)
```

```
0 1 2 3 4 5 6 7
```

```
arr = np.reshape(a,(4,2),'C')
print(arr)
```

Now.

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

Column major

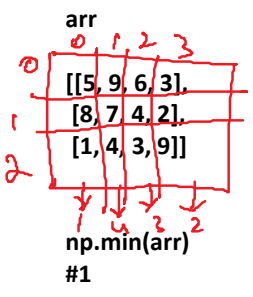
```
[[0 1]
 [2 3]
 [4 5]
 [6 7]]
```

```
arr = np.reshape(a,(4,2),'F')
print(arr)
```

```
[[0 4]
 [1 5]
 [2 6]
 [3 7]]
```

Axis 0 (Direction along Rows) –

- Axis 0 is called the first axis of the Numpy array.
- This axis 0 runs vertically downward along the rows of Numpy multidimensional arrays, i.e., performs **column-wise** operations.



```
np.min(arr,axis=0)
#array([1, 4, 3, 2])
```

Axis 1 (Direction along with columns) –

Axis 1 is called the second axis of multidimensional Numpy arrays. As a result, Axis 1 sums horizontally along with the columns of the arrays. It performs **row-wise** operations.

arr

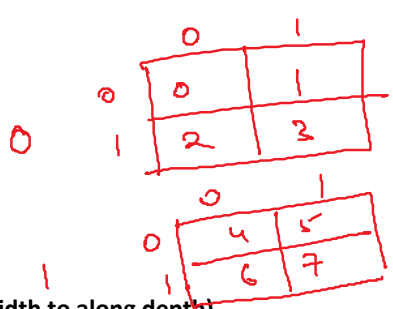
```
[[5, 9, 6, 3],
 [8, 7, 4, 2],
 [1, 4, 3, 9]]
```

```
np.max(arr,axis=1)
#array([9, 8, 9])
```

numpy.rollaxis(arr, axis, start)

```
a = np.arange(8).reshape(2,2,2)
print(a)
```

```
[[[0 1]
 [2 3]]
 [[4 5]
 [6 7]]]
```



```
## to roll axis-2 to axis-0 (along width to along depth)
print (np.rollaxis(a,2) )
```

```
[[[0 2]
 [4 6]]
```

```
[[[1 3]
 [5 7]]]
```

```
numpy.concatenate((a1, a2, ...), axis)
```

```
a = np.array([[1,2],[3,4]])
print(a)
```

```
b = np.array([[5,6],[7,8]])
print(b)
```

```
print(np.concatenate((a,b)))#by default axis=0
```

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

```
print (np.concatenate((a,b),axis = 0))
```

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

```
print(np.concatenate((a,b),axis = 1))
```

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

```
np.logspace(2, 3, 10, True)
#array([ 100.        , 129.1549665 , 166.81005372, 215.443469 ,
        278.25594022, 359.38136638, 464.15888336, 599.48425032,
        774.26368268, 1000.        ])
```