Base

In [3]:

```
import numpy as np
ac=np.array([12,23,45,34,56])
y=ac.copy()
x=ac.view()

print(y.base)
print(x.base)
```

```
None
[12 23 45 34 56]
```

Reshaping

In [9]:

```
ar= np.array([1,2,34,45])
n=ar.reshape(2,2)
print(n)
```

```
[[ 1  2]
 [34 45]]
```

iterationg array

In [11]:

```
ar= np.array([1,2,34,45,56,67])
for i in ar:
    print(i)
```

```
1
2
34
45
56
67
```

In [19]:

```
ar= np.array([1,2,34,45,56,67,78,89,98])
n=ar.reshape(3,3)
for i in n:
    print(n)
```

```
[[ 1  2 34]
 [45 56 67]
 [78 89 98]]
[[ 1  2 34]
 [45 56 67]
 [78 89 98]]
[[ 1  2 34]
 [45 56 67]
 [78 89 98]]
```

Join two arrays

In [20]:

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

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

In [25]:

```
a=np.array([[1,2,3],[4,5,6]])
b=np.array([[7,8,9],[10,11,12]])
print(a)
print(b)
con=np.concatenate((a,b),axis=0)
print(con)
```

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

In [26]:

```
a=np.array([[1,2,3],[4,5,6]])
b=np.array([[7,8,9],[10,11,12]])
print(a)
print(b)
con=np.concatenate((a,b),axis=1)
print(con)
```

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

Joining array using stack function

In [27]:

```
a=np.array([1,2,3])
b=np.array([4,5,6])
con=np.stack((a,b),axis=1)
print(con)
```

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

hstack()

In [28]:

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

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

vstack()

In [29]:

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

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

dstack()

In [30]:

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

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

Splitting array

In [31]:

```
import numpy as np
a1=np.array([1,2,3,4,5,6,7,8,9])
n=np.array_split(a1,5)
print(n)
```

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

Splitting 2-D array

In [34]:

```
import numpy as np
a1=np.array([[1,2,3],[4,5,6],[7,8,9],[10,11,12]])
n=np.array_split(a1,3)
print(n)
```

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

hsplit()

In [36]:

```
import numpy as np
a1=np.array([[1,2,3],[4,5,6],[7,8,9],[10,11,12]])
n=np.hsplit(a1,3)
print(n)
```

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

vsplit()

In [38]:

```
import numpy as np
a1=np.array([[1,2,3],[4,5,6],[7,8,9],[10,11,12]])
n=np.vsplit(a1,2)
print(n)
```

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

dsplit()

In [42]:

```
import numpy as np
a1=np.array([[[1,2,3],[4,5,6],[7,8,9]]])
n=np.dsplit(a1,3)
print(n)
```

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

Searching array

In [54]:

```
arr=np.array([78,45,43,23,89,85,65,80,64,80,44,32,44,45])
a=np.where(arr==44)
print(a)
```

```
(array([10, 12], dtype=int64),)
```

Replace element in array

In [56]:

```
arr=np.array([78,45,43,23,89,85,65,80,64,80,44,32,44,45])
a=np.arange(1,20)
print(a)
```

```
[ 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19]
```

missing values in an array

In [57]:

```
a=np.array([1,np.nan,78,86,90,np.nan,8,np.nan])
a
```

Out[57]:

```
array([ 1., nan, 78., 86., 90., nan,  8., nan])
```

In [58]:

```
a=np.array([1,np.nan,78,86,90,np.nan,8,np.nan])
np.isnan(a)
```

Out[58]:

```
array([False,  True, False, False, False,  True, False,  True])
```

In [60]:

```
a=np.array([1,np.nan,78,86,90,np.nan,8,np.nan])
a[np.isnan(a)]=89
np.isnan(a).any()
a
```

Out[60]:

```
array([ 1., 89., 78., 86., 90., 89.,  8., 89.])
```

In [61]:

```
a=np.array([1,np.nan,78,86,90,np.nan,8,np.nan])
a[np.isnan(a)]=89
np.isnan(a).any()
```

Out[61]:

```
False
```

common items between two arrays

In [64]:

```
a=np.array([10,11,12,13,14,15])
b=np.array([12,13,45,14,15,11])
np.intersect1d(a,b)
```

Out[64]:

```
array([11, 12, 13, 14, 15])
```

Remove common elements

In [65]:

```
a=np.array([10,11,12,13,14,15])
b=np.array([12,13,45,14,15,11])
np.setdiff1d(a,b)
```

Out[65]:

```
array([10])
```

Reverse an array

In [66]:

```
a=np.array([10,11,12,13,14,15])
np.flip(a)
```

Out[66]:

```
array([15, 14, 13, 12, 11, 10])
```

sort array

In [67]:

```
a=np.array([10,11,12,13,14,15])
np.sort(a)
```

Out[67]:

```
array([10, 11, 12, 13, 14, 15])
```

In [ ]:

```
```

In [ ]:

```
```