

```
In [20]: import numpy as np
arr=np.array([[5.24,3.28,6.99],[3.24,5.82,2.39],[2.54,3.39,6.39]])
# to display the sum of of all elments , sum of each columnn , sum of each row
sum_ele=arr.sum()
sum_row=arr.sum(axis=0)
sum_col=arr.sum(axis=1)
print("the sum of of all elments:--",sum_ele)
print("the sum of of each row :--",sum_row)
print("the sum of of each columns :--",sum_col)
```

```
the sum of of all elments:-- 39.28
the sum of of each row :-- [11.02 12.49 15.77]
the sum of of each columns :-- [15.51 11.45 12.32]
```

```
In [15]: # Substitute elements of the above array which are equal to 5.32 with 15.32
arr1=arr
for i in range(len(arr1)):
    for j in range(len(arr1[i])):
        if (arr1[i][j]==5.32):
            arr1[i][j]=15.32
print(" After Substituting elements of the above array which are equal to 5.32 with 15.32")
print(arr1)
```

```
After Substitute elements of the above array which are equal to 5.32 with 15.32
[[5.24 3.28 6.99]
 [3.24 5.82 2.39]
 [2.54 3.39 6.39]]
```

```
In [16]: # Substitute elements of the above array which are less than 5.32 with 15.32
arr2=arr
for i in range(len(arr2)):
    for j in range(len(arr2[i])):
        if (arr2[i][j]<5.32):
            arr2[i][j]=15.32
print(" After Substituting elements of the above array which are less than 5.32 with 15.32")
print(arr2)
```

```
After Substitute elements of the above array which are less than 5.32 with 15.32
[[15.32 15.32  6.99]
 [15.32  5.82 15.32]
 [15.32 15.32  6.39]]
```

```
In [21]: # Substitute elements of the above array which are greater than 5.32 with 15.32
```

```

arr_gr=arr
for i in range(len(arr_gr)):
    for j in range(len(arr_gr[i])):
        if (arr_gr[i][j]>5.32):
            arr_gr[i][j]=15.32
print(" After Substituting elements of the above array which are greater than 5.32 with 15.32")
print(arr_gr)

```

After Substitute elements of the above array which are greater than 5.32 with 15.32

```

[[ 5.24  3.28 15.32]
 [ 3.24 15.32  2.39]
 [ 2.54  3.39 15.32]]

```

In [35]: *#Sorting row and column in ascending order*

```

arr=np.array([[5.24,3.28,6.99],[3.24,5.82,2.39],[2.54,3.39,6.39]])
arr_sc=np.sort(arr,axis=0)
print("Sorting the array by column:\n",arr_sc)
arr_sr=np.sort(arr,axis=1)
print("Sorting the array by row:\n",arr_sr)

```

Sorting the array by column:

```

[[2.54 3.28 2.39]
 [3.24 3.39 6.39]
 [5.24 5.82 6.99]]

```

Sorting the array by row:

```

[[3.28 5.24 6.99]
 [2.39 3.24 5.82]
 [2.54 3.39 6.39]]

```

In [37]: *#Split the array into two arrays along the second axis*

```

narr=np.arange(16)
narr=np.reshape(narr,(4,4))
hnarr=np.hsplit(narr,(3,6))
print("Splitting the above array into two arrays along the second axis")
print(hnarr)

```

Splitting the above array into two arrays along the second axis

```

[array([[ 0,  1,  2],
        [ 4,  5,  6],
        [ 8,  9, 10],
        [12, 13, 14]]), array([[ 3],
        [ 7],
        [11],
        [15]])], array([], shape=(4, 0), dtype=int32))

```

```
In [33]: #to sort the complex array using the real part first, then the imaginary part.
arr_cp = [(2+3j), (4-1j), (2-2j), (4-3j), (3+5j)]
print("given array:--")
print(arr_cp)
print("complex array using the real part first, then the imaginary part.")
print(np.sort_complex(arr_cp))
```

given array:
 [(2+3j), (4-1j), (2-2j), (4-3j), (3+5j)]
 complex array using the real part first, then the imaginary part.
 [2.-2.j 2.+3.j 3.+5.j 4.-3.j 4.-1.j]

```
In [39]: #to sort the array on height by assuming an structured array from given set of name, height, class and their data types
dtype = [('name', 'S115'), ('height', int), ('age', float)]
values = [('John', 6, 52.5 ), ('Naught', 6, 48.5 ), ('Prince', 3, 41.1 ), ('Paul', 4, 43.11)]
# creating a structured array
a = np.array(values, dtype=dtype)
print("before sorting")
print(a)
print("After sorting")
print(np.sort(a, order='height'))
```

before sorting
 [(b'John', 6, 52.5) (b'Naught', 6, 48.5) (b'Prince', 3, 41.1)
 (b'Paul', 4, 43.11)]
 After sorting
 [(b'Prince', 3, 41.1) (b'Paul', 4, 43.11) (b'John', 6, 52.5)
 (b'Naught', 6, 48.5)]

```
In [ ]:
```