

```
[103] # Operation on numpy array
      #Addition of elements of 1D array

      import numpy as np
      a = np.array([1,2,3,4]) # 1D array
      sum(a) # sum of all elements of array
```

10

```
[110] a = np.arange(8) #array elements from 0 to 7
      a
```

array([0, 1, 2, 3, 4, 5, 6, 7])

```
[111] sum(a) # sum of all elements of array
```

28

```
▶ #sum of two 1D arrays
arr = np.array( [1,2,3,4] ) # 1D array
arr1 = np .array( [1,2,3,4] ) # 1D array
sum(arr, arr1) # sum of two arrays
```

```
[131] #Addition of elements of 2D array
      a = np.array ( [ [1,2,3,4], [5,6,7,8] ] )
      a.sum() # sum of all elements of array
```

36

```
✓ [132] a. sum ( axis = 0 ) # column wise sum of all elements of array
js
```

array([6, 8, 10, 12])

```
✓ [130] a. sum (axis = 1) # row wise sum of all elements of array
js
```

array([10, 26])

```
✓ ▶ #sum of two 2D array
js
arr = np.array( [[1,2,3,4], [5,6,7,8] ]) # 2D array
arr1 = np.array( [[1,2,3,4], [5,6,7,8] ]) # 2D array
sum(arr, arr1) # sum of two 2D arrays
```

array([[7, 10, 13, 16],
 [11, 14, 17, 20]])

```
np.add(arr, arr1) # for addition add() also can be used
```

```
array([[ 2,  4,  6,  8],  
       [10, 12, 14, 16]])
```

```
[148] arr = np.array( [1,2,3,4] ) # 1D array  
      np.add(a,2)      # we can add any constant value to array
```


```
array([3, 4, 5, 6])
```

```
[142] #Subtraction of elements of 1D array  
      arr = np.array( [10,20,30,40] ) # 1D array  
      arr1 = np.array( [[1,2,3,4] ] ) # 1D array  
      np.subtract(arr, arr1) # # subtraction of elements of two 1D arrays
```

```
array([[ 9, 18, 27, 36]])
```

```
[147] #Subtraction of elements of 2D array  
      a1 = np.array([[2, -4, 5], [-6, 2, 0]]) # 2D array  
      a2 = np.array([[0, -7, 5], [5, -2, 9]]) # 2D array  
      np.subtract(a1, a2) # subtraction of elements of two 2D arrays
```

```
array([[ 2,  3,  0],  
       [-11,  4, -9]])
```

```
5  arr = np.array( [10,20,30,40] ) # 1D array  
np.subtract(arr, 5)      # we can subtract any constant value to array
```

```
array([ 5, 15, 25, 35])
```

```
✓ [158] #Multiplication of elements of 1D array
0s arr = np.array( [1,2,3] ) # 1D array
np.multiply(arr, 3) # multiplication of any element with the elements of 1D arrays

array([3, 6, 9])
```

```
✓ [159] #Multiplication of elements of 2D array
0s arr = np.array( [[1,2,3], [5,6,7]] ) # 2D array
np.multiply(arr, 2) # multiplication of any element with the elements of 2D arrays

array([[ 2,  4,  6],
       [10, 12, 14]])
```

```
[159] #Multiplication of elements of two 1D array
arr = np.array( [10,20,30,40] ) # 1D array
arr1 = np.array( [1,2,3,4] ) # 1D array
np.multiply(arr, arr1) # multiplication of elements of two 1D arrays

array([[ 10,  40,  90, 160]])
```

```
✓ [161] #Multiplication of elements of two 2D array
1s arr = np.array([[10,20,30,40], [2,4,6,8]] ) # 2D array
arr1 = np.array( [[1,2,3,4], [1,2,3,4]] ) # 2D array
np.multiply(arr, arr1) # multiplication of elements of two 2D arrays

array([[ 10,  40,  90, 160],
       [  2,   8,  18,  32]])
```

```
✓ [166] #Division of elements of 1D array
0s arr = np.array( [10,20,30] ) # 1D array
np.divide(arr, 5) # Division of any element with the elements of 1D arrays

array([2., 4., 6.])
```

```
[167] arr = np.array( [[10,20,30], [50,60,70]] ) # 2D array
np.divide(arr, 2) # division of any element with the elements of 2D array

array([[ 5., 10., 15.],
       [25., 30., 35.]])
```

```
[168] arr = np.array( [10,20,30,40] ) # 1D array
arr1 = np.array( [2,10,15,5] ) # 1D array
np.divide(arr, arr1) # division of elements of two 1D arrays

array([[5., 2., 2., 8.]])
```

```

▶ arr = np.array([[10,20,30,40], [2,4,6,8]] ) # 2D array
arr1 = np.array( [[2,2,3,4], [2,2,3,4]] ) # 2D array
np.divide(arr, arr1) # division of elements of two 2D arrays

array([[ 5., 10., 10., 10.],
       [ 1.,  2.,  2.,  2.]])

```

```

[170] #Use of Power()
arr = np.array( [10,20,30,40] ) # 1D array
np.power(arr, 3)

array([ 1000,  8000, 27000, 64000])

```

```

▶ arr = np.array([[10,20,30,40], [2,4,6,8]] ) # 2D array
np.power(arr, 2)

⇒ array([[ 100,  400,  900, 1600],
        [  4,   16,   36,   64]])

```

```

✓ 0s ▶ arr = np.array( [10,20,30,40] ) # 1D array
arr1=np.array([1,2,3,4])
np.power(arr, arr1 )

array([    10,    400, 27000, 2560000])

```

```

✓ 0s [174] #Use of mod()
arr = np.array( [10,20,30,40] ) # 1D array
np.mod(arr, 3)

array([1, 2, 0, 1])

```

```

✓ 0s [175] arr = np.array([[10,20,30,40], [2,4,6,8]] ) # 2D array
np.mod(arr, 2)

array([[0, 0, 0, 0],
       [0, 0, 0, 0]])

```

```

✓ 0s ▶ arr = np.array( [10,20,30,40] ) # 1D array
arr1=np.array([3,7,3,5])
np.mod(arr, arr1 )

⇒ array([1, 6, 0, 0])

```

```
✓ [180] #Use of reciprocal() - returns the reciprocal of argument, element-wise.  
0s #For elements with absolute values larger than 1, the result is always 0 and for integer 0, overflow warning is issued.  
arr = np.array( [25,2.2,34,46] ) # 1D array  
np.reciprocal(arr)  
  
array([0.04      , 0.45454545, 0.02941176, 0.02173913])
```

```
✓ [180] # 2D array  
0s arr = np.array([[13,2.5,3,42], [1,7,4.1,8]] ) # 2D array  
np.reciprocal(arr)  
  
array([[0.07692308, 0.4      , 0.33333333, 0.02380952],  
       [1.      , 0.14285714, 0.24390244, 0.125      ]])
```

```
✓ [200] # Use of min(), max()  
0s arr = np.array([110,22,37,40]) # 1D array  
np.max(arr)  
  
110
```

```
✓ [199] arr = np.array([110,22,37,40]) # 1D array  
0s np.min(arr)  
  
22
```

```
✓ [207] arr = np.array([110,22,37,40]) # 1D array  
0s np.min(arr, axis=0)  
  
22
```


```
✓ [207] arr = np.array([110,22,37,40]) # 1D array  
0s np.max(arr, axis=0)  
  
110
```


✓
0s [202] arr = np.array([[10,20,30,40], [2,4,6,8]]) # 2D array
np.max(arr, axis=0)

array([10, 20, 30, 40])

✓
0s [203] arr = np.array([[10,20,30,40], [2,4,6,8]]) # 2D array
np.min(arr, axis=0)

array([2, 4, 6, 8])

✓
0s  arr = np.array([[10,20,30,40], [2,4,6,8]]) # 2D array
np.min(arr, axis=1)

 array([10, 2])

✓
0s [205] arr = np.array([[10,20,30,40], [2,4,6,8]]) # 2D array
np.max(arr, axis=1)

array([40, 8])