> Creating Array

```
>>> a = np.array ([1, 2, 3])
>>> b = np.array ([(1.5, 2, 3), (4, 5, 6)], dtype = float)
>>> c = np.array ([(1.5, 2, 3), (4, 5, 6)], (4, 5, 6)], dtype = float)
```

Saving & Loading Text Files

```
>>> np.loadtxt (“myfile.txt”)
>>> np.genfromtxt (“my_file.csv”, delimiter = ’,’)
>>> np.savetxt (“myarray.txt”, a,  delimiter = “  “)
```

> Sorting Arrays

```
>>> a.sort ( ) #sort array
>>> c.sort (axis = ⊙) #sort the elements of an array’s axis
```

Slicing

```
>>> a [0:2] # Select items at index 0 and 1
    array ([1, 2])
>>> b [0:2, 1] # Select items at row 0 and 1 in column 1
    array ([2. , 5. ])
>>> c [1, ...] # Same as [1, : , :]
    array ([[ 3., 2., 1., ],
                [ 4., 5., 6., ]]]),
>>> a [:   : - 1] # Reversed array a array ([3, 2 ,1)]
```

Adding/Removing Elements

```
>>>  h.resize(( 2, 6))  # Return a new array with shape (2, 6)
>>>  np.append (h , g)  # Append items to an array
>>>  np.insert (a, 1, 5)  # Inserts items in an array
>>>  np.delete (a, [1])  # Delete items from an array
```

Combining Arrays

```
>>>  np.concatenate((a, d), axis = 0) #concatenate arrays
      Array ([1, 2, 3, 10, 15, 2 0])
```

STATISTICS

np.mean (arr,axis=0)	- Returns mean along specific axis
arr.sum()	- Returns sum of **arr**
arr.min()	- Returns minimum value of **arr**
arr.max (axis=0)	- Returns maximum value of specific axis
np.var(arr)	- Returns the variance of array
np.std(arr,axis=1)	- Returns the standard deviation of specific axis
arr.corrcoef( )	- Returns correlation coefficient of array

> Inspecting Your Array

```
>>> a. shape                 #Array dimensions
>>> len(a)                   #Length of array
>>> b.ndim                   #Number of array dimensions
>>> e. size                  #Number of array elements
>>> b.dtype                  #Data type of array elements
>>> b.dtype.name             #name of Data type
>>> b.astype(int)            #Convert on array to a different type
```

> Array Mathematics

Arithmetic Operations

```
>>> g = a – b #Subtraction
array ([[ -0.5, 0., 0. ], 0. ],
          [ -3. , -3. , -3. ]])
>>>np.subtract (a, b) # Subtraction
```

```
>>> b + a  Addition
        array ([[[ 2.5, 4., 6., ],
                    [5. , 7. , 9. ,]]])
>>> np.add (b, a) Addition
>>> a/ b # Division
array ([[ θ . 66666667, 1., 1., 1],
        [θ. 25, θ.4, θ.5]])
>>>np.divide(a ,b) #Division
>>> a * b #Multiplication
array ([[ 1.5, 4., 9.],
          [4 , 10, 18]])
>>> np.multiply (a, b) #Multiplication
>>> np.exp(b) #Exponentiation
>>> np.sqrt(b) #Square root
>>> np.sin (a) #Print sines of an array
>>> np.cos (b) #Element – wise cosine
>>> np.log (a)  #Element – wise natural logarithm
>>> e.dot (f) Dot Product
array ([[ 7., 7.])
          [ 7., 7. ]])
```

Aggregate Functions

```
>>> a. sum( )              #Array-wise sum
>>> a.min( )               #Array-wise minimum value
>>> b.max(axis=0)          #Maximum value of an array row
>>> b.cumsum (axis=1)      #Cumulative sum of the elements
>>> a.mean ( )                    #Mean
>>> np.median(b)           #Median
```

```
>>> np.corrcoef(a)              #Correlation coefficient
>>> np. std(b)                          #Standard deviation
import statistics as stats
a = np.array([1,3,5,7,2,5,9,2,2])
print(stats.mode(a))
# mode is reoccuring values
```