

## ✓ Numpy Library

```
import numpy as np;
```

## ✓ Arrays

```
a = np.array([1,2,3])
print("1D Array--")
print(a)
```

```
↔ 1D Array--
[1 2 3]
```

```
b = np.array([(1,2,3), (4,5,6)])
print("2D Array--")
print(b)
```

```
↔ 2D Array--
[[1 2 3]
 [4 5 6]]
```

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

```
print("3D Array--")
print(c)
```

```
↔ 3D Array--
[[[1 2 3]
  [4 5 6]]

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

```
print("Dimensions: ", np.ndim(c))
print("Shape: ", np.shape(c))
print("Size: ", np.size(c))
```

```
↔ Dimensions: 3
Shape: (2, 2, 3)
Size: 12
```

```
z = np.zeros((3,3), dtype=int)
print("Zeros Array")
print(z)
```

```
↔ Zeros Array
[[0 0 0]
 [0 0 0]
 [0 0 0]]
```

```
o = np.ones((3,3),dtype=int)
print("Ones Array")
print(o)
```

```
↔ Ones Array
[[1 1 1]
 [1 1 1]
 [1 1 1]]
```

```
d = np.arange(10,50,5)
print("Arange - ", d);
```

```
↔ Arange - [10 15 20 25 30 35 40 45]
```

```
temp = np.linspace(0,2,9)
print("Temp-",temp)
```

```
↔ Temp- [0.    0.25 0.5   0.75 1.    1.25 1.5   1.75 2.   ]
```

```
np.random.random((2,2))
```

```
array([[0.03150856, 0.99566822],
       [0.76911817, 0.07544317]])
```

## File

```
np.save('my_array', a)
np.savez('array.npz', a, b)
Text = np.load('array.npz')
print(Text)
```

```
# myFile = np.loadtxt("/content/Asif")
# print(myFile)
```

```
NpzFile 'array.npz' with keys: arr_0, arr_1
```

## Arithmetic Operation

```
add = np.add(z,o)
print(add)
```

```
[[1 1 1]
 [1 1 1]
 [1 1 1]]
```

```
sub = np.subtract(z,o)
print(sub)
```

```
[[ -1 -1 -1]
 [ -1 -1 -1]
 [ -1 -1 -1]]
```

```
div = np.divide(b,a)
print(div)
```

```
[[1.  1.  1. ]
 [4.  2.5 2.  ]]
```

```
mul = np.multiply(z,o)
print(mul)
```

```
[[0 0 0]
 [0 0 0]
 [0 0 0]]
```

```
max = np.max(a)
print(max)
```

```
3
```

```
min = np.min(a)
print(min)
```

```
1
```

```
exp = np.exp(5)
print(exp)
```

```
148.4131591025766
```

```
sqrt = np.sqrt(16)
print(sqrt)
```

```
4.0
```

```
dot = np.dot(add,sub)
print(dot)
```

```
[[ -3 -3 -3]
 [ -3 -3 -3]
 [ -3 -3 -3]]
```

## ✓ Comparision

```
sub == add
```

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

```
a < 2
```

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

```
np.array_equal(a, b)
```

```
→ False
```

## ✓ Aggregate Functions

```
print("Sum of array a: ", a.sum())
print("Min from array a: ", a.min())
print("Max from array b: ", b.max(axis=1))
print("Mean of array a: ", a.mean())
# print("Median of array a: ", b.median())
```

```
→ Sum of array a: 6
   Min from array a: 1
   Max from array b: [3 6]
   Mean of array a: 2.0
```

## ✓ Copying and Sorting

```
g = np.copy(a)
h = a.copy()
print(g)
print(h)
```

```
→ [1 2 3]
   [1 2 3]
```

```
a1 = np.array([7,8,1,9,12,0,13])
a1.sort()
b.sort(axis=0)
print("A1 after sorting: ", a1)
print("\nB array after sorting\n", b,)
```

```
→ A1 after sorting: [ 0  1  7  8  9 12 13]

   B array after sorting
   [[1 2 3]
    [4 5 6]]
```

## ✓ Subsetting, Slicing, Indexing

```
print("Subsettings of different arrays: ", a[2], b[1,2])
print("Slicing from index 2 to 5: ", a[2:6])
print("Slicing all element except last one: ", a[:-1])
print("Slicing 2d array: \n", b[:2])
```

```
→ Subsettings of different arrays: 3 6
   Slicing from index 2 to 5: [ 7  8  9 12]
   Slicing all element except last one: [ 0  1  7  8  9 12]
   Slicing 2d array:
   [[1 2 3]
    [4 5 6]]
```

```
print("Boolean Indexing: ", a1[a1>=8])
```

```
→ Boolean Indexing: [ 8  9 12 13]
```

## ✓ Other Manipulation

```
t = np.transpose(b)
print("Transpose of array b: \n", t)
print("\nReshape of t array: \n", t.reshape(1,6))
np.append(b,a)
print("\nB array after adding a to it\n", b)
a = np.insert(a, 1, 5)
print("\nA array after adding element to it\n", a)
a = np.delete(a,[1])
print("\nA after deleting element from it\n", a)
```

```
↳ Transpose of array b:
[[1 4]
 [2 5]
 [3 6]]
```

```
Reshape of t array:
[[1 4 2 5 3 6]]
```

```
B array after adding a to it
[[1 2 3]
 [4 5 6]]
```

```
A array after adding element to it
[1 5 2 3]
```

```
A after deleting element from it
[1 2 3]
```

```
aa1 = np.concatenate((a, a1), axis=0)
print(aa1)
```

```
↳ [ 1  2  3  0  1  7  8  9 12 13]
```