

```

# Name : Atharv Sakharam Patil
# Roll No. : 105
# PRN No. : UIT23M1105
# Title : Assignment based on Numpy arrays and functions (Numpy
Library)

import pandas as pd
import numpy as np

arr1=np.array([1,2,3]) # 1D Array.
arr1
array([1, 2, 3])

arr2=np.array([[1,2],[3,4]]) # 2D Array.
arr2
array([[1, 2],
       [3, 4]])

arr3=np.zeros((2,2)) # Array filled with zeros
arr3
array([[0., 0.],
       [0., 0.]])

arr4=np.ones((3,3)) # Array filled with ones
arr4
array([[1., 1., 1.],
       [1., 1., 1.],
       [1., 1., 1.]])

arr5=np.full((2,2),5) # Array filled with constant values
arr5
array([[5, 5],
       [5, 5]])

arr6=np.arange(0,10,2) # Array with evenly spaced values.
arr6
array([0, 2, 4, 6, 8])

arr7=np.linspace(0,1,5) # Array with specific number of equally spaced
values within limits
arr7
array([0. , 0.25, 0.5 , 0.75, 1.  ])

print(arr2+arr5) # Array Addition

```

```

[[6 7]
 [8 9]]

print(arr2-arr5) # Array subtraction

[[-4 -3]
 [-2 -1]]

print(arr2*arr5) # Array Multiplication

[[ 5 10]
 [15 20]]

print(arr2/arr5) # Array Division

[[0.2 0.4]
 [0.6 0.8]]

print(arr2**2) # Array Power

[[ 1  4]
 [ 9 16]]

print(np.sum(arr2)) # Sum of Elements
10

print(np.mean(arr2)) # Mean of Elements
2.5

print(np.median(arr2)) # Median of Elements
2.5

print(np.std(arr2)) # Standard Deviation of Elements
1.118033988749895

print(np.var(arr2)) # Variance of Elements
1.25

print(np.min(arr2)) # Smallest Value
1

print(np.max(arr2)) # Largest Value
4

print(np.prod(arr2)) # Product of Value
24

```

```

print(np.cumsum(arr2)) # Commulative Sum
[ 1  3  6 10]
arr1.size # Total No. of elements
3
arr1.shape # Dimension as a Tuple
(3,)
arr1.ndim # No. of Dimensions
1
arr3.dtype # Returns Datatype
dtype('float64')
arr1.itemsize # Return the size in byte of each element
8
print(arr1[2]) # Access element of specified index
3
print(arr1[1:3]) # Extract a subportion i.e Subarray
[2 3]
print(arr1[-1]) # Access element of specified index from end
3
print(arr1[arr1>2]) # Access element based on condition
[3]
arr10=np.arange(0,10,2) # Generate values with fixed step size
arr10
array([0, 2, 4, 6, 8])
arr11=np.arange(6).reshape(2,3) # Changes the shape of array without
changing data
arr11
array([[0, 1, 2],
       [3, 4, 5]])
print(np.argmax(arr11)) # Return the index of maximum element
5

```

```
print(np.argmin(arr11)) # Return the index of minimum element
```

```
0
```

```
print(np.diff(arr11)) # Computes Succesive Difference
```

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