

▼ Assignment 4

Name: [Mukesh Kumar Mahto](#)

Reg_ID: GO_STP_9639

Question on Numpy-

1. Import the numpy package under the name np and Print the numpy version and the configuration

```
import numpy as np
print(np.__version__)
print(np.__config__)

1.19.5
<module 'numpy.__config__' from '/usr/local/lib/python3.7/dist-packages/num
```

2. Create a null vector of size 10

```
a = np.zeros(10)
a

array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.]
```

3. Create Simple 1-D array and check type and check data types in array

```
a = np.array([1, 2, 3, 4, 5])
print(a)
print(type(a))
print(a.dtype)

[1 2 3 4 5]
<class 'numpy.ndarray'>
int64
```

4. How to find number of dimensions, bytes per element and bytes of memory used?

```
# import sys
a = np.array([1, 2, 3, 4, 5])
print(a.ndim)
print(a.itemsize)
print(a.size)
```

✓ 0s completed at 8:39 AM



5. Create a null vector of size 10 but the fifth value which is 1

```
a = np.zeros(10)
a[4] = 1
a

array([0., 0., 0., 0., 1., 0., 0., 0., 0., 0.] )
```

6. Create a vector with values ranging from 10 to 49

```
a = np.arange(10, 50)
a

array([10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26,
       27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43,
       44, 45, 46, 47, 48, 49])
```

Double-click (or enter) to edit

7. Reverse a vector (first element becomes last)

```
a = np.arange(10)
print(f"Original vector: {a}")
print(f"Reverse of vector: {a[::-1]}")

Original vector: [0 1 2 3 4 5 6 7 8 9]
Reverse of vector: [9 8 7 6 5 4 3 2 1 0]
```

8. Create a 3x3 matrix with values ranging from 0 to 8

```
a = np.arange(9)
print(a.reshape(3,3))

[[0 1 2]
 [3 4 5]
 [6 7 8]]
```

9. Find indices of non-zero elements from [1,2,0,0,4,0]

```
[[1. 0. 0.]  
 [0. 1. 0.]  
 [0. 0. 1.]]
```

```
print(np.identity(3))
```

```
[[1. 0. 0.]  
 [0. 1. 0.]  
 [0. 0. 1.]]
```

11. Create a 3x3x3 array with random values

```
import random
```

```
a = np.random.random((3,3,3))
```

```
a
```

```
array([[[0.84020113, 0.03525898, 0.9205802 ],  
        [0.22519112, 0.47765508, 0.74878103],  
        [0.97680007, 0.70463439, 0.72354642]],  
       [[0.30942836, 0.22357287, 0.97681141],  
        [0.91058428, 0.12795474, 0.17090333],  
        [0.98882491, 0.95181726, 0.00419013]],  
       [[0.86601424, 0.9143214 , 0.64009522],  
        [0.42285472, 0.74097769, 0.02985548],  
        [0.21765818, 0.08612381, 0.884381  ]]])
```

12. Create a 10x10 array with random values and find the minimum and maximum values

```
import random
```

```
a = np.random.random((10,10))
```

```
print(a)
```

```
print(f"Maximum value: {a.max()} \nMinimum value: {a.min()}")
```

```
[[0.20667186 0.76923231 0.82427197 0.45352815 0.57283222 0.57224387  
 0.13946947 0.39789375 0.46023537 0.96452196]
```

13. Create a random vector of size 30 and find the mean value

```
import random
a = np.random.random(30)
print(a)
print(f"Mean: {a.mean()}")
```

[0.88392407 0.24573931 0.19184607 0.03579218 0.79517466 0.40642126
0.03240841 0.61509461 0.48459853 0.67064855 0.97917959 0.57891286
0.77518601 0.71691213 0.94833906 0.7292446 0.02016484 0.58306753
0.31787812 0.68844226 0.43927064 0.37398815 0.90679553 0.74879992
0.79836384 0.78143514 0.76300191 0.2638225 0.46903692 0.16348956]
Mean: 0.5468992920164042

14. Create a 2d array with 1 on the border and 0 inside

```
a = np.ones((5,6))
print(f"Original matrix:\n {a}")
a[1:-1,1:-1] = 0
print(f"\nMatrix after modification:\n {a}")
```

Original matrix:

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

Matrix after modification:

```
[[1. 1. 1. 1. 1. 1.]
 [1. 0. 0. 0. 0. 1.]
 [1. 0. 0. 0. 0. 1.]
 [1. 0. 0. 0. 0. 1.]
 [1. 0. 0. 0. 0. 1.]]
```

16. How to Accessing, changing specific elements, rows, columns, etc in Numpy array?

Example - `[[1 2 3 4 5 6 7] [8 9 10 11 12 13 14]]`

Get 13, get first row only, get 3rd column only, get [2, 4, 6], replace 13 by 20

```
a = [[1, 2, 3, 4, 5, 6, 7],[8, 9, 10, 11, 12, 13, 14]]
a[1][5] = 20
a
[[1, 2, 3, 4, 5, 6, 7], [8, 9, 10, 11, 12, 20, 14]]
```

17. How to Convert a 1D array to a 2D array with 2 rows

```
a = np.arange(10)
a.reshape(2,5)
array([[0, 1, 2, 3, 4],
       [5, 6, 7, 8, 9]])
```

Create the following pattern without hardcoding. Use only numpy functions and the below input array a.

Input:

```
import time

size = 1000000
list1 = range(size)
list2 = range(size)
array1 = np.arange(size)
array2 = np.arange(size)

initialTime = time.time()
resultantList = [(a * b) for a, b in zip(list1, list2)]
print("Time taken by Lists to perform multiplication:",(time.time() - initialTim

initialTime = time.time()
resultantArray = array1 * array2
print("Time taken by NumPy Arrays to perform multiplication:",(time.time() - ini

    Time taken by Lists to perform multiplication: 0.13395905494689941 seconds
    Time taken by NumPy Arrays to perform multiplication: 0.003741741180419922
```

