NUMPY_

May 17, 2023

```
[1]: import numpy as np
```

1 CREATING ARRAY

```
[2]: array = np.array([1,2,3,4,5,6,7,8])
     print(array)
```

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

2 CREATING MATRIX

```
[3]: matrix = np.array([[1,2,3],[4,5,6],[7,8,9]])
     matrix
```

```
[3]: array([[1, 2, 3],
            [4, 5, 6],
            [7, 8, 9]])
```

3 TRANSPOSING MATRIX

```
[4]: matrix.T
```

```
[4]: array([[1, 4, 7],
            [2, 5, 8],
            [3, 6, 9]])
```

4 SQUARE OF NUMBERS

```
[5]: #underroot of 25
     square = np.sqrt(25)
     square
```

```
[5]: 5.0
```

```
[6]: square1 = np.square(5)
     square1
```

[6]: 25

5 ARRANGE NUMBERS

[7]:
```
arrange = np.arange(1,100)
arrange
```

[7]:
```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 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, 50, 51,
       52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68,
       69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85,
       86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99])
```

6 LINESPACE

[8]:
```
linespace = np.linspace(1,10,2)
linespace
```

[8]:
```
array([ 1., 10.])
```

7 RESHAPING

[9]:
```
#we do reshping of arrays to shape it into matrix form
array.reshape(4,2)
```

[9]:
```
array([[1, 2],
       [3, 4],
       [5, 6],
       [7, 8]])
```

8 ZEROS

[10]:
```
zeros = np.zeros([4,4])
zeros
```

[10]:
```
array([[0., 0., 0., 0.],
       [0., 0., 0., 0.],
       [0., 0., 0., 0.],
       [0., 0., 0., 0.]])
```

9 ONES

```
[11]: ones = np.ones([4,5])
      ones
```

```
[11]: array([[1., 1., 1., 1., 1.],
             [1., 1., 1., 1., 1.],
             [1., 1., 1., 1., 1.],
             [1., 1., 1., 1., 1.]])
```

10 IDENTITY MATRIX

```
[12]: ident = np.eye(5,4)
      ident
```

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

11 UNIQUE

```
[13]: variable = np.array([1,2,3,4,3,2,1,4,"john","love","got","love","dare"])
      variable
```

```
[13]: array(['1', '2', '3', '4', '3', '2', '1', '4', 'john', 'love', 'got',
             'love', 'dare'], dtype='<U11')
```

```
[14]: np.unique(variable)
```

```
[14]: array(['1', '2', '3', '4', 'dare', 'got', 'john', 'love'], dtype='<U11')
```

12 Mathematical Operations of 2 Arrays

```
[15]: array2 = np.array([4,5,6,7,2,7,4,8])
      array3 = array+array2
      array3
```

```
[15]: array([ 5,  7,  9, 11,  7, 13, 11, 16])
```

```
[16]: array4 = array*array2
      array4
```

```
[16]: array([ 4, 10, 18, 28, 10, 42, 28, 64])
```

```
[17]: array5 = array-array2
      array5
```

```
[17]: array([-3, -3, -3, -3,  3, -1,  3,  0])
```

```
[18]: array6 = array/array2
      array6
```

```
[18]: array([0.25      , 0.4       , 0.5       , 0.57142857, 2.5       ,
             0.85714286, 1.75      , 1.        ])
```

13 RANDOM NUMBER

```
[19]: random = np.random.randn(4,3)
      random
```

```
[19]: array([[-0.74154704,  1.19339736,  0.66206974],
             [ 0.22596869, -0.48155268,  2.50867991],
             [ 0.08593274,  1.29308431,  0.55213082],
             [ 0.96280469, -1.82256628, -1.53452861]])
```

```
[20]: random1 = np.random.rand(4,3)
      random1
```

```
[20]: array([[0.36277475, 0.20104252, 0.89031452],
             [0.64191552, 0.90491527, 0.2220023 ],
             [0.14379107, 0.9454914 , 0.91635241],
             [0.97100933, 0.17924924, 0.51096554]])
```

14 RANDOM NUMBERS GENERATION AND SORTING

```
[21]: random_number = np.random.randint(10,20, size=10)
      random_number
```

```
[21]: array([17, 11, 11, 11, 16, 11, 10, 14, 12, 19])
```

```
[22]: np.sort(random_number)
```

```
[22]: array([10, 11, 11, 11, 11, 12, 14, 16, 17, 19])
```

```
[23]: random>random1
```

```
[23]: array([[False,  True, False],
             [False, False,  True],
             [False,  True, False],
             [False, False, False]])
```

[24]:

```
np.max(random)
```

[24]:

```
2.5086799078227964
```

[25]:

```
np.min(random)
```

[25]:

```
-1.822566282743153
```

[26]:

```
random>random1
```

[26]:

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

[27]:

```
print(random[random>1])
```

```
[1.19339736 2.50867991 1.29308431]
```

[28]:

```
import nbconvert
```

[29]:

```
import pandoc
```

[ ]:

```

```