

DEENA 20104016

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

zeros and ones

```
In [2]: import numpy as np
print(np.zeros(5))
print(np.ones(5))
```

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

Array

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

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

Random array

```
In [4]: a=np.empty(5)
print(a)
```

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

Range of values with even intervals

```
In [5]: print(np.linspace(0,100,num=50, dtype=np.int64))
```

```
[ 0  2  4  6  8 10 12 14 16 18 20 22 24 26 28 30 32 34
 36 38 40 42 44 46 48 51 53 55 57 59 61 63 65 67 69 71
 73 75 77 79 81 83 85 87 89 91 93 95 97 100]
```

Range of values with specified intervals

```
In [6]: print(np.linspace(0,10,num=5, dtype=np.int64))
```

```
[ 0  2  5  7 10]
```

Access and manipulate array

```
In [7]: a=np.array([1,2,3,4,5,6,7])
        print(a[2])
        a[5]=10
        print(a)
```

```
3
[ 1  2  3  4  5 10  7]
```

Create 2D array and Check shape of the array

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

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

Arange and Linspace()

```
In [9]: print(np.arange(0,11))
        print(np.linspace(0,10,num=10,dtype=np.int64))
```

```
[ 0  1  2  3  4  5  6  7  8  9 10]
[ 0  1  2  3  4  5  6  7  8 10]
```

Array of random values between 0 & 1

```
In [10]: print(np.random.random())
```

```
0.43981064203332754
```

Repeat each element of array using repeat and tile()

```
In [11]: a=np.array([1,2,3,4,5,6,7,8])
        print(np.repeat(a,3))
```

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

```
In [12]: print(np.tile(a,5))
```

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

Shape and Size of a array

```
In [13]: b=np.array([[1,2,3],[4,5,6],[7,8,9]])  
print(np.shape(b))
```

(3, 3)

```
In [14]: print(np.size(b))
```

9

Total no of elements in the array

```
In [15]: c=np.array([[1,2,3],[4,5,6],[7,8,9],[4,2,6]])  
print(np.size(c))
```

12

Dimensions of the array

```
In [16]: print(np.ndim(c))
```

2

Create an array and Reshape it

```
In [17]: a=np.arange(1,13).reshape(6,1,2)  
print(a)
```

```
[[[ 1  2]]  
 [[ 3  4]]  
 [[ 5  6]]  
 [[ 7  8]]  
 [[ 9 10]]  
 [[11 12]]]
```

Null array

```
In [18]: print(np.zeros(10,dtype=np.int64))
```

[0 0 0 0 0 0 0 0 0 0]

Print array ranging 10 to 49 and divide the array b y 7 and whose remainder is 0

```
In [19]: a=np.arange(10,50)
         print(a[(a%7==0)])
```

```
[14 21 28 35 42 49]
```

Create an array and check any to conditions

```
In [20]: a=np.array([1,2,3,4,5,6,7,8,9])
         print(a[(a>1)&(a<10)])
```

```
[2 3 4 5 6 7 8 9]
```

Using arithmetic operators in array

```
In [21]: a=np.array([1,2,3,4])
         b=np.array([5,6,7,8])
         print(a+b)
```

```
[ 6  8 10 12]
```

```
In [22]: print(b-a)
```

```
[4 4 4 4]
```

```
In [23]: print(a/b)
```

```
[0.2      0.33333333 0.42857143 0.5      ]
```

```
In [24]: print(a*b)
```

```
[ 5 12 21 32]
```

Using relational operator

```
In [25]: print(a<b)
```

```
[ True  True  True  True]
```

```
In [26]: print(a>b)
```

```
[False False False False]
```

```
In [27]: print(a==b)
```

```
[False False False False]
```

```
In [28]: print(a>=b)
```

```
[False False False False]
```

```
In [29]:
```

```
print(a<=b)
```

```
[ True  True  True  True]
```

Difference between python and Ipython

Python is a high level general purpose programming language. Python is easy to read, understand and learn. You can build many different types of applications using python, like Web Applications, Desktop Applications, Command Line Utilities, Machine Learning models etc.

IPython is an interactive shell that is built with python. It provides a more useful shell environment to execute python code in REPL (Read Eval Print Loop). It makes it more interactive by adding features like syntax highlighting, code completion etc. IPython also comes with other tools like Jupyter Notebooks etc.