# Numpy Essentials

August 21, 2022

### 0.0.1 Numpy Basics

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

- creat single & Multiple type array
- Single Dimension Numpy

[2]:
```
n1=np.array([1,2,45,7,5343])
n1
```

[2]:
```
array([   1,    2,   45,    7, 5343])
```

[3]:
```
type(n1)
```

[3]:
```
numpy.ndarray
```

[4]:
```
n2=np.array([[32,45,3,24,5],[34,5,6,734,6]])
n2
```

[4]:
```
array([[ 32,  45,   3,  24,   5],
       [ 34,   5,   6, 734,   6]])
```

[5]:
```
type(n2)
```

[5]:
```
numpy.ndarray
```

[6]:
```
n3=np.array([[23,45,2,45,3,4],[4,34,56,34,6,3]])
n3
```

[6]:
```
array([[23, 45,  2, 45,  3,  4],
       [ 4, 34, 56, 34,  6,  3]])
```

[7]:
```
n4=np.zeros((1,2))
n4
```

[7]:
```
array([[0., 0.]])
```

[8]:
```
n5=np.zeros((5,5))
n5
```

[8]:
```
array([[0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.]])
```

[9]:
```
n6=np.zeros((2,3))
n6
```

[9]:
```
array([[0., 0., 0.],
       [0., 0., 0.]])
```

[10]:
```
n7=np.zeros((6,6))
n7
```

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

- creat 2 matrix full with 10

[11]:
```
n8=np.full((2,2),(10))
n8
#(2,2) are dimensions while it filled with 10
```

[11]:
```
array([[10, 10],
       [10, 10]])
```

[12]:
```
n9=np.full((5,5),(5))
n9
```

[12]:
```
array([[5, 5, 5, 5, 5],
       [5, 5, 5, 5, 5],
       [5, 5, 5, 5, 5],
       [5, 5, 5, 5, 5],
       [5, 5, 5, 5, 5]])
```

[13]:
```
n10=np.full((3,6),(9))
n10
```

[13]:
```
array([[9, 9, 9, 9, 9, 9],
       [9, 9, 9, 9, 9, 9],
       [9, 9, 9, 9, 9, 9]])
```

### 0.0.2 Initializing Numpy array within a range

[14]:
```
n1=np.arange(10,20)
n1
```

[14]:
```
array([10, 11, 12, 13, 14, 15, 16, 17, 18, 19])
```

[15]:
```
n2=np.arange(20,50,5)
n2
```

[15]:
```
array([20, 25, 30, 35, 40, 45])
```

[16]:
```
n3=np.arange(23,100)
n3
```

[16]:
```
array([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])
```

[17]:
```
n4=np.arange(23,100,20)
n4
```

[17]:
```
array([23, 43, 63, 83])
```

### 0.0.3 Initializing Numpy array with random Numbers

[18]:
```
np1=np.random.randint(100,1000,10)
n1
```

[18]:
```
array([10, 11, 12, 13, 14, 15, 16, 17, 18, 19])
```

### 0.0.4 Checking shape of Numpy arrays

[19]:
```
n2=np.array([[2,3,4],[3,3,5]])
n2.shape
```

[19]:
```
(2, 3)
```

[20]:
```
n2.shape=(3,2)
n2
```

[20]:
```
array([[2, 3],
       [4, 3],
       [3, 5]])
```

[21]:
```
n4=np.array([[36,32,5],[2,31,5]])
n4.shape
```

[21]:
```
(2, 3)
```

[22]:
```
n4.shape = (3,2)
n4.shape
```

[22]:
```
(3, 2)
```

### 0.0.5 Joining Numpy Arrays

[23]:
```
n1=np.array([10,20,33])
n2=np.array([100,23,200])
np.vstack((n1,n2))
```

[23]:
```
array([[ 10,  20,  33],
       [100,  23, 200]])
```

[24]:
```
np.hstack((n2,n2))
```

[24]:
```
array([100,  23, 200, 100,  23, 200])
```

[25]:
```
np.column_stack((n1,n2))
```

[25]:
```
array([[ 10, 100],
       [ 20,  23],
       [ 33, 200]])
```

### 0.0.6 Numpy Intersection & Difference

[26]:
```
n1=np.array([10,20,30,40,50])
n2=([22,50,30])
```

[27]:
```
np.intersect1d(n1,n2)
#it will show the common elements
```

[27]:
```
array([30, 50])
```

[28]:
```
np.setdiff1d(n1,n2)
#it will show the diffrent elements which is present in n1
#and ignore the common
```

[28]:
```
array([10, 20, 40])
```

[29]:
```
np.setdiff1d(n2,n1)
#it show the element of n2 which is different
```

[29]:
```
array([22])
```

### 0.0.7 Mathematics Numpy array

[30]:
```
n1=np.array([30,20])
n2=np.array([10,20])
```

[31]:
```
np.sum([n1,n2])
#it will sum n1 and n2 all elements
```

[31]:
```
80
```

[32]:
```
np.sum([n2,n1])
```

[32]:
```
80
```

[33]:
```
np.sum([n1,n2],axis=0)
```

[33]:
```
array([40, 40])
```

[34]:
```
np.sum([n1,n2],axis=1)
```

[34]:
```
array([50, 30])
```

[35]:
```
### adding integer in array
n11=np.array([1,2,3])
n12=np.array([4,5,6])
```

[36]:
```
# Addition
```

[37]:
```
n11+1
```

[37]:
```
array([2, 3, 4])
```

[38]:
```
n12+20
```

[38]:
```
array([24, 25, 26])
```

[39]:
```
### ubtraction
```

[40]:
```
n11-3
```

[40]:
```
array([-2, -1,  0])
```

[41]:
```
n12-1
```

[41]:
```
array([3, 4, 5])
```

[42]:
```
# Division
```

[43]:
```
n11/2
```

[43]:
```
array([0.5, 1. , 1.5])
```

[44]:
```
n12/3
```

[44]:
```
array([1.33333333, 1.66666667, 2.        ])
```

[45]:
```
# Multiplication
```

[46]:
```
n11*4
```

[46]:
```
array([ 4,  8, 12])
```

[47]:
```
n12*3
```

[47]:
```
array([12, 15, 18])
```

### 0.0.8 Mean,Median,Standard Diviation

- Mean

[48]:
```
n22=np.array([2,3,4,5])
```

[49]:
```
np.mean(n22)
```

[49]:
```
3.5
```

- Median

[50]:
```
np.median(n22)
```

[50]:
```
3.5
```

- Standard Diviasion

[51]:
```
np.std(n22)
```

[51]:
```
1.118033988749895
```

### 0.0.9 Saving and Loading Numpy

[52]:
```
n32=np.array([23,3,4,56,7,4,6677,5])
```

[53]:
```
np.save('my_numpyarray',n32)
```

[54]:
```
n33=np.load('my_numpyarray.npy')
```

[55]:
```
n32
```

[55]:
```
array([  23,    3,    4,   56,    7,    4, 6677,    5])
```

[56]:
```
n1=([2,3,4])
np.save('mine_array',n1)
```

[57]:
```
n2=np.load('mine_array.npy')
```

[58]:
```
n1
```

[58]:
```
[2, 3, 4]
```