numpy practice session

In [1]:
```
# import numpy library in jupyter notebook
import numpy as np
```

Creating an array using numpy

In [2]:
```
# 1_ D array
import numpy as np
food=np.array(["Samosa","Pakora","Raita"])
food
```

Out[2]:
```
array(['Samosa', 'Pakora', 'Raita'], dtype='<U6')
```

In [3]:
```
price=np.array([5,5,5])
price
```

Out[3]:
```
array([5, 5, 5])
```

In [4]:
```
type(food)
```

Out[4]:
```
numpy.ndarray
```

In [5]:
```
type(price)
```

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

In [6]:
```
len(food)
```

Out[6]:
```
3
```

In [7]:
```
len(price)
```

Out[7]:
```
3
```

In [8]:
```
food[0:]
```

Out[8]:
```
array(['Samosa', 'Pakora', 'Raita'], dtype='<U6')
```

In [9]:
```
price.mean()
```

Out[9]:
```
5.0
```

In [10]:
```
# zeros
np.zeros(5)
```

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

In [11]:
```
#ones
np.ones(6)
```

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

In [12]:
```
#empty
np.empty(5)
```

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

In [13]:
```
# range
np.arange(10)
```

Out[13]:
```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

In [14]:
```
# specify
np.arange(2,20)
```

Out[14]:
```
array([ 2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
       19])
```

In [15]:
```
# specify
np.arange(2,20,2)
```

Out[15]:
```
array([ 2,  4,  6,  8, 10, 12, 14, 16, 18])
```

In [16]:
```
# specific interval
np.arange(5,55,5)
```

Out[16]:
```
array([ 5, 10, 15, 20, 25, 30, 35, 40, 45, 50])
```

In [17]:
```
# line space
np.linspace(1,100,num=10)
```

Out[17]:
```
array([  1.,  12.,  23.,  34.,  45.,  56.,  67.,  78.,  89., 100.])
```

In [18]:
```
# specify your data
np.ones(50,dtype=np.int64)
```

Out[18]:
```
array([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, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1], dtype=int64)
```

In [19]:
```
# specify your data
```

```
np.ones(50,dtype=np.float64)
```

Out[19]:
```
array([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., 1., 1.,
       1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.])
```

Array Functions

In [20]:
```
a=np.array([10,12,15,2,4,6,100,320,0.5,10.3])
a
```

Out[20]:
```
array([ 10. ,  12. ,  15. ,   2. ,   4. ,   6. , 100. , 320. ,   0.5,
        10.3])
```

In [21]:
```
a.sort()
a
```

Out[21]:
```
array([  0.5,   2. ,   4. ,   6. ,  10. ,  10.3,  12. ,  15. , 100. ,
       320. ])
```

In [22]:
```
b=np.array([1,3,5,7,9,6,10,0.8,5])
b
```

Out[22]:
```
array([ 1. ,  3. ,  5. ,  7. ,  9. ,  6. , 10. ,  0.8,  5. ])
```

In [23]:
```
import numpy as np
c=np.concatenate((a,b))
c
```

Out[23]:
```
array([  0.5,   2. ,   4. ,   6. ,  10. ,  10.3,  12. ,  15. , 100. ,
       320. ,   1. ,   3. ,   5. ,   7. ,   9. ,   6. ,  10. ,   0.8,
         5. ])
```

In [24]:
```
c.sort()
c
```

Out[24]:
```
array([  0.5,   0.8,   1. ,   2. ,   3. ,   4. ,   5. ,   5. ,   6. ,
         6. ,   7. ,   9. ,  10. ,  10. ,  10.3,  12. ,  15. , 100. ,
       320. ])
```

2_ D array

In [25]:
```
x=np.array([[1,2,3,4],[5,4,3,2]])
x
```

Out[25]:
```
array([[1, 2, 3, 4],
       [5, 4, 3, 2]])
```

In [26]:
```
y=np.array([[6,7,8,9],[7,8,9,6]])
y
```

Out[26]:
```
array([[6, 7, 8, 9],
       [7, 8, 9, 6]])
```

In [27]:
```
c=np.concatenate((x,y),axis=0)
c
```

Out[27]:
```
array([[1, 2, 3, 4],
       [5, 4, 3, 2],
       [6, 7, 8, 9],
       [7, 8, 9, 6]])
```

In [28]:
```
c=np.concatenate((x,y),axis=1)
c
```

Out[28]:
```
array([[1, 2, 3, 4, 6, 7, 8, 9],
       [5, 4, 3, 2, 7, 8, 9, 6]])
```

In [29]:
```
a=np.array([[[0,1,2,3],
             [4,5,6,7]],

            [[0,1,2,3],
            [4,5,6,7]],

            [[0,1,2,3,],
             [4,5,6,7]]])
a
```

Out[29]:
```
array([[[0, 1, 2, 3],
        [4, 5, 6, 7]],

       [[0, 1, 2, 3],
        [4, 5, 6, 7]],

       [[0, 1, 2, 3],
        [4, 5, 6, 7]]])
```

In [30]:
```
# to find number of dimensions
a.ndim
```

Out[30]:
```
3
```

In [31]:
```
b=np.array([[5,6,7],
            [8,9,10],
            [10,11,12]])
b
```

Out[31]:
```
array([[ 5,  6,  7],
       [ 8,  9, 10],
       [10, 11, 12]])
```

In [32]:
```
b.ndim
```

Out[32]:
```
2
```

In [33]:
```
# size (number of elements)
a.size
```

Out[33]:
```
24
```

In [34]:
```
a
```

Out[34]:
```
array([[[0, 1, 2, 3],
        [4, 5, 6, 7]],

       [[0, 1, 2, 3],
        [4, 5, 6, 7]],

       [[0, 1, 2, 3],
        [4, 5, 6, 7]]])
```

In [35]:
```
# shape
a.shape
```

Out[35]:
```
(3, 2, 4)
```

In [36]:
```
# range
a=np.arange(9) # 3*3
a
```

Out[36]:
```
array([0, 1, 2, 3, 4, 5, 6, 7, 8])
```

In [37]:
```
# Reshape
b=a.reshape(3,3)#3*3=9
b
```

Out[37]:
```
array([[0, 1, 2],
       [3, 4, 5],
       [6, 7, 8]])
```

In [38]:
```
# reshape
np.reshape(a,newshape=(1,9),order='c')
```

Out[38]:
```
array([[0, 1, 2, 3, 4, 5, 6, 7, 8]])
```

In [39]:
```
# convert 1D into 2 D
a=np.array([1,2,3,4,5,6,7,8,9])
a
```

Out[39]:
```
array([1, 2, 3, 4, 5, 6, 7, 8, 9])
```

In [40]:
```
a.shape
```

Out[40]:
```
(9,)
```

In [41]:
```
# row wise conversion 2D
b=a[np.newaxis,:]
b
```

Out[41]:
```
array([[1, 2, 3, 4, 5, 6, 7, 8, 9]])
```

In [42]:
```
b.shape
```

Out[42]:
```
(1, 9)
```

In [43]:
```
# column wise conversion 2D
c=a[:,np.newaxis]
c
```

Out[43]:
```
array([[1],
       [2],
       [3],
       [4],
       [5],
       [6],
       [7],
       [8],
       [9]])
```

In [44]:
```
a
```

Out[44]:
```
array([1, 2, 3, 4, 5, 6, 7, 8, 9])
```

In [45]:
```
a[2:9]
```

Out[45]:
```
array([3, 4, 5, 6, 7, 8, 9])
```

In [46]:
```
a*6
```

Out[46]:
```
array([ 6, 12, 18, 24, 30, 36, 42, 48, 54])
```

In [47]:
```
a+6
```

Out[47]:
```
array([ 7,  8,  9, 10, 11, 12, 13, 14, 15])
```

In [49]:
```
a.sum()
```

Out[49]:
```
45
```

In [50]:
```
a.mean()
```

Out[50]:
```
5.0
```

In [ ]: