# Numpy practice session

In [ ]:
```
# pip install numpy
```

In [1]:
```
# import numpy in j.notebook
import numpy as np
```

# Creating an array using nupmy library

In [2]:
```
# 1-D Array
food = np.array(["pakora", "samosa", "raita"])
food
```

Out[2]:
```
array(['pakora', 'samosa', 'raita'], dtype='<U6')
```

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

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

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

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

In [6]:
```
type(price)
```

Out[6]:
```
numpy.ndarray
```

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

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

In [8]:
```
len(price)
```

Out[8]:
```
3
```

In [9]:
```
food[1]
```

Out[9]:
```
'samosa'
```

In [10]:
```
price[2]
```

Out[10]:
```
5
```

In [11]:
```
price[0:]
```

Out[11]:
```
array([5, 5, 5])
```

In [12]:
```
price.mean()
```

Out[12]:
```
5.0
```

# Methods

In [13]:
```
# Zero
np.zeros(5)
```

Out[13]:
```
array([0., 0., 0., 0., 0.])
```

In [14]:
```
# Ones
np.ones(7)
```

Out[14]:
```
array([1., 1., 1., 1., 1., 1., 1.])
```

In [15]:
```
# Empty
np.empty(4)
```

Out[15]:
```
array([6.23042070e-307, 4.67296746e-307, 1.69121096e-306, 1.86919785e-306])
```

In [16]:
```
# Range
np.arange(8)
```

Out[16]:
```
array([0, 1, 2, 3, 4, 5, 6, 7])
```

In [17]:
```
# Specify range
np.arange(4,24)
```

Out[17]:
```
array([ 4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
       21, 22, 23])
```

In [19]:
```
# Specific range with interval
np.arange(3,30,3)
```

Out[19]:
```
array([ 3,  6,  9, 12, 15, 18, 21, 24, 27])
```

In [20]:
```
# Table of 5
np.arange(5,50,5)
```

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

In [21]:
```
np.arange(5,55,5)
```

Out[21]:
```
array([ 5, 10, 15, 20, 25, 30, 35, 40, 45, 50])
```

In [25]:
```
# Line space
np.linspace(0, 10, num=6)
```

Out[25]:
```
array([ 0.,  2.,  4.,  6.,  8., 10.])
```

In [26]:
```
np.linspace(1,100, num=10)
```

Out[26]:
```
array([  1.,  12.,  23.,  34.,  45.,  56.,  67.,  78.,  89., 100.])
```

In [27]:
```
np.linspace(1,100, num=30)
```

Out[27]:
```
array([  1.        ,   4.4137931 ,   7.82758621,  11.24137931,
        14.65517241,  18.06896552,  21.48275862,  24.89655172,
        28.31034483,  31.72413793,  35.13793103,  38.55172414,
        41.96551724,  45.37931034,  48.79310345,  52.20689655,
        55.62068966,  59.03448276,  62.44827586,  65.86206897,
        69.27586207,  72.68965517,  76.10344828,  79.51724138,
        82.93103448,  86.34482759,  89.75862069,  93.17241379,
        96.5862069 , 100.        ])
```

In [28]:
```
# Specify your data type
np.ones(50, dtype=np.int64)
```

Out[28]:
```
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], dtype=int64)
```

In [29]:
```
np.ones(50, dtype=np.int32)
```

Out[29]:
```
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])
```

In [30]:
```
np.ones(50, dtype=np.int64)
```

Out[30]:
```
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.])
```

In [34]:
```
np.ones(50, dtype=np.float64)
```

Out[34]:
```
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.])
```

In [35]:
```
np.ones(50, dtype=np.float32)
```

Out[35]:
```
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.],
      dtype=float32)
```

# Array functions

In [2]:
```
import numpy as np
a = np.array([12,34,67,89,46,28,7.8,5.9,3.0])
a
```

Out[2]:
```
array([12. , 34. , 67. , 89. , 46. , 28. ,  7.8,  5.9,  3. ])
```

In [3]:
```
a.sort()
a
```

Out[3]:
```
array([ 3. ,  5.9,  7.8, 12. , 28. , 34. , 46. , 67. , 89. ])
```

In [4]:
```
b = np.array([10.2,23.7,56.9,19.3])
b
```

Out[4]:
```
array([10.2, 23.7, 56.9, 19.3])
```

In [6]:
```
# Concatenate
c = np.concatenate((a,b))
c
```

Out[6]:
```
array([ 3. ,  5.9,  7.8, 12. , 28. , 34. , 46. , 67. , 89. , 10.2, 23.7,
       56.9, 19.3])
```

In [7]:
```
np.sort(c)
```

Out[7]:
```
array([ 3. ,  5.9,  7.8, 10.2, 12. , 19.3, 23.7, 28. , 34. , 46. , 56.9,
       67. , 89. ])
```

In [ ]:
```
# 2-D Arrays
```

In [8]:
```
m = np.array([[2,4,6,8],[9,6,8,4]])
m
```

Out[8]:
```
array([[2, 4, 6, 8],
       [9, 6, 8, 4]])
```

In [9]:
```
np.sort(m)
```

Out[9]:
```
array([[2, 4, 6, 8],
       [4, 6, 8, 9]])
```

In [10]:
```
n = np.array([[3,7,9,3],[2,7,4,0]])
n
```

Out[10]:
```
array([[3, 7, 9, 3],
       [2, 7, 4, 0]])
```

In [11]:
```
# Concatenate
o = np.concatenate((m,n))
o
```

Out[11]:
```
array([[2, 4, 6, 8],
       [9, 6, 8, 4],
       [3, 7, 9, 3],
       [2, 7, 4, 0]])
```

In [13]:
```
o = np.concatenate((m,n), axis=0)
o
```

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

In [15]:
```
o = np.concatenate((m,n), axis=1)
o
```

Out[15]:
```
array([[2, 4, 6, 8, 3, 7, 9, 3],
       [9, 6, 8, 4, 2, 7, 4, 0]])
```

In [16]:
```
len(o)
```

Out[16]:
```
2
```

In [17]:
```
type(o)
```

Out[17]:
```
numpy.ndarray
```

In [31]:
```
p = np.array([[[1,3,5,7],[9,7,5,3]]
      ,[[7,8,5,3],[7,9,3,1]],
      [[8,4,2,3],[0,7,5,8]],
     ])
p
```

Out[31]:
```
array([[[1, 3, 5, 7],
        [9, 7, 5, 3]],

       [[7, 8, 5, 3],
        [7, 9, 3, 1]],

       [[8, 4, 2, 3],
        [0, 7, 5, 8]]])
```

In [32]:
```
# To find the number of dimensions
p.ndim
```

Out[32]:
```
3
```

In [33]:
```
p = np.array([[2,3],[4,9],[8,5]])
p
```

Out[33]:
```
array([[2, 3],
       [4, 9],
       [8, 5]])
```

In [35]:
```
p.ndim
```

Out[35]:
```
2
```

In [37]:
```
q = np.array([[[1,3,5,7],[9,7,5,3]]
     ,[[7,8,5,3],[7,9,3,1]]])
q
```

Out[37]:
```
array([[[1, 3, 5, 7],
        [9, 7, 5, 3]],

       [[7, 8, 5, 3],
        [7, 9, 3, 1]]])
```

In [38]:
```
q.ndim
```

Out[38]:
```
3
```

In [39]:
```
# Size(number of elements)
q.size
```

Out[39]:
```
16
```

In [42]:
```
q
```

Out[42]:
```
array([[[1, 3, 5, 7],
        [9, 7, 5, 3]],

       [[7, 8, 5, 3],
        [7, 9, 3, 1]]])
```

In [46]:
```
q.ndim
```

Out[46]:
```
3
```

In [41]:
```
# Shape
q.shape
```

Out[41]:
```
(2, 2, 4)
```

In [43]:
```
p
```

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

In [45]:
```
p.ndim
```

Out[45]:
```
2
```

In [44]:
```
p.shape
```

Out[44]:
```
(3, 2)
```

In [54]:
```
# Range
p = np.arange(25) # 5*5
p
```

Out[54]:
```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
       17, 18, 19, 20, 21, 22, 23, 24])
```

In [55]:
```
# Reshape
b = p.reshape(5,5) # 5*5=25
b
```

Out[55]:
```
array([[ 0,  1,  2,  3,  4],
       [ 5,  6,  7,  8,  9],
       [10, 11, 12, 13, 14],
       [15, 16, 17, 18, 19],
       [20, 21, 22, 23, 24]])
```

In [58]:
```
b.ndim
```

Out[58]:
```
2
```

In [57]:
```
# Reshape
np.reshape(p, newshape=(1,25), order='c')
```

Out[57]:
```
array([[ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15,
        16, 17, 18, 19, 20, 21, 22, 23, 24]])
```

In [59]:
```
# Convert 1-D into 2-D Array
a = np.array([1,2,4,6,8,9,4,2,7])
a
```

Out[59]:
```
array([1, 2, 4, 6, 8, 9, 4, 2, 7])
```

In [60]:
```
a.shape
```

Out[60]:
```
(9,)
```

In [63]:
```
# Row wise 2-D conversion
b = a[np.newaxis, :]
b
```

Out[63]:
```
array([[1, 2, 4, 6, 8, 9, 4, 2, 7]])
```

In [62]:
```
b.shape
```

Out[62]:
```
(1, 9)
```

In [64]:
```
# Colum wise 2-D conversion
c= a[:,np.newaxis]
c
```

Out[64]:
```
array([[1],
       [2],
       [4],
       [6],
       [8],
       [9],
       [4],
       [2],
       [7]])
```

In [68]:
```
a = np.array([1,2,3,4,5,6,7,8])
a
```

Out[68]:
```
array([1, 2, 3, 4, 5, 6, 7, 8])
```

In [69]:
```
a[3:8]
```

Out[69]:
```
array([4, 5, 6, 7, 8])
```

In [70]:
```
a*4
```

Out[70]:
```
array([ 4,  8, 12, 16, 20, 24, 28, 32])
```

In [71]:
```
a+4
```

Out[71]:
```
array([ 5,  6,  7,  8,  9, 10, 11, 12])
```

In [73]:
```
a.sum()
```

Out[73]:
```
36
```

In [74]:
```
a.mean()
```

Out[74]:
```
4.5
```