Learning Numpy Library and practise

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

Basics

In [ ]:
```
One_DArray  =  np.array([12,3,4,3,5,3], dtype='float16')
One_DArray
```

Out[ ]:
```
array([12.,  3.,  4.,  3.,  5.,  3.], dtype=float16)
```

In [ ]:
```
Two_DArray = np.array([[4.0,5.0,7.0],[4.0,5.0,7.0], [4.0,5.0,3.0],[4.0,5.0,3.0]])
Two_DArray
```

Out[ ]:
```
array([[4., 5., 7.],
       [4., 5., 7.],
       [4., 5., 3.],
       [4., 5., 3.]])
```

In [ ]:
```
Three_DArray = np.array([[[4.0,5.0,7.0],[4.0,5.0,7.0]],[[4.0,5.0,7.0],[4.0,5.0,7.0]],[[
Three_DArray
```

Out[ ]:
```
array([[[4., 5., 7.],
        [4., 5., 7.]],

       [[4., 5., 7.],
        [4., 5., 7.]],

       [[4., 5., 7.],
        [4., 5., 7.]],

       [[4., 5., 7.],
        [4., 5., 7.]],

       [[4., 5., 7.],
        [4., 5., 7.]]])
```

In [ ]:
```
# what type is that array

print('Type of One_DArray : ', type(One_DArray))
print('Type of Two_DArray : ', type(Two_DArray))
print('Type of Three_DArray : ', type(Three_DArray))
```

```
Type of One_DArray :  <class 'numpy.ndarray'>
Type of Two_DArray :  <class 'numpy.ndarray'>
Type of Three_DArray :  <class 'numpy.ndarray'>
```

In [ ]:
```
# How to get the dimension of the array
print('One_DArray.ndim : ',One_DArray.ndim)
```

```
print('Two_DArray.ndim : ',Two_DArray.ndim)
print('Three_DArray.ndim : ',Three_DArray.ndim)
```

```
One_DArray.ndim :  1
Two_DArray.ndim :  2
Three_DArray.ndim :  3
```

In [ ]:
```
# How to get the shape of the array
print('One_DArray.shape : ',One_DArray.shape)
print('Two_DArray.shape : ',Two_DArray.shape)
print('Three_DArray.shape : ',Three_DArray.shape)
```

```
One_DArray.shape :  (6,)
Two_DArray.shape :  (4, 3)
Three_DArray.shape :  (5, 2, 3)
```

In [ ]:
```
#how to get the type of the array

print('One_DArray.dtype : ', One_DArray.dtype)
print('Two_DArray.dtype : ', Two_DArray.dtype)
print('Three_DArray.dtype : ', Three_DArray.dtype)
```

```
One_DArray.dtype :  float16
Two_DArray.dtype :  float64
Three_DArray.dtype :  float64
```

In [ ]:
```
#print the size of the array

print('One_DArray.size : ',One_DArray.size)
print('Two_DArray.size : ',Two_DArray.size)
print('Three_DArray.size : ',Three_DArray.size)
```

```
One_DArray.size :  6
Two_DArray.size :  12
Three_DArray.size :  30
```

In [ ]:
```
#print the size of the iteam array note the number is repersenting the size in bytes

print('One_DArray.itemsize : ',One_DArray.itemsize)
print('Two_DArray.itemsize : ',Two_DArray.itemsize)
print('Three_DArray.itemsize : ',Three_DArray.itemsize)
```

```
One_DArray.itemsize :  2
Two_DArray.itemsize :  8
Three_DArray.itemsize :  8
```

Accessing/Changing elements, rows, columns etc

In [ ]:
```
arr1 = np.array([[1,2,3,4,5,6,7],[8,9,10,11,12,13,14]])
arr1
```

Out[ ]:
```
array([[ 1,  2,  3,  4,  5,  6,  7],
       [ 8,  9, 10, 11, 12, 13, 14]])
```

In [ ]:
```
#get a specific element in this formate [row, column]
arr1[1, -2]
```

Out[ ]:
```
13
```

In [ ]:
```
#get a specific row

arr1[1, :]
```

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

In [ ]:
```
#get a specific column
arr1[:, 3]
```

Out[ ]:
```
array([ 4, 11])
```

In [ ]:
```
#gettig a little more fancy [startIndex:endIndex:stepsize]

print(arr1[0, 1:6:2])
print(arr1[0, ::2])
print(arr1[0, 1::2])
```

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

In [ ]:
```
arr1[1,-1] = 50
arr1
```

Out[ ]:
```
array([[ 1,  2,  3,  4,  5,  6,  7],
       [ 8,  9, 10, 11, 12, 13, 50]])
```

In [ ]:
```
arr1[:,2] = 0
arr1
```

Out[ ]:
```
array([[ 1,  2,  0,  4,  5,  6,  7],
       [ 8,  9,  0, 11, 12, 13, 50]])
```

In [ ]:
```
arr1[1,:] = 0
arr1
```

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

In [ ]:
```
arr2 = np.array([
  [[1,2],[3,4]],
  [[5,6],[7,8]],
  [[9,10],[11,12]],
  [[13,14],[15,16]],
])
arr2
```

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

       [[ 5,  6],
        [ 7,  8]],

       [[ 9, 10],
        [11, 12]],

       [[13, 14],
        [15, 16]]])
```

In [ ]:
```
arr2[2,1,1]
```

Out[ ]:
```
12
```

In [ ]:
```
arr2[:,:,0]
```

Out[ ]:
```
array([[ 1,  3],
       [ 5,  7],
       [ 9, 11],
       [13, 15]])
```

Initializing all type of array

In [ ]:
```
# All 0's arrays

arr3 = np.zeros((3,))
arr3
```

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

In [ ]:
```
np.ones(3, dtype='int32')
```

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

In [ ]:
```
np.full((3,4), 45, dtype='int32')
```

Out[ ]:
```
array([[45, 45, 45, 45],
       [45, 45, 45, 45],
       [45, 45, 45, 45]])
```

In [ ]:
```
#random values array

np.random.rand(4,2)
```

Out[ ]:
```
array([[0.7281525 , 0.13773112],
       [0.28762793, 0.74616723],
       [0.75231015, 0.54399883],
       [0.58872602, 0.698354  ]])
```

In [ ]:
```
#np identity matrix
```

```
np.identity(4)
```

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

In [ ]:
```
arr1 = np.ones((5,5))
arr1
```

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

In [ ]:
```
arr2 = np.zeros((3,3))
arr2
```

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

In [ ]:
```
arr2[1,1] = 9
arr2
```

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

In [ ]:
```
arr1[1:-1, 1:-1] = arr2
arr1
```

Out[ ]:
```
array([[1., 1., 1., 1., 1.],
       [1., 0., 0., 0., 1.],
       [1., 0., 9., 0., 1.],
       [1., 0., 0., 0., 1.],
       [1., 1., 1., 1., 1.]])
```

In [ ]:
```
#how to copy an array

a =  np.array([1,2,3,4])
b  = a.copy()

b[1] = 333

print('a ==> ', a)
print('b ==> ', b)
```

```
a ==>  [1 2 3 4]
b ==>  [  1 333   3   4]
```

Mathematics

In [ ]:

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

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

In [ ]:
```
a + 2
```

Out[ ]:
```
array([3, 4, 5, 6])
```

In [ ]:
```
a - 3
```

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

In [ ]:
```
a * 2
```

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

In [ ]:
```
a / 100
```

Out[ ]:
```
array([0.01, 0.02, 0.03, 0.04])
```

In [ ]:
```
b = [1,0,1,0]

a+b
```

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

In [ ]:
```
print(np.sin(a))
print(np.cos(a))
```

```
[ 0.84147098  0.90929743  0.14112001 -0.7568025 ]
[ 0.54030231 -0.41614684 -0.9899925  -0.65364362]
```

challange question

In [ ]:
```
a=''
for i in range(30):
  a = a + (str(i)) + ','

a
```

Out[ ]:
```
'0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,'
```

In [ ]:
```
a = np.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]
])
```

In [ ]:
```
a
```

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

In [ ]:
```
a[2:4,:2]
```

Out[ ]:
```
array([[11, 12],
       [16, 17]])
```

In [ ]:
```
a[[0,1,2,3], [1,2,3,4]]
```

Out[ ]:
```
array([ 2,  8, 14, 20])
```

In [ ]:
```
a[[0,4,5], 3:]
```

Out[ ]:
```
array([[ 4,  5],
       [24, 25],
       [29, 30]])
```

In [ ]:
```

```