Welcome to numpy tutorials

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

In [ ]:
```
myarr = np.array([[3,6,23,7]],np.int64)
myarr[0,1] = 45
myarr
```

Out[ ]:
```
array([[ 3, 45, 23,  7]])
```

In [ ]:
```
# conversion from other
listArray =np.array([[1,2,3],[4,5,6],[7,8,9]])
listArray.dtype
# np.array({23,4,5})
```

Out[ ]:
```
dtype('int64')
```

In [ ]:
```
# function to create arrays
# zero function to create arrays with 0
zeroes = np.zeros((3,3))
zeroes.shape
```

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

In [ ]:
```
# range fuction -to create an array from 0 to that number
rng = np.arange(2)
rng
```

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

In [ ]:
```
# linspace function linspace(x,y,z) create z items between x and y and store it in array. all in arithmetic progres
np.linspace(0,3,4)
```

Out[ ]:
```
array([0., 1., 2., 3.])
```

In [ ]:
```
np.empty((4,6)) #create arrays with random items
```

```
# empty liek create a empty erray of size mentioned
```

Out[ ]:
```
array([[4.67974645e-310, 0.00000000e+000, 9.82157975e+252,
        8.89489936e+252, 6.01346954e-154, 6.01347002e-154],
       [6.01347002e-154, 6.01347002e-154, 9.08366793e+223,
        1.14177168e+243, 2.45126797e+198, 1.06083187e-153],
       [2.35625393e+251, 6.01334511e-154, 6.01347002e-154,
        6.01347002e-154, 6.01347002e-154, 1.88556770e+122],
       [4.96820036e+180, 6.80600993e+212, 1.10317376e+217,
        1.19490107e+190, 2.06642651e+161, 5.44760669e-109]])
```

In [ ]:
```
# idenity
np.identity(3)
```

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

In [ ]:
```
# reshape function
arr = np.arange(6)
newarr=arr.reshape((3,2))
newarr.ravel()
```

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

In [ ]:
```
array = np.array([[1,2,3],[4,5,6],[7,8,9]])
# new = arr.reshape((3,3))
# T for transpose
# sum(axis=x) for having sum of axis
array.T
```

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

In [ ]:
```
for item in array.flat:
    print(item)
# cheking bytes consumed by nbytes
# ndim for cheking the dimension of array
array.nbytes
```

```
1
2
3
4
5
6
7
8
9
```

Out[ ]: 72

In [ ]:
```
one = np.array([1,23,4,5,6])
one.argmax() #gives us the index in whic value is maxm
one.argsort()
```

Out[ ]: array([0, 2, 3, 4, 1])

In [ ]:
```
cube = np.array([[1,2,3],[4,5,6],[7,8,9]])
cube.argmax()
cube.argsort()
```

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