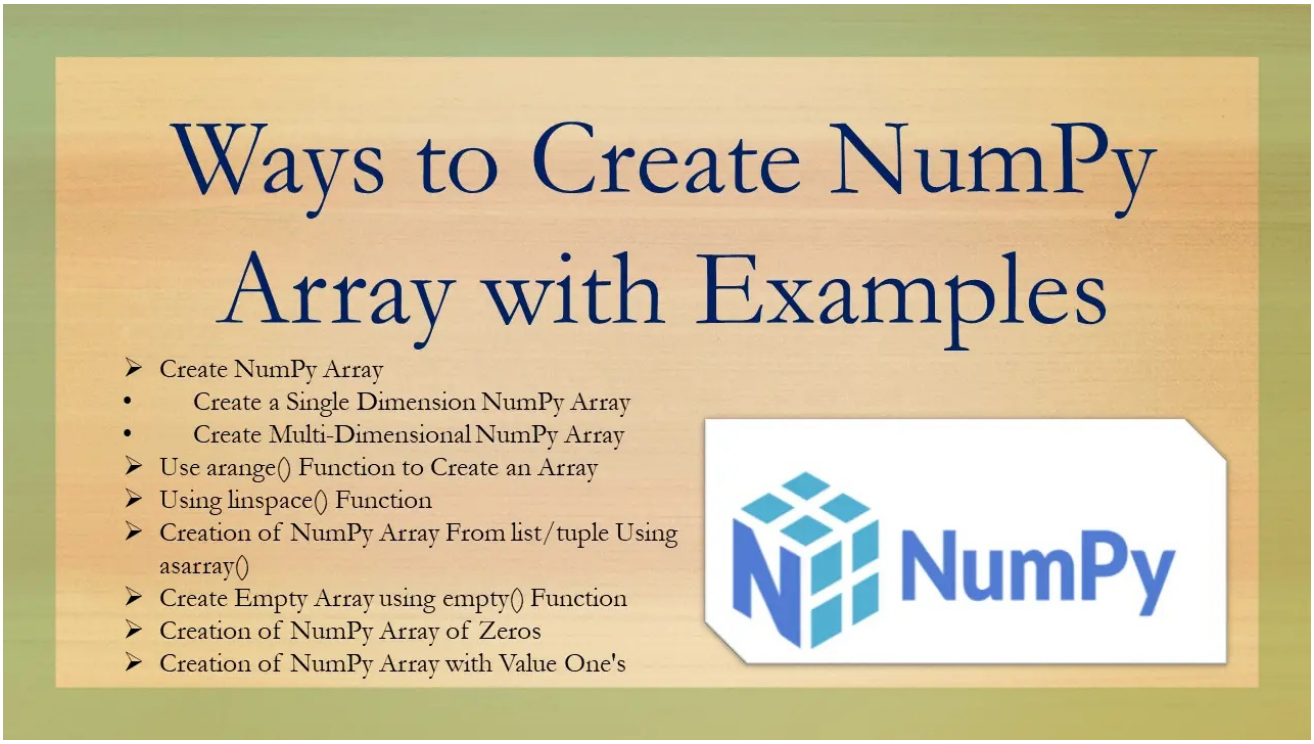

In [1]:
```
import numpy as np
a=np.arange(1,10)    #start from 1, end in 10 means 10-1
```

In [2]:
```
print(a)
```

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

Numpy array contain similar data types, if we create string numpy array then only string, its not like a list but it is faster than list.

In [7]:
```
type(a)
```

Out[7]:
```
numpy.ndarray
```

In [9]:
```
np.ndim(a)
```

Out[9]:
```
1
```

In [10]:
```
#means it is one dimensional array.
```

In [11]:
```
#Creating one domesion array using array
c=np.array([1,2,3,4,5,6,7,8,9])
```

In [12]:
```
print(c)
```

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

In [13]:
```
#if we keep 5 in arange the start and step will be default? yes
d=np.arange(5)
print(d)
```

```
[0 1 2 3 4]
```

In [2]:
```
#Now creating using linspace (it create array with equal interval)
import numpy as np
e=np.linspace(1,10)
print(e)
```

```
[ 1.         1.18367347  1.36734694  1.55102041  1.73469388  1.91836735
  2.10204082  2.28571429  2.46938776  2.65306122  2.83673469  3.02040816
  3.20408163  3.3877551   3.57142857  3.75510204  3.93877551  4.12244898
  4.30612245  4.48979592  4.67346939  4.85714286  5.04081633  5.2244898
  5.40816327  5.59183673  5.7755102   5.95918367  6.14285714  6.32653061
  6.51020408  6.69387755  6.87755102  7.06122449  7.24489796  7.42857143
  7.6122449   7.79591837  7.97959184  8.16326531  8.34693878  8.53061224
  8.71428571  8.89795918  9.08163265  9.26530612  9.44897959  9.63265306
  9.81632653 10.        ]
```

In [3]:
```
#default is 50, it will give 50 values
e=np.linspace(1,10,7)
```

In [4]:
```
print(e)
```

```
[ 1.   2.5  4.   5.5  7.   8.5 10. ]
```

In [5]:
```
#in this way we can get numbers of values we want using linspace
```

In [6]:
```
#now lets use asarray to create array from list & tuples
a=[1,2,3,4]
b=(2,3,4,5)
#a is list and b is tuple
```

In [7]:
```
array1=np.asarray(a)
array2=np.asarray(b)
```

In [8]:
```
print(array1)
```

```
[1 2 3 4]
```

In [10]:
```
#creating one dimension, zero dimension and two dimension array
a=np.array(42)
b=np.array([1,2,3,4,5])
c=np.array([[1,2,3],[4,5,6],[7,8,9]])
```

In [11]:
```
print(a)
```

```
42
```

In [12]:
```
a.dtype
```

Out[12]:
```
dtype('int32')
```

In [13]:
```
print(np.ndim(a))
```

```
0
```

In [14]:
```
print(np.ndim(b))
print(np.ndim(c))
```

```
1
2
```

In [15]:
```
#Creating arrays with one,zero, empty
a=np.ones((1,2))
```

In [16]:
```
print(a)
```

```
[[1. 1.]]
```

In [18]:
```
a=np.ones((1,3))
print(a)
```

```
[[1. 1. 1.]]
```

In [20]:
```
a=np.ones((2,3))
print(a)
```

```
[[1. 1. 1.]
 [1. 1. 1.]]
```

In [21]:
```
#so did we understand? front number in one array is row and second number is column.
```

In [24]:
```
#similar for zero and empty
b=np.zeros((1,3),int)
print(b)
```

```
[[0 0 0]]
```

In [25]:
```
#see we can remove decimal or from float to int, by indicating int datatype.
```