Numpy

```
In [2]: import sys
        sys.version
```

```
Out[2]: '3.13.5 | packaged by Anaconda, Inc. | (main, Jun 12 2025, 16:37:03) [MSC v.192
        9 64 bit (AMD64)]'
```

```
In [3]: import numpy as np #np,ab,etc we can give anyything
```

```
In [4]: np.__version__
```

```
Out[4]: '2.1.3'
```

Create list

```
In [6]: my_list=[0,1,2,3,4,5]
        my_list
```

```
Out[6]: [0, 1, 2, 3, 4, 5]
```

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

```
Out[7]: list
```

```
In [9]: arr=np.array(my_list)
        arr
```

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

```
In [10]: type(arr)
```

```
Out[10]: numpy.ndarray
```

```
In [12]: print(type(arr))
         print(type(my_list))
```

```
<class 'numpy.ndarray'>
<class 'list'>
```

```
In [15]: np.arange(10) #arange the number from start to end with step number
```

```
Out[15]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
In [16]: np.arange(10,20)
```

```
Out[16]: array([10, 11, 12, 13, 14, 15, 16, 17, 18, 19])
```

```
In [18]: np.arange(10,50,5)
```

```
Out[18]: array([10, 15, 20, 25, 30, 35, 40, 45])
```

```
In [19]: np.arange(10,30,3)

Out[19]: array([10, 13, 16, 19, 22, 25, 28])

In [20]: np.arange(10,50,5,4)

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[20], line 1
----> 1 np.arange(10,50,5,4)

TypeError: Cannot interpret '4' as a data type

In [22]: np.arange(20,10)

Out[22]: array([], dtype=int64)

In [23]: np.arange(8,20)

Out[23]: array([ 8,  9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19])

In [29]: np.arange(20,8) #first arg should be less than second arg

Out[29]: array([], dtype=int64)

In [26]: np.arange(-20,8)

Out[26]: array([-20, -19, -18, -17, -16, -15, -14, -13, -12, -11, -10,  -9,  -8,
                 -7,  -6,  -5,  -4,  -3,  -2,  -1,   0,   1,   2,   3,   4,   5,
                  6,   7])

In [30]: n=np.arange(-20,8)
         n

Out[30]: array([-20, -19, -18, -17, -16, -15, -14, -13, -12, -11, -10,  -9,  -8,
                 -7,  -6,  -5,  -4,  -3,  -2,  -1,   0,   1,   2,   3,   4,   5,
                  6,   7])

In [31]: np.zeros(3)

Out[31]: array([0., 0., 0.])

In [33]: np.zeros(3,dtype=int) #this is called hyper-parameter tuning(user-defined)

Out[33]: array([0, 0, 0])

In [34]: z=np.zeros(5)

In [35]: z

Out[35]: array([0., 0., 0., 0., 0.])

In [43]: np.zeros((5,3)) # here 5 is row and 3 is column
```

```
Out[43]:  array([[0., 0., 0.],
                 [0., 0., 0.],
                 [0., 0., 0.],
                 [0., 0., 0.],
                 [0., 0., 0.]])
```

```
In [40]:  np.zeros((2,2)) #2d array
```

```
Out[40]:  array([[0., 0.],
                 [0., 0.]])
```

```
In [42]:  np.zeros((3,3),dtype=int) #3d array
```

```
Out[42]:  array([[0, 0, 0],
                 [0, 0, 0],
                 [0, 0, 0]])
```

```
In [45]:  nd=np.zeros((5,9),dtype=int) #nd array or multi-d array
          nd
```

```
Out[45]:  array([[0, 0, 0, 0, 0, 0, 0, 0, 0],
                 [0, 0, 0, 0, 0, 0, 0, 0, 0],
                 [0, 0, 0, 0, 0, 0, 0, 0, 0],
                 [0, 0, 0, 0, 0, 0, 0, 0, 0],
                 [0, 0, 0, 0, 0, 0, 0, 0, 0]])
```

```
In [46]:  np.ones(3)
```

```
Out[46]:  array([1., 1., 1.])
```

```
In [48]:  np.ones(3,dtype=int)
```

```
Out[48]:  array([1, 1, 1])
```

```
In [49]:  np.ones((10,10),dtype=int)
```

```
Out[49]:  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],
                 [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]])
```

```
In [50]:  nd1=np.ones((10,10),dtype=int)
          nd1
```

```
Out[50]:  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],
                 [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]])
```

```
  In [ ]:
```