Numpy Arrays

In [1]:

```
import numpy as np
```

In [2]:

```
mylist=[1,2,3,4]
```

In [3]:

```
type(mylist)
```

Out[3]:

```
list
```

In [4]:

```
np.array(mylist)
```

Out[4]:

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

In [5]:

```
myarr=np.array(mylist)
```

In [7]:

```
type(myarr)
```

Out[7]:

```
numpy.ndarray
```

In [8]:

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

In [9]:

```
mymatrix
```

Out[9]:

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

In [13]:

```

np.array(mymatrix)
```

Out[13]:

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

Numpy Built In Methods

1.arange

In [14]:

```
np.arange(0,10)
```

Out[14]:

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

In [17]:

```
np.arange(0,101,10)
```

Out[17]:

```
array([  0,  10,  20,  30,  40,  50,  60,  70,  80,  90, 100])
```

2.zeros

In [18]:

```
np.zeros(5)
```

Out[18]:

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

In [19]:

```
np.zeros((5,5))
```

Out[19]:

```
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.]])
```

In [20]:

```
np.zeros((5,2))
```

Out[20]:

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

3.ones

In [21]:

```
np.ones((4,4))
```

Out[21]:

```
array([[1., 1., 1., 1.],
       [1., 1., 1., 1.],
       [1., 1., 1., 1.],
       [1., 1., 1., 1.]])
```

In [22]:

```
np.ones(5)
```

Out[22]:

```
array([1., 1., 1., 1., 1.])
```

4.linspace

In [23]:

```
np.linspace(0,10,20)
```

Out[23]:

```
array([ 0.        ,  0.52631579,  1.05263158,  1.57894737,  2.10526316,
        2.63157895,  3.15789474,  3.68421053,  4.21052632,  4.73684211,
        5.26315789,  5.78947368,  6.31578947,  6.84210526,  7.36842105,
        7.89473684,  8.42105263,  8.94736842,  9.47368421, 10.        ])
```

In [24]:

```
len(np.linspace(0,10,20))
```

Out[24]:

```
20
```

In [26]:

```
np.linspace(0,10)
```

Out[26]:

```
array([ 0.        ,  0.20408163,  0.40816327,  0.6122449 ,  0.81632653,
        1.02040816,  1.2244898 ,  1.42857143,  1.63265306,  1.83673469,
        2.04081633,  2.24489796,  2.44897959,  2.65306122,  2.85714286,
        3.06122449,  3.26530612,  3.46938776,  3.67346939,  3.87755102,
        4.08163265,  4.28571429,  4.48979592,  4.69387755,  4.89795918,
        5.10204082,  5.30612245,  5.51020408,  5.71428571,  5.91836735,
        6.12244898,  6.32653061,  6.53061224,  6.73469388,  6.93877551,
        7.14285714,  7.34693878,  7.55102041,  7.75510204,  7.95918367,
        8.16326531,  8.36734694,  8.57142857,  8.7755102 ,  8.97959184,
        9.18367347,  9.3877551 ,  9.59183673,  9.79591837, 10.        ])
```

5.identity

In [25]:

```
np.eye(4)
```

Out[25]:

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

6.random

In [33]:

```
np.random.rand(3)
```

Out[33]:

```
array([0.54735238, 0.62582196, 0.45673339])
```

In [36]:

```
np.random.rand(5,3)
```

Out[36]:

```
array([[0.4503795 , 0.95963438, 0.42125466],
       [0.65315422, 0.97890836, 0.07104798],
       [0.18931638, 0.47456803, 0.18980078],
       [0.76958856, 0.3566911 , 0.66121656],
       [0.72335134, 0.22619518, 0.91516511]])
```

In [38]:

```
np.random.randn(6)
```

Out[38]:

```
array([-0.51309192,  1.13153897,  1.4030723 ,  0.05547472, -0.70724735,
       -2.32040261])
```

In [41]:

```
np.random.randn(2,3)
```

Out[41]:

```
array([[-0.86913677, -0.9051443 ,  0.97550837],
       [-1.31818155,  0.24443864,  0.60047743]])
```

In [59]:

```
np.random.randint(0,101,10)
```

Out[59]:

```
array([91, 84, 31, 27, 94, 97, 41, 56, 87, 39])
```

In [56]:

```
np.random.randint(0,101,(5,4))
```

Out[56]:

```
array([[65, 93, 12, 98],
       [71, 72, 87, 71],
       [62, 27, 70, 48],
       [ 7, 62, 25, 64],
       [94, 41,  7, 84]])
```

7.seed

In [60]:

```
np.random.seed(42)
np.random.rand(4)
```

Out[60]:

```
array([0.37454012, 0.95071431, 0.73199394, 0.59865848])
```

In [63]:

```
np.random.seed(42)
np.random.rand(5)
```

Out[63]:

```
array([0.37454012, 0.95071431, 0.73199394, 0.59865848, 0.15601864])
```

In [65]:

```
np.random.seed(101)
np.random.rand(3)
```

Out[65]:

```
array([0.51639863, 0.57066759, 0.02847423])
```

In [66]:

```
np.random.rand(3)
```

Out[66]:

```
array([0.17152166, 0.68527698, 0.83389686])
```

In [67]:

```
np.random.rand(3)
```

Out[67]:

```
array([0.30696622, 0.89361308, 0.72154386])
```

8.reshape

In [71]:

```
arr=np.arange(0,25)
```

In [72]:

```
arr
```

Out[72]:

```
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 [75]:

```
arr.reshape(5,5)
```

Out[75]:

```
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 [76]:

```
arr.reshape(5,4)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [76], in <cell line: 1>()
----> 1 arr.reshape(5,4)

ValueError: cannot reshape array of size 25 into shape (5,4)
```

9.min & max

In [79]:

```
ranarr=np.random.randint(0,101,10)
```

In [84]:

```
ranarr
```

Out[84]:

```
array([64,  5, 12, 93, 40, 49, 83,  8, 29, 59])
```

In [82]:

```
ranarr.max()
```

Out[82]:

```
93
```

In [85]:

```
ranarr.min()
```

Out[85]:

```
5
```

In [87]:

```
ranarr.argmax()
```

Out[87]:

```
3
```

In [88]:

```
ranarr.argmin()
```

Out[88]:

```
1
```

In [89]:

```
ranarr.dtype
```

Out[89]:

```
dtype('int32')
```

In [90]:

```
arr.shape
```

Out[90]:

```
(25,)
```

In [91]:

```
arr=arr.reshape(5,5)
```

In [92]:

```
arr
```

Out[92]:

```
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 [93]:

```
arr.shape
```

Out[93]:

```
(5, 5)
```

10.logspace

In [95]:

```
np.logspace(1,3,5,base=12)
```

Out[95]:

```
array([  12.        ,   41.56921938,  144.        ,  498.83063258,
       1728.        ])
```

11.flatten

In [112]:

```
a=([123],[4,5,6])
```

In [111]:

```
np.array(a)
```

```
C:\Users\Shubham\AppData\Local\Temp\ipykernel_10788\1669458627.py:1: Visible
DeprecationWarning: Creating an ndarray from ragged nested sequences (which
is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths or
shapes) is deprecated. If you meant to do this, you must specify 'dtype=obje
ct' when creating the ndarray.
  np.array(a)
```

Out[111]:

```
array([list([123]), list([4, 5, 6])], dtype=object)
```

In [ ]:

```

```