

Module 02

Numpy Arrays

Data Science Developer

Using Numpy

```
In [1]: import numpy as np
```

Creating Numpy Arrays

From a Python List

```
In [19]: my_list = [1,2,3]
         my_list
```

```
Out[19]: [1, 2, 3]
```

```
In [16]: np.array(my_list)
```

```
Out[16]: array([1, 2, 3])
```

```
In [20]: my_matrix = [[1,2,3],[4,5,6],[7,8,9]]
         my_matrix
```

```
Out[20]: [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

```
In [21]: np.array(my_matrix)
```

```
Out[21]: array([[1, 2, 3],
                [4, 5, 6],
                [7, 8, 9]])
```

Built-in Methods

arange

```
In [22]: np.arange(0,10)
```

```
Out[22]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
In [23]: np.arange(0,11,2)
```

```
Out[23]: array([ 0,  2,  4,  6,  8, 10])
```

Built-in Methods

zeros and ones

```
In [24]: np.zeros(3)
```

```
Out[24]: array([ 0.,  0.,  0.])
```

```
In [26]: np.zeros((5,5))
```

```
Out[26]: 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 [27]: np.ones(3)
```

```
Out[27]: array([ 1.,  1.,  1.])
```

```
In [28]: np.ones((3,3))
```

```
Out[28]: array([[ 1.,  1.,  1.],
                 [ 1.,  1.,  1.],
                 [ 1.,  1.,  1.]])
```

Built-in Methods

linspace

```
In [29]: np.linspace(0,10,3)
```

```
Out[29]: array([ 0.,  5., 10.])
```

```
In [31]: np.linspace(0,10,50)
```

```
Out[31]: 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.          ])
```

Built-in Methods

eye

```
In [11]: np.eye(4)
```

```
Out[11]: array([[1., 0., 0., 0.],  
                [0., 1., 0., 0.],  
                [0., 0., 1., 0.],  
                [0., 0., 0., 1.]])
```

Built-in Methods

random.rand

```
In [47]: np.random.rand(2)
```

```
Out[47]: array([ 0.11570539,  0.35279769])
```

```
In [19]: np.random.rand(5,5)
```

```
Out[19]: array([[0.46733762, 0.61821618, 0.55572844, 0.66595536, 0.82351667],
                [0.6767309 , 0.21507597, 0.80908423, 0.31201054, 0.89075435],
                [0.79426775, 0.19536746, 0.29059715, 0.24310793, 0.1538956 ],
                [0.27480898, 0.0868552 , 0.46727468, 0.25064969, 0.35202803],
                [0.09306495, 0.39282406, 0.96541148, 0.08188501, 0.54730353]])
```


Built-in Methods

random.randn

```
In [17]: np.random.randn(2)
```

```
Out[17]: array([1.15178134, 0.81114334])
```

```
In [21]: np.random.randn(5,5)
```

```
Out[21]: array([[ -0.61466551,  0.56820696, -0.90523534,  1.63720345,  0.23213716],
                 [ -0.55540677,  1.49907936, -0.3743271 , -0.73975908, -0.49645825],
                 [  0.05348141, -0.78840726, -1.33904729,  0.22485744, -1.1621428 ],
                 [-1.57131486, -0.96163608, -0.60896763, -0.05697113, -0.24200832],
                 [  0.11211018, -1.78386241,  0.03627135, -1.80388622, -1.20737987]])
```

Built-in Methods

random.randint

```
In [50]: np.random.randint(1,100)
```

```
Out[50]: 44
```

```
In [4]: np.random.randint(1,100, 10)
```

```
Out[4]: array([ 6, 93, 20, 34, 84, 14, 21, 25, 69, 59])
```

Array Attributes and Methods

```
In [7]: arr = np.arange(25)  
        ranarr = np.random.randint(0,50,10)
```

```
In [8]: arr
```

```
Out[8]: 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 [57]: ranarr
```

```
Out[57]: array([10, 12, 41, 17, 49,  2, 46,  3, 19, 39])
```

Array Attributes and Methods

reshape

```
In [49]: arr.reshape(5,5)
```

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

Array Attributes and Methods

max, min, argmax, argmin

```
In [64]: ranarr
```

```
Out[64]: array([10, 12, 41, 17, 49,  2, 46,  3, 19, 39])
```

```
In [61]: ranarr.max()
```

```
Out[61]: 49
```

```
In [62]: ranarr.argmax()
```

```
Out[62]: 4
```

```
In [63]: ranarr.min()
```

```
Out[63]: 2
```

```
In [60]: ranarr.argmin()
```

```
Out[60]: 5
```

Array Attributes and Methods

shape

```
In [9]: # Vector  
arr.shape
```

```
Out[9]: (25,)
```

```
In [66]: # Notice the two sets of brackets  
arr.reshape(1,25)
```

```
Out[66]: 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 [69]: arr.reshape(1,25).shape
```

```
Out[69]: (1, 25)
```

Array Attributes and Methods

shape

```
In [70]: arr.reshape(25,1)
```

```
Out[70]: 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(25,1).shape
```

```
Out[76]: (25, 1)
```

Array Attributes and Methods

dtype

```
In [52]: arr.dtype
```

```
Out[52]: dtype('int32')
```

```
In [60]: arr1 = np.array(['1', 'test', '5.0'])
```

```
In [62]: arr1.dtype
```

```
Out[62]: dtype('<U4')
```