Module 02

Numpy Arrays

Data Science Developer

Purwadhika
Startup and Coding School

Outline

- What is array ?
- Numpy ?
- Attributes and methods for numpy array

Purwadhika
Startup and Coding School

What is Array ?

What is Array ?

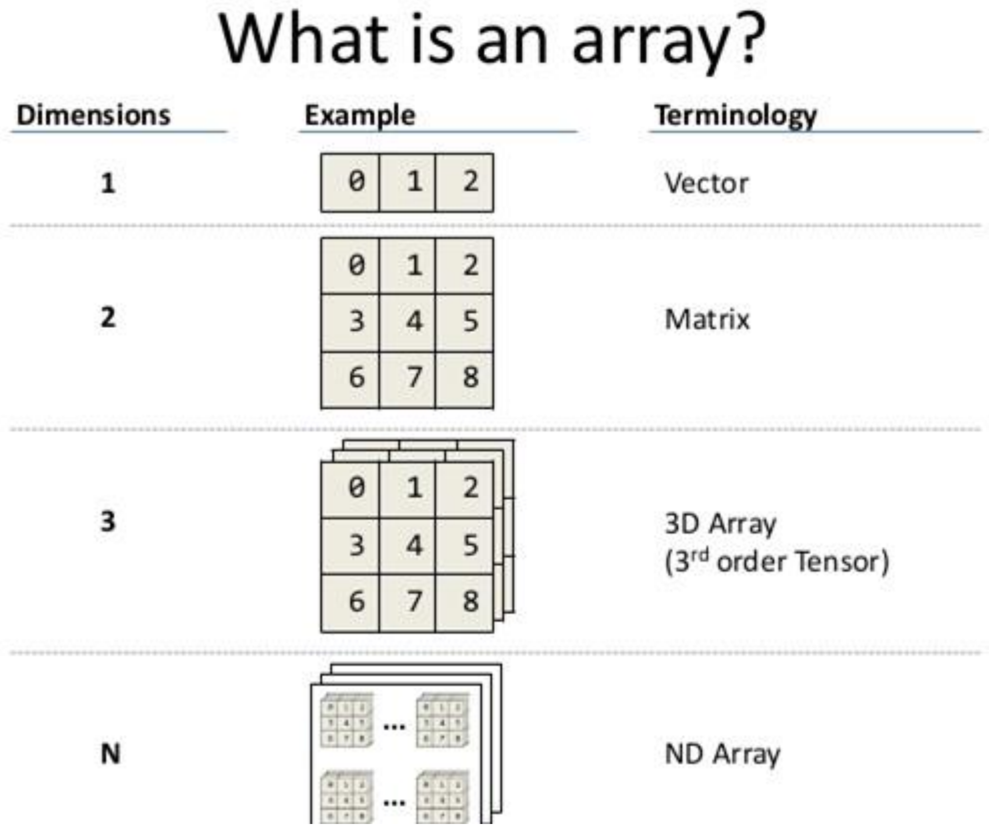
Array is a structured data type that store multiple value with the same type.

Array is mutable

Array has index and started from 0

Array has many form : 1D, 2D, 3D, …, nD

Terminologies in Array



What is an array?

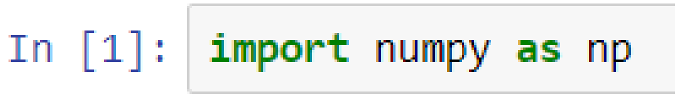
Dimensions	Example	Terminology
1	0 1 2	Vector
2	0 1 2 / 3 4 5 / 6 7 8	Matrix
3	0 1 2 / 3 4 5 / 6 7 8	3D Array (3rd order Tensor)
N	... ...	ND Array

Numpy

Numpy

- Numpy is a python library used for working with array
- Numpy also can be used to work with linear algebra, matrix operation, and any advance math operation
- Numpy stand for = Numerical Python

How to use numpy in python ?

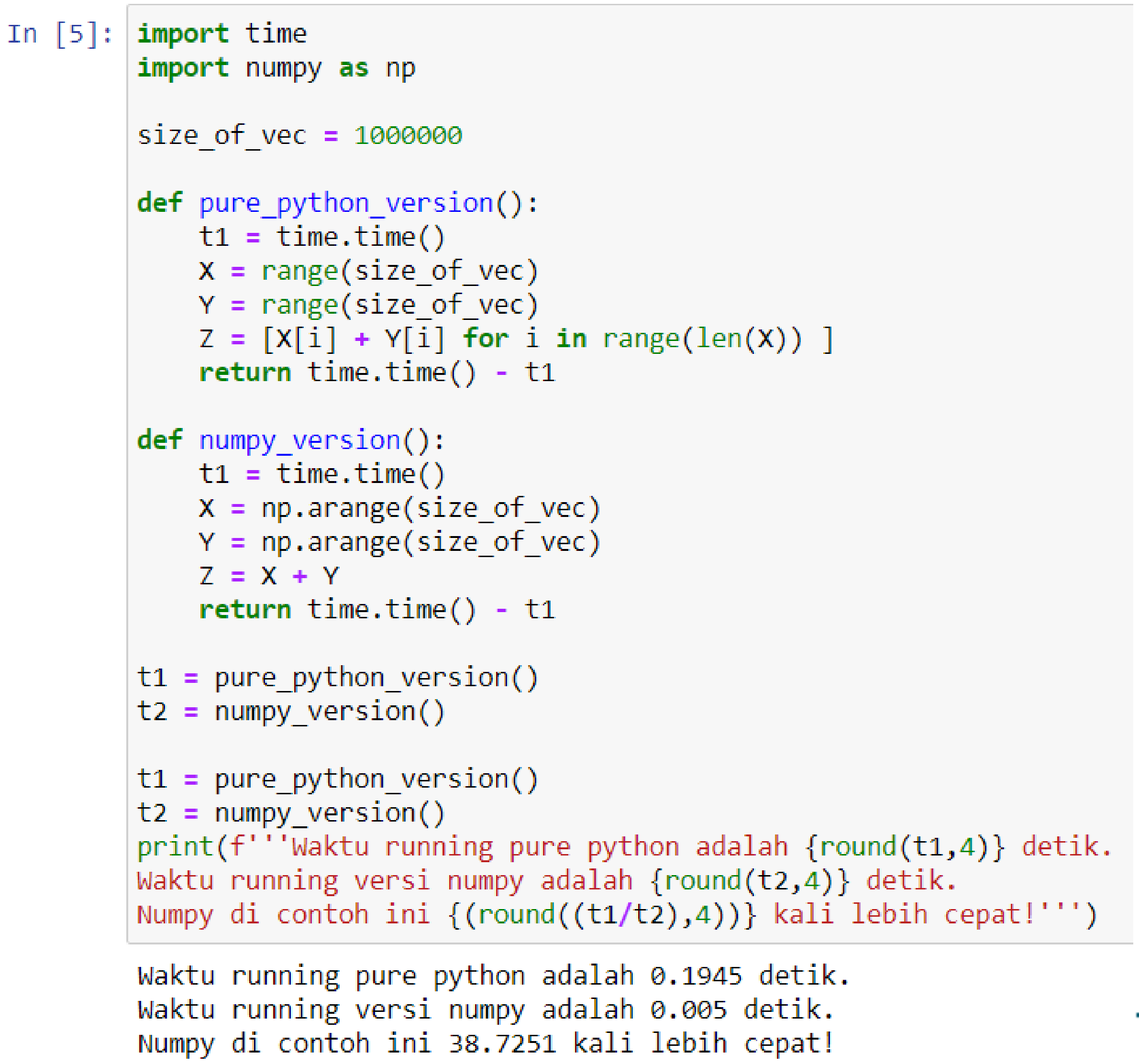


```
In [1]: import numpy as np
```

Why should we use numpy ?

- Array are 50x faster than python list
- Numpy array has a lot of supported function
- Array area frequently used in data science, where speed and resource are very important

How faster is Numpy?



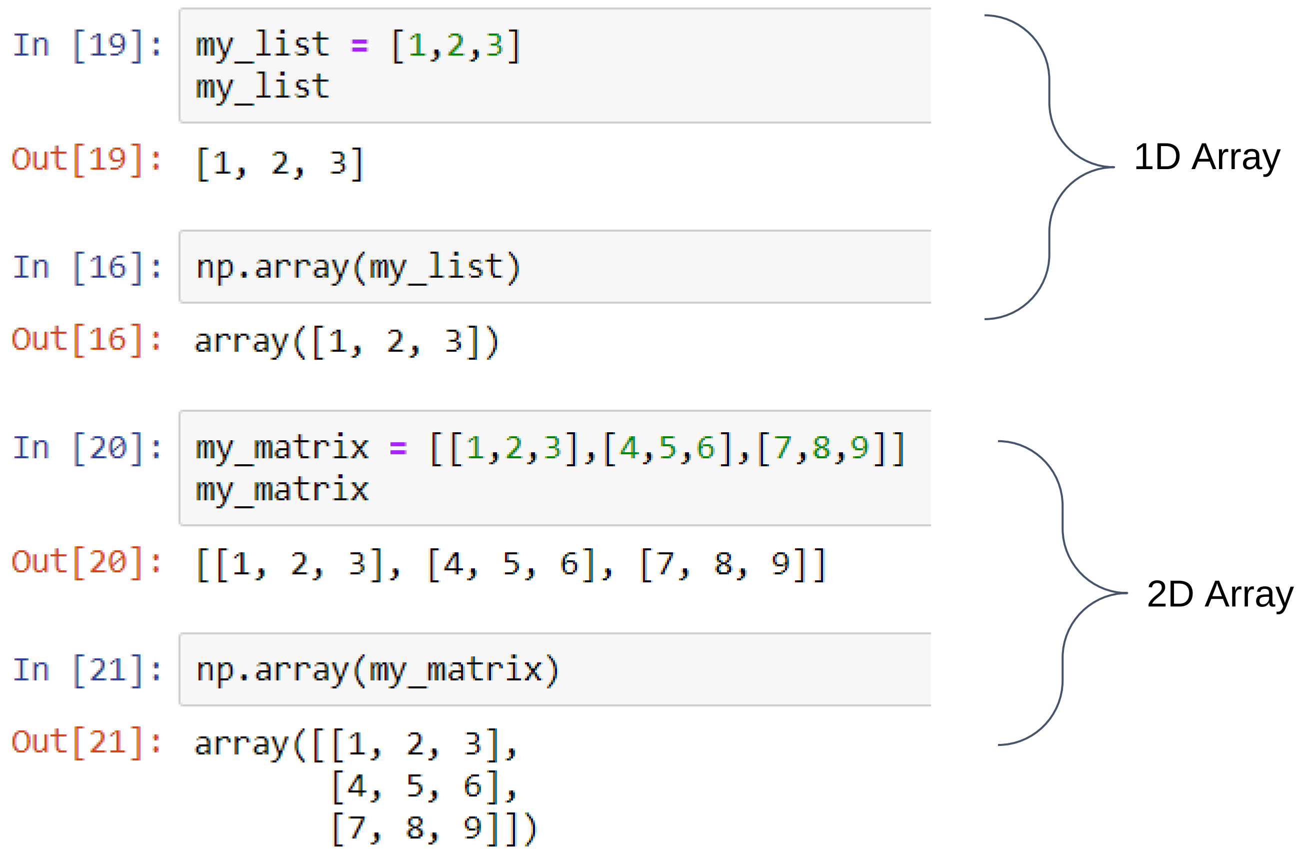
```
In [5]: import time
        import numpy as np

        size_of_vec = 1000000

        def pure_python_version():
            t1 = time.time()
            X = range(size_of_vec)
            Y = range(size_of_vec)
            Z = [X[i] + Y[i] for i in range(len(X)) ]
            return time.time() - t1

        def numpy_version():
            t1 = time.time()
            X = np.arange(size_of_vec)
            Y = np.arange(size_of_vec)
            Z = X + Y
            return time.time() - t1

        t1 = pure_python_version()
        t2 = numpy_version()

        t1 = pure_python_version()
        t2 = numpy_version()
        print(f'''Waktu running pure python adalah {round(t1,4)} detik.
        Waktu running versi numpy adalah {round(t2,4)} detik.
        Numpy di contoh ini {(round((t1/t2),4))} kali lebih cepat!''')
```

```
Waktu running pure python adalah 0.1945 detik.
Waktu running versi numpy adalah 0.005 detik.
Numpy di contoh ini 38.7251 kali lebih cepat!
```

Purwadhika
Startup and Coding School

Creating Numpy Arrays

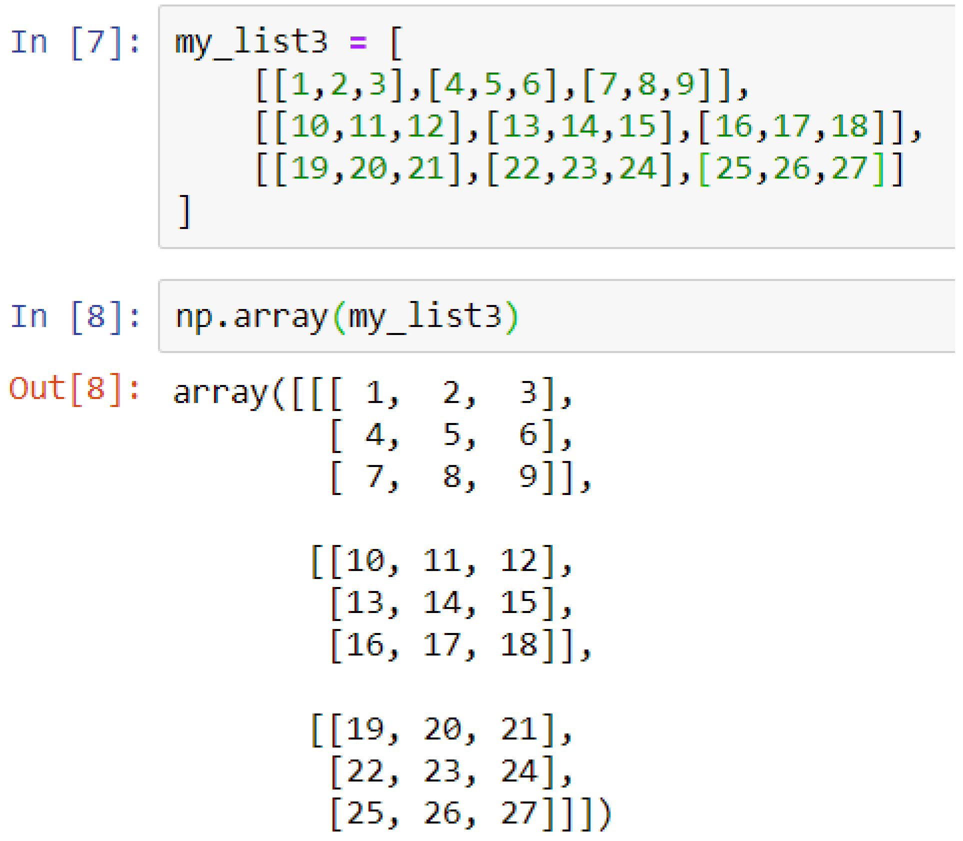
From a Python List



```
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]])
```

Creating Numpy Arrays

From a Python List



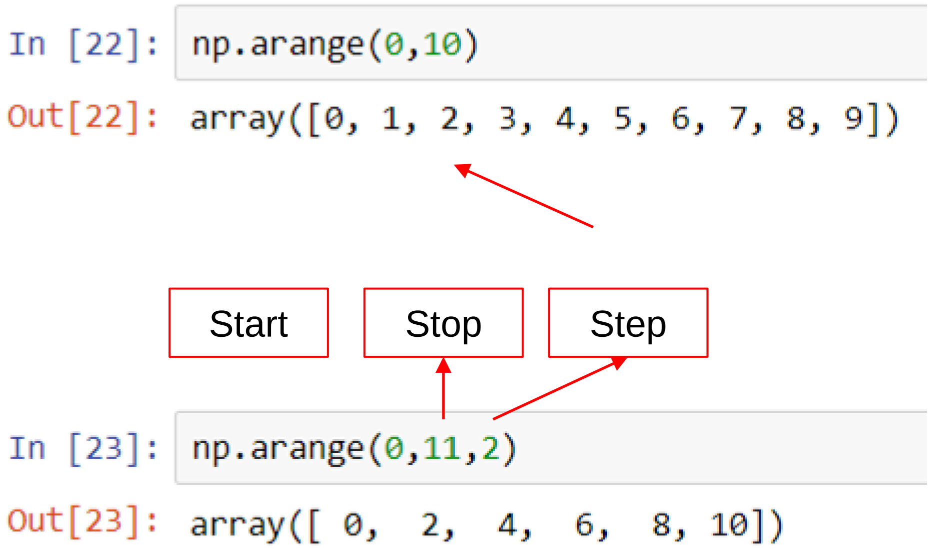
```
In [7]: my_list3 = [
            [[1,2,3],[4,5,6],[7,8,9]],
            [[10,11,12],[13,14,15],[16,17,18]],
            [[19,20,21],[22,23,24],[25,26,27]]
        ]
```

```
In [8]: np.array(my_list3)

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

               [[10, 11, 12],
                [13, 14, 15],
                [16, 17, 18]],

               [[19, 20, 21],
                [22, 23, 24],
                [25, 26, 27]]])
```

3D Array

Creating Numpy Arrays

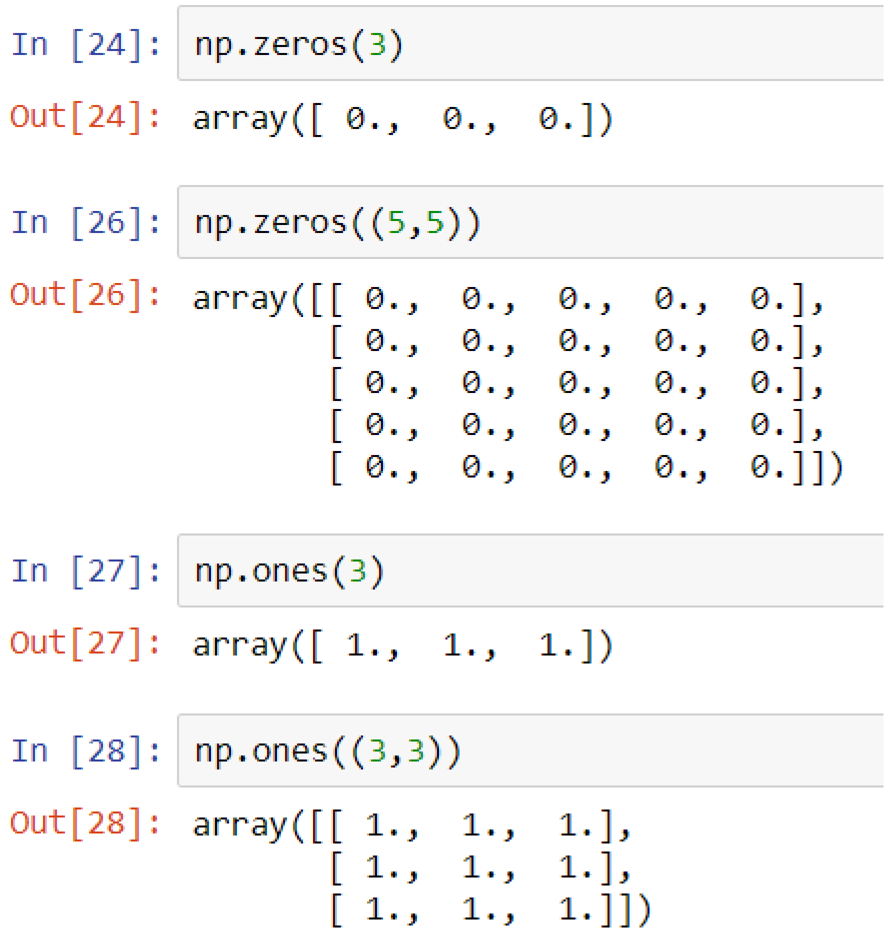
arange



```
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])
```

Creating Numpy Arrays

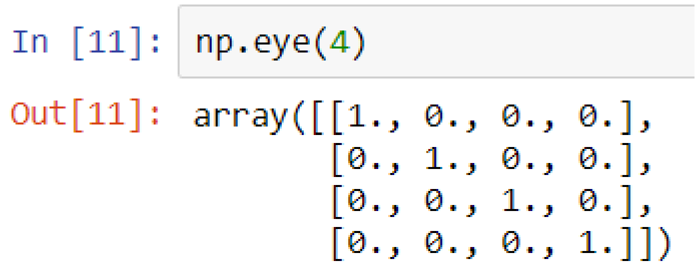
zeros and ones



```
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.]])
```

Creating Numpy Arrays

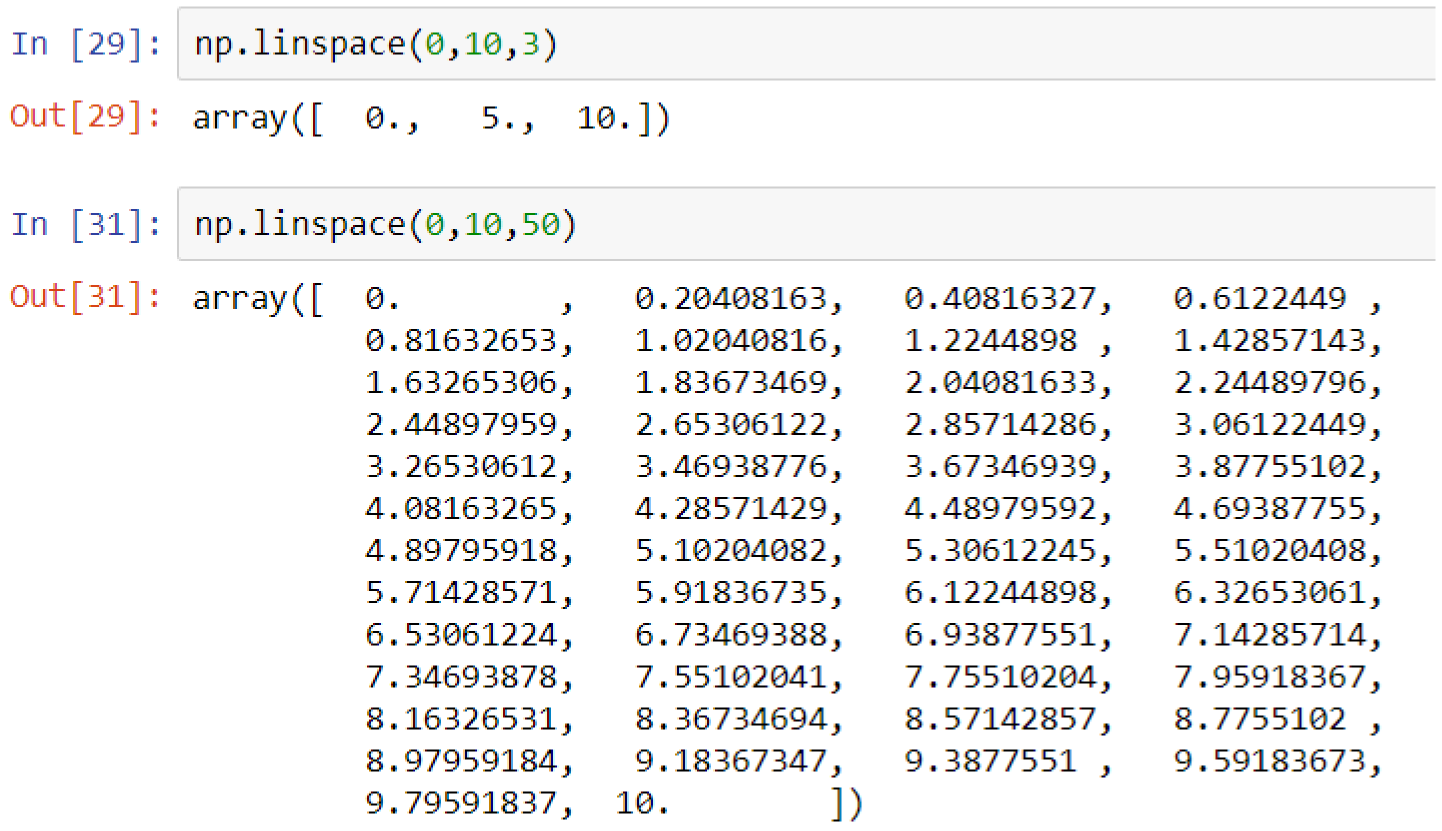
eye



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

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

Creating Numpy Arrays

linspace

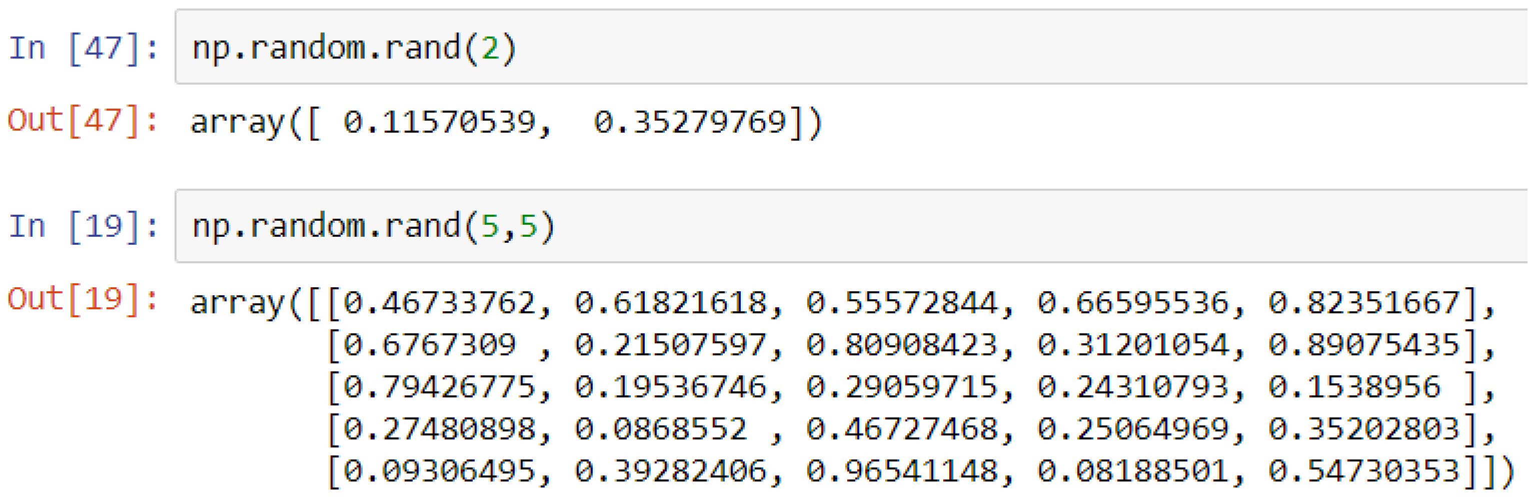


```
In [29]: np.linspace(0,10,3)

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.        ])
```

Purwadhika
Startup and Coding School

Creating Numpy Arrays

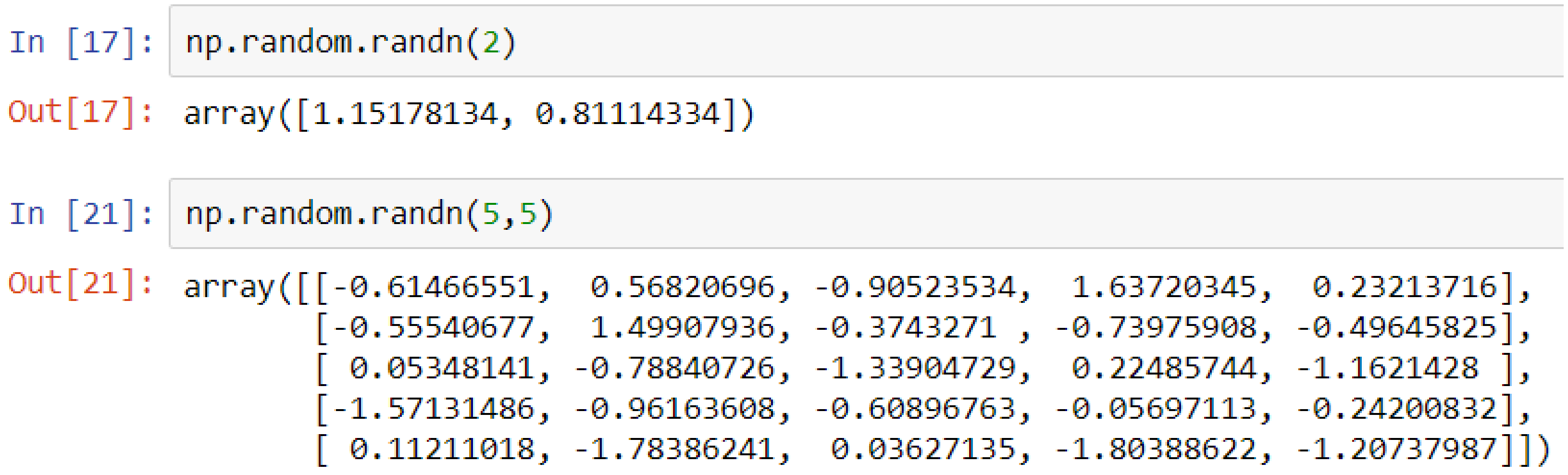
random.rand



```
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]])
```

Purwadhika
Startup and Coding School

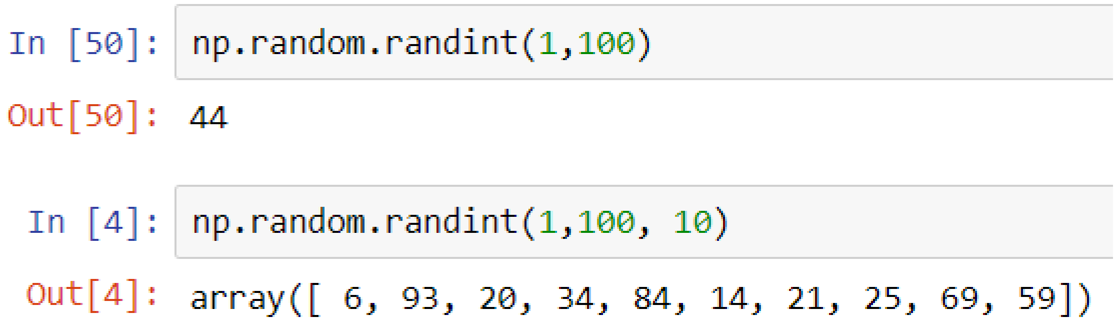
Creating Numpy Arrays
random.randn



```
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]])
```

Purwadhika
Startup and Coding School

Creating Numpy Arrays
random.randint



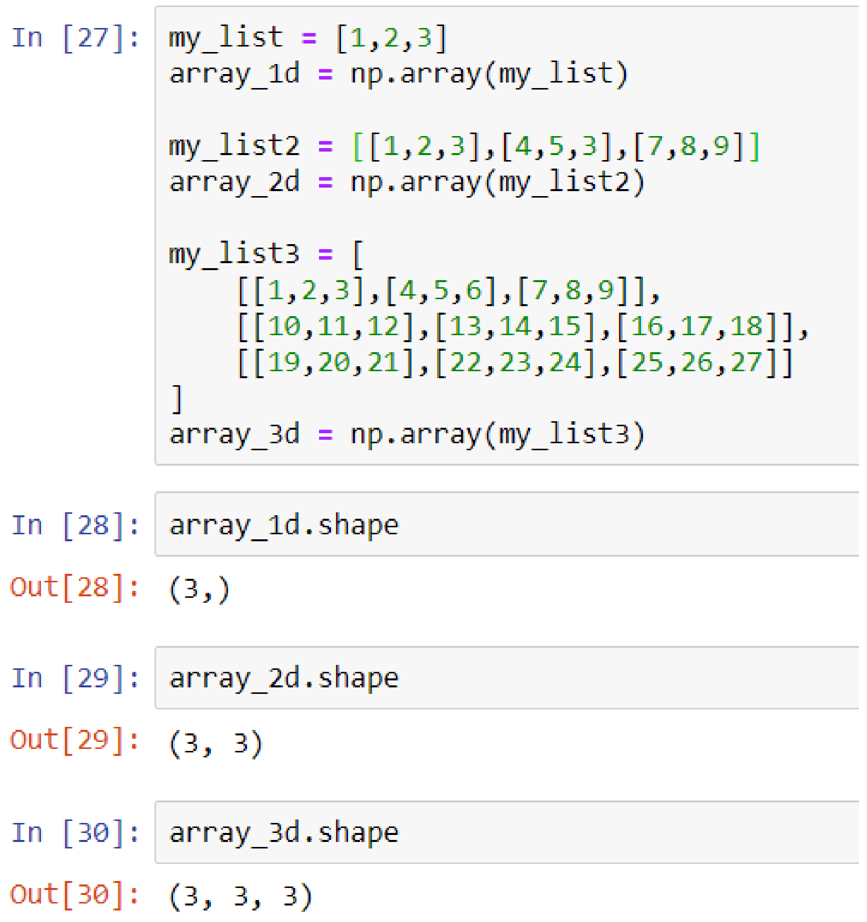
```
In [50]: np.random.randint(1,100)
Out[50]: 44

 In [4]: np.random.randint(1,100, 10)
 Out[4]: array([ 6, 93, 20, 34, 84, 14, 21, 25, 69, 59])
```

Attributes and Methods for Numpy Array

Purwadhika
Startup and Coding School

Array Attributes and Methods

shape

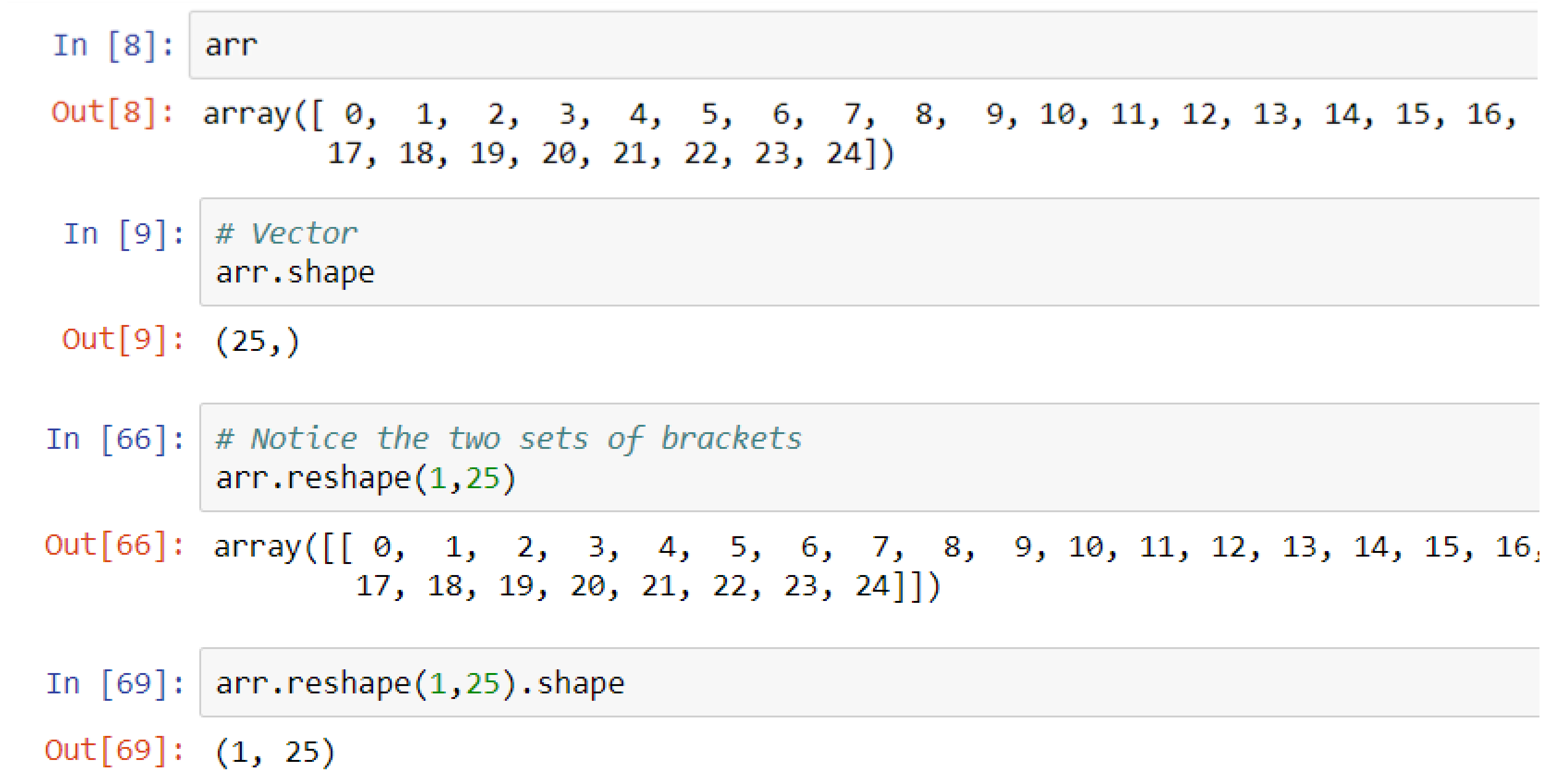


```
In [27]: my_list = [1,2,3]
         array_1d = np.array(my_list)

         my_list2 = [[1,2,3],[4,5,3],[7,8,9]]
         array_2d = np.array(my_list2)

         my_list3 = [
             [[1,2,3],[4,5,6],[7,8,9]],
             [[10,11,12],[13,14,15],[16,17,18]],
             [[19,20,21],[22,23,24],[25,26,27]]
         ]
         array_3d = np.array(my_list3)

In [28]: array_1d.shape

Out[28]: (3,)

In [29]: array_2d.shape

Out[29]: (3, 3)

In [30]: array_3d.shape

Out[30]: (3, 3, 3)
```

Purwadhika
Startup and Coding School

Array Attributes and Methods

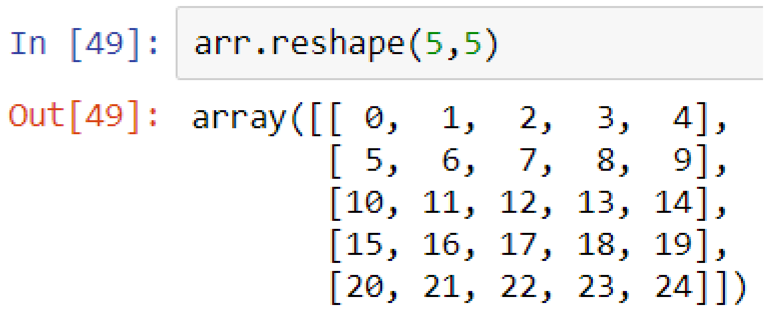
reshape



```
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 [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)
```

Purwadhika
Startup and Coding School

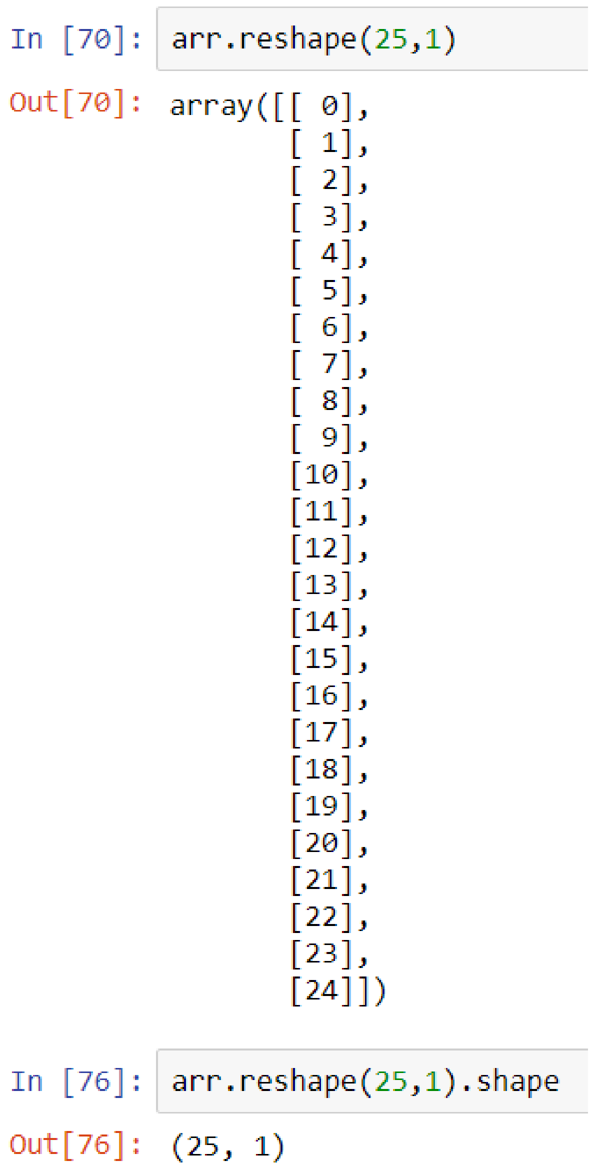
Array Attributes and Methods
reshape



```
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

reshape

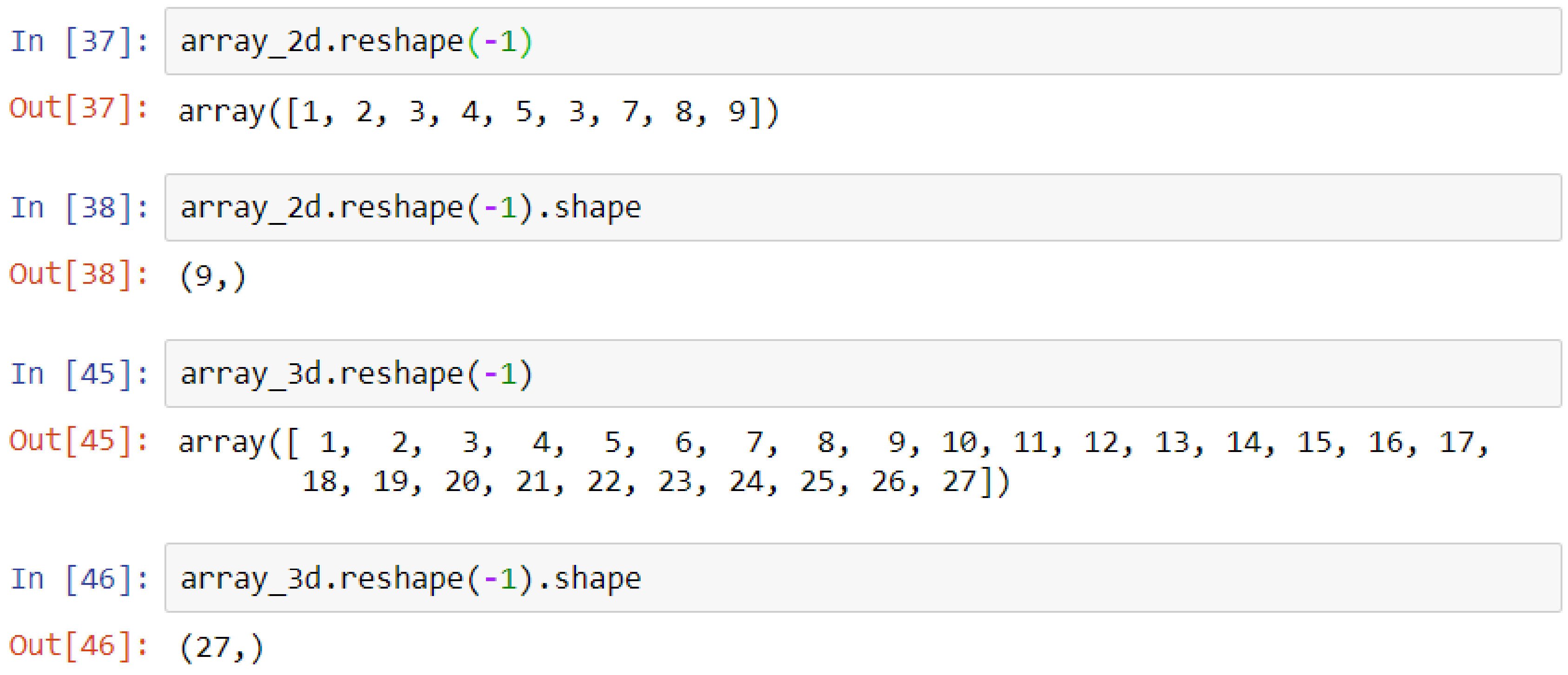


```
In [70]: arr.reshape(25,1)

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

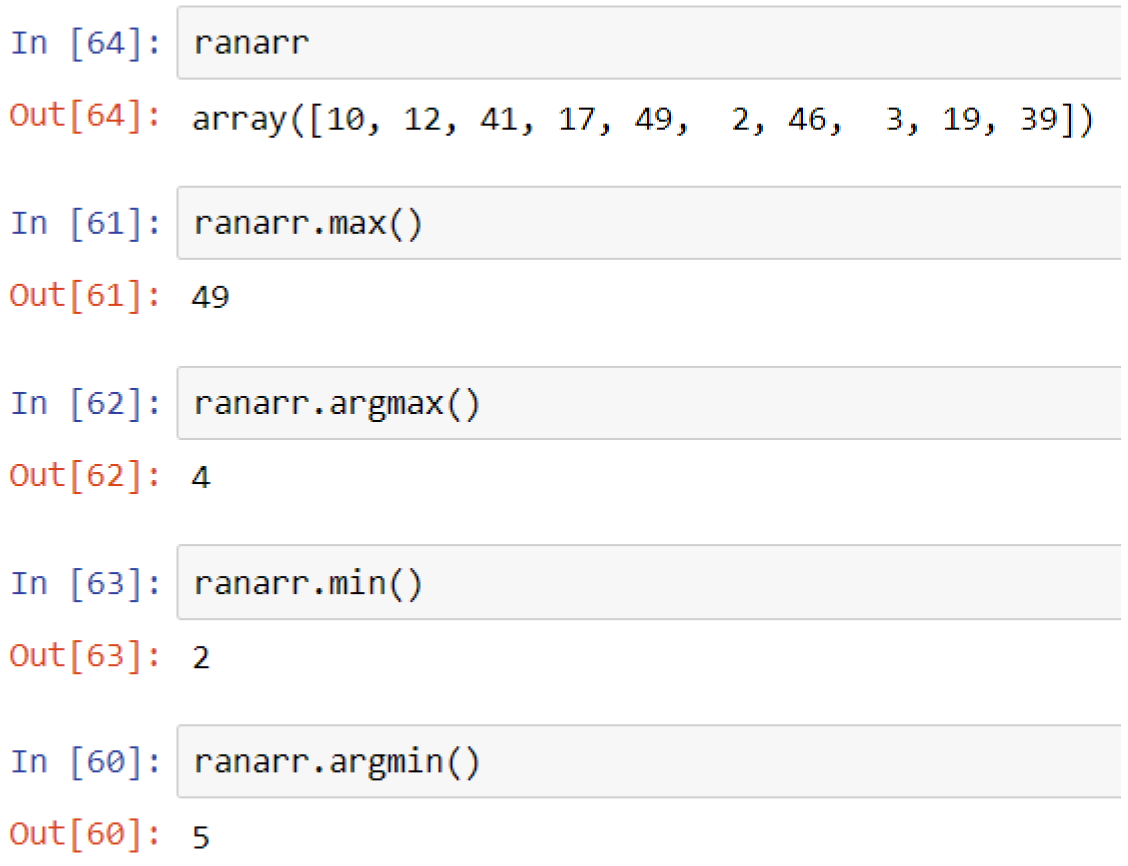
reshape



```
In [37]: array_2d.reshape(-1)

Out[37]: array([1, 2, 3, 4, 5, 3, 7, 8, 9])

In [38]: array_2d.reshape(-1).shape

Out[38]: (9,)

In [45]: array_3d.reshape(-1)

Out[45]: array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17,
                18, 19, 20, 21, 22, 23, 24, 25, 26, 27])

In [46]: array_3d.reshape(-1).shape

Out[46]: (27,)
```

Purwadhika
Startup and Coding School

Array Attributes and Methods
max, min, argmax, argmin

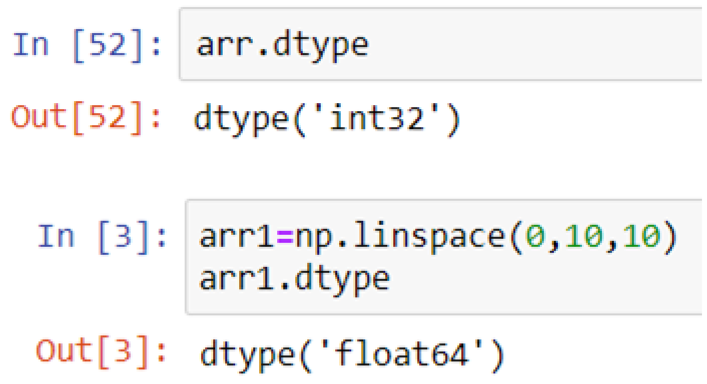


```
In [64]: ranarr
Out[64]: array([10, 12, 41, 17, 49,  2, 46,  3, 19, 39])

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
```

Purwadhika
Startup and Coding School

Array Attributes and Methods

dtype



```
In [52]: arr.dtype

Out[52]: dtype('int32')

In [3]: arr1=np.linspace(0,10,10)
        arr1.dtype

Out[3]: dtype('float64')
```

Purwadhika
Startup and Coding School

Array Dimension

Purwadhika
Startup and Coding School

Reference

- Python Lists vs. Numpy Arrays - What is the difference? https://webcourses.ucf.edu/courses/1249560/pages/python-lists-vs-numpy-arrays-what-is-the-difference
- Numpy Routines. https://numpy.org/doc/stable/reference/routines.html
- Why do we Use a Multidimensional Array? http://www.geekinterview.com/question_details/20396
- Min, Max dan Range. https://www.nedarc.org/statisticalhelp/basicStatistics/minAndMax.html

Purwadhika
Startup and Coding School