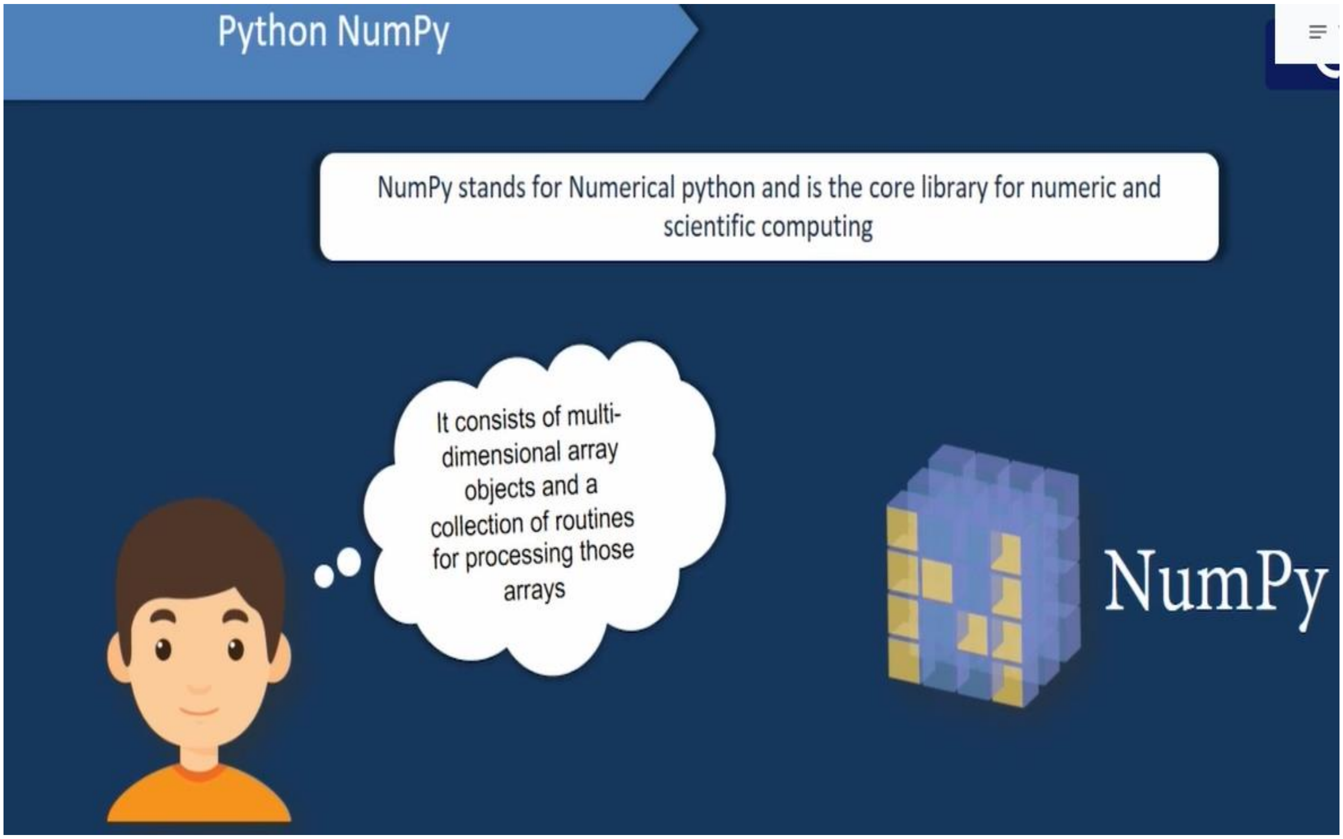
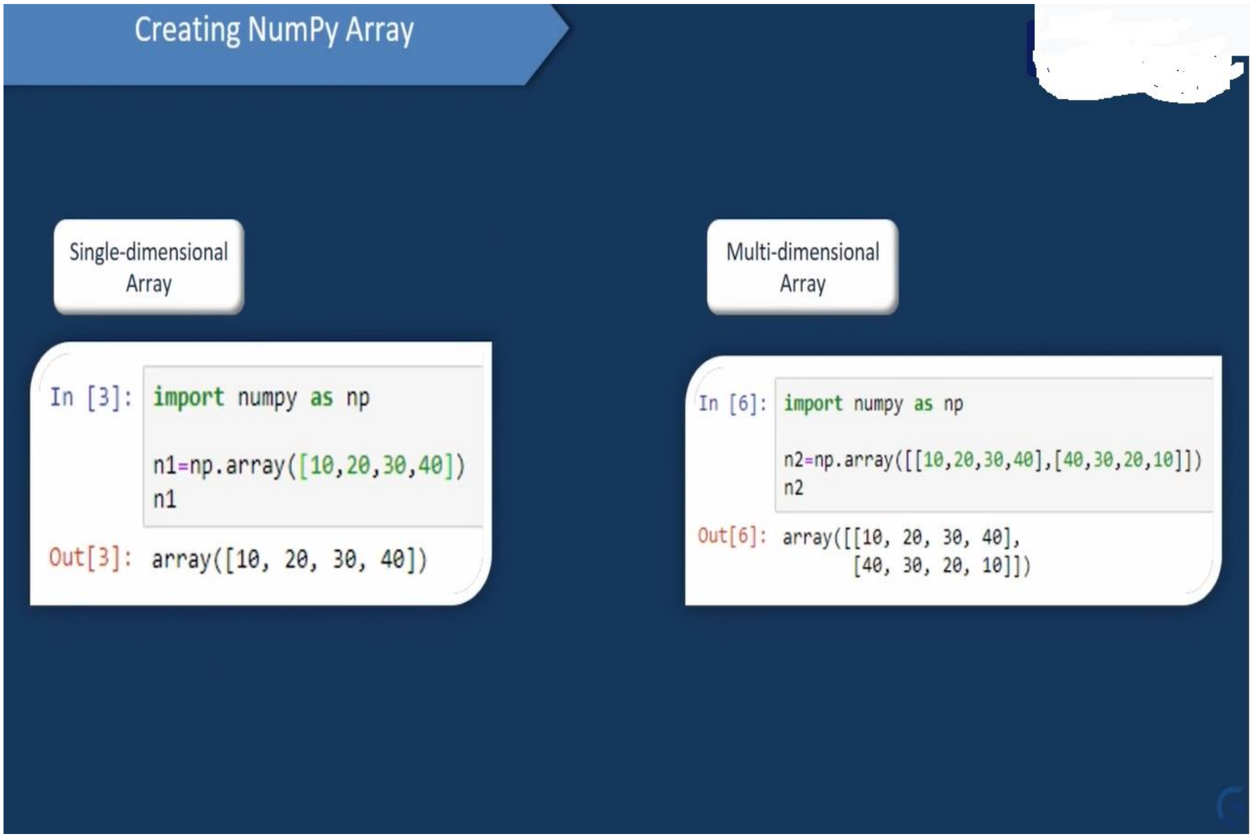


Python NumPy

NumPy stands for Numerical python and is the core library for numeric and scientific computing

It consists of multi-dimensional array objects and a collection of routines for processing those arrays



Creating NumPy Array

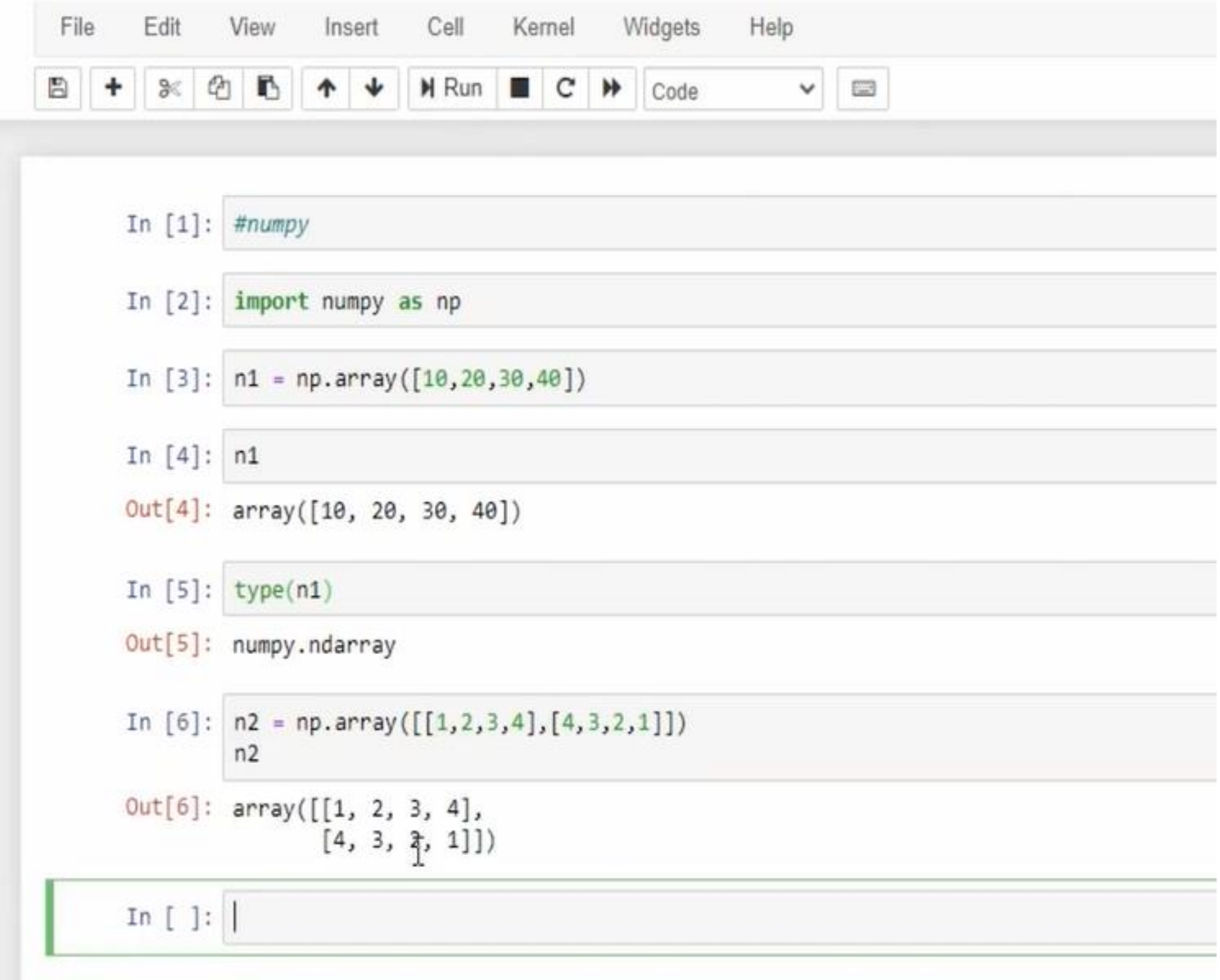
Single-dimensional Array

```
In [3]: import numpy as np

        n1=np.array([10,20,30,40])
        n1

Out[3]: array([10, 20, 30, 40])
```

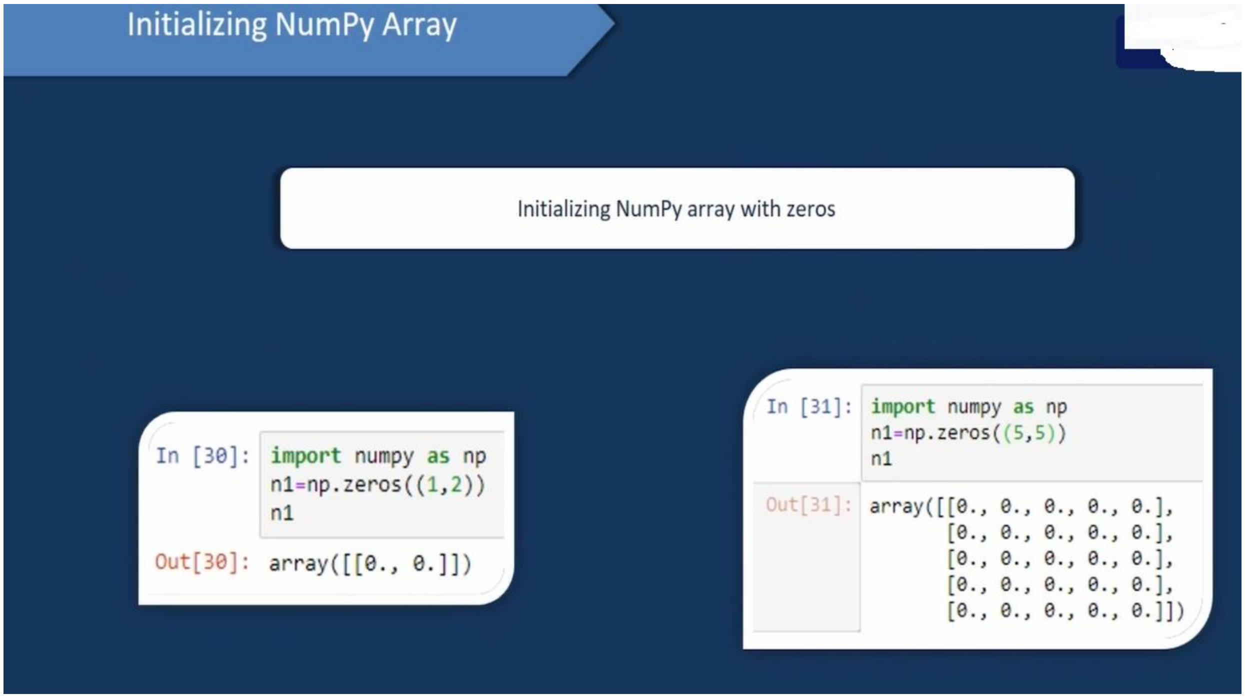
Multi-dimensional Array

```
In [6]: import numpy as np

        n2=np.array([[10,20,30,40],[40,30,20,10]])
        n2

Out[6]: array([[10, 20, 30, 40],
               [40, 30, 20, 10]])
```

(Passing single list)↑ (Passing multiple lists)↑



```
In [1]: #numpy

In [2]: import numpy as np

In [3]: n1 = np.array([10,20,30,40])

In [4]: n1

Out[4]: array([10, 20, 30, 40])

In [5]: type(n1)

Out[5]: numpy.ndarray

In [6]: n2 = np.array([[1,2,3,4],[4,3,2,1]])
        n2

Out[6]: array([[1, 2, 3, 4],
               [4, 3, 2, 1]])

In [ ]: 
```

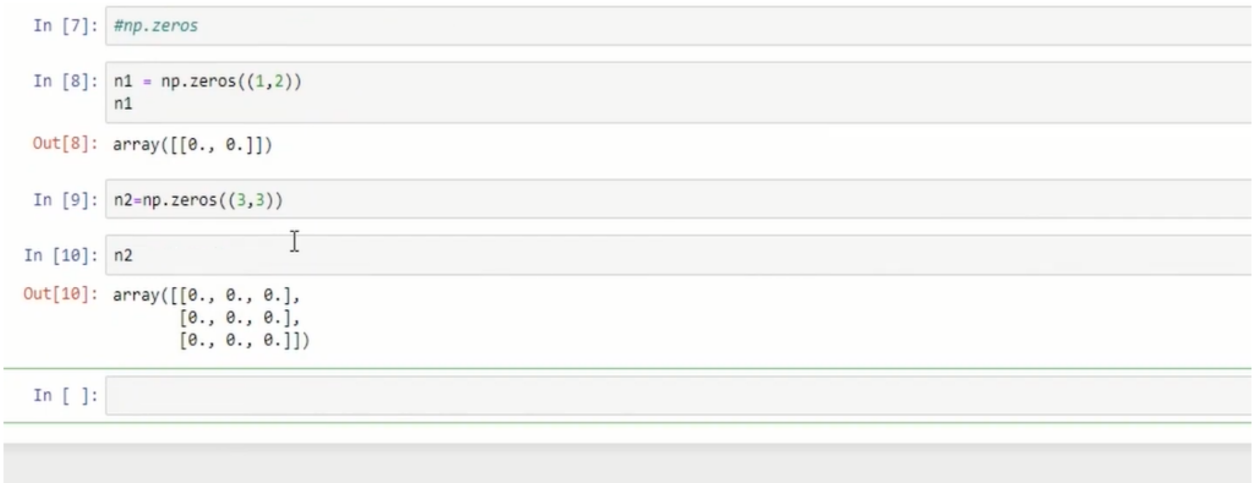


Initializing NumPy Array

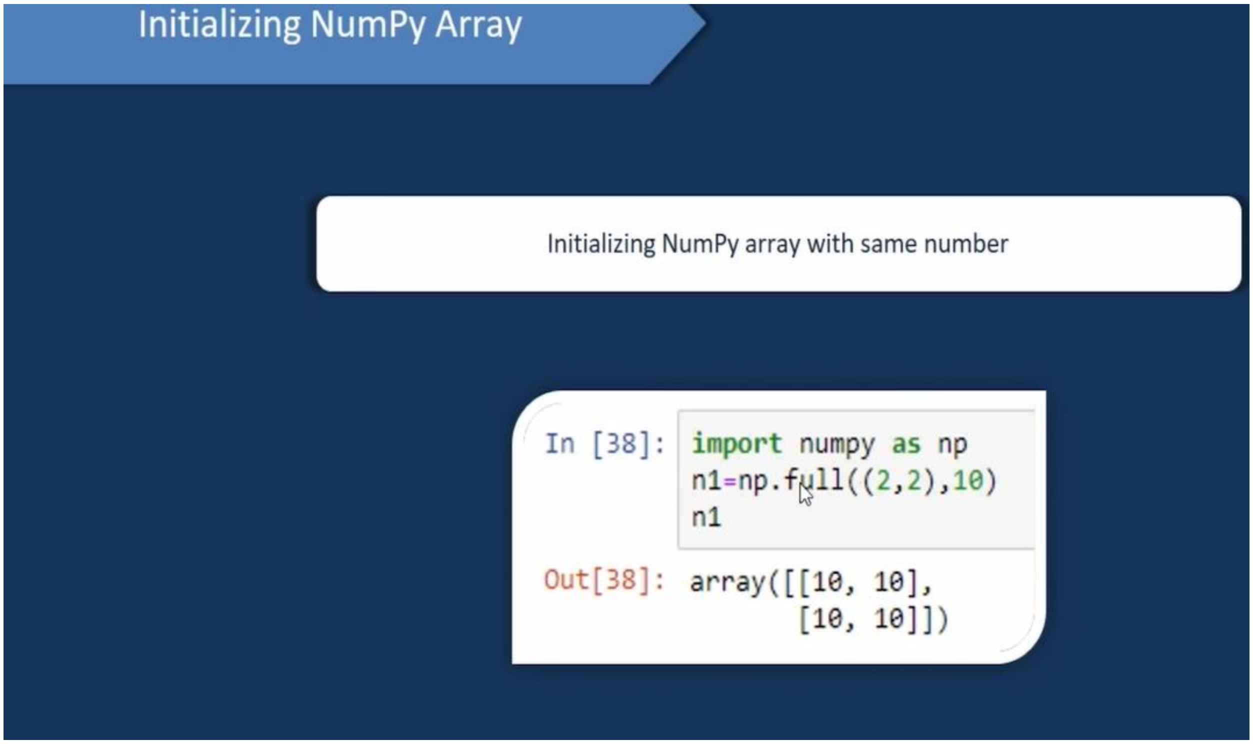
Initializing NumPy array with zeros

```
In [30]: import numpy as np
         n1=np.zeros((1,2))
         n1

Out[30]: array([[0., 0.]])
```

```
In [31]: import numpy as np
         n1=np.zeros((5,5))
         n1

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



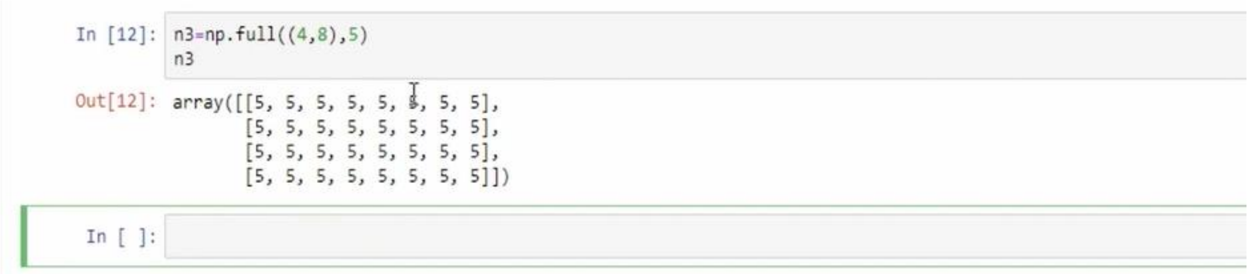
```
In [7]: #np.zeros

In [8]: n1 = np.zeros((1,2))
        n1

Out[8]: array([[0., 0.]])

In [9]: n2=np.zeros((3,3))

In [10]: n2

Out[10]: array([[0., 0., 0.],
                [0., 0., 0.],
                [0., 0., 0.]])

In [ ]:
```



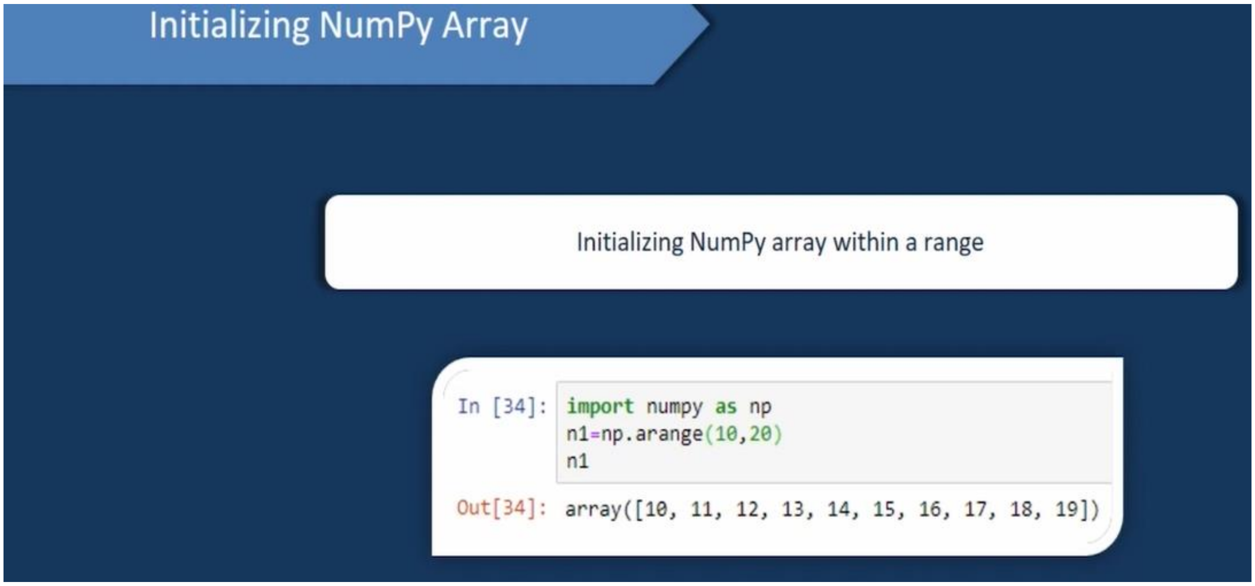
Initializing NumPy Array

Initializing NumPy array with same number

```
In [38]: import numpy as np
         n1=np.full((2,2),10)
         n1

Out[38]: array([[10, 10],
                [10, 10]])
```



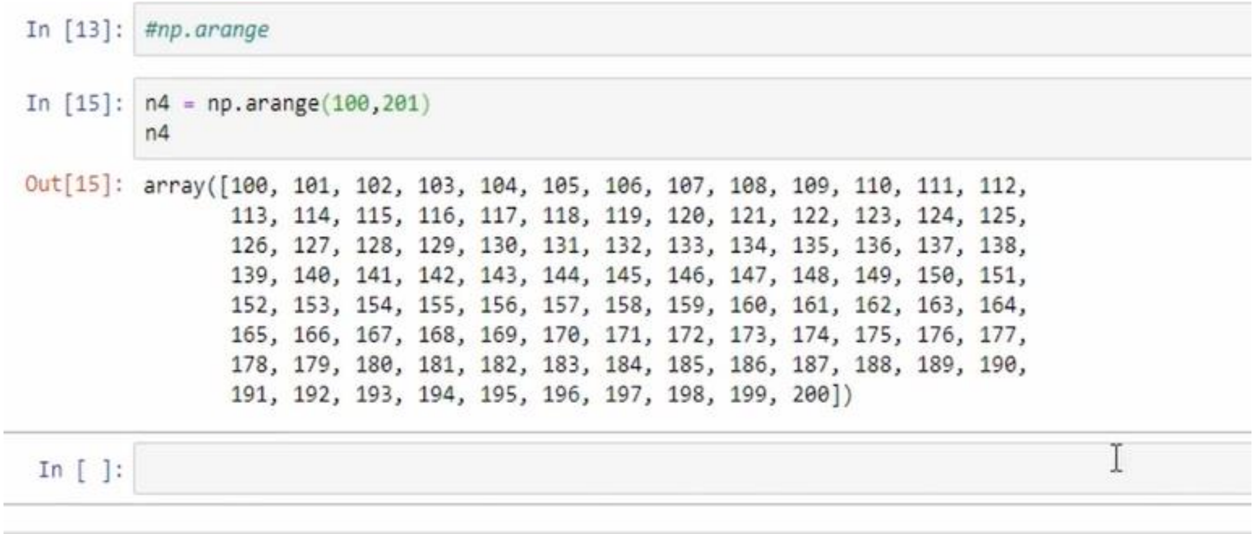
```
In [12]: n3=np.full((4,8),5)
         n3

Out[12]: array([[5, 5, 5, 5, 5, 5, 5, 5],
                [5, 5, 5, 5, 5, 5, 5, 5],
                [5, 5, 5, 5, 5, 5, 5, 5],
                [5, 5, 5, 5, 5, 5, 5, 5]])

In [ ]:
```



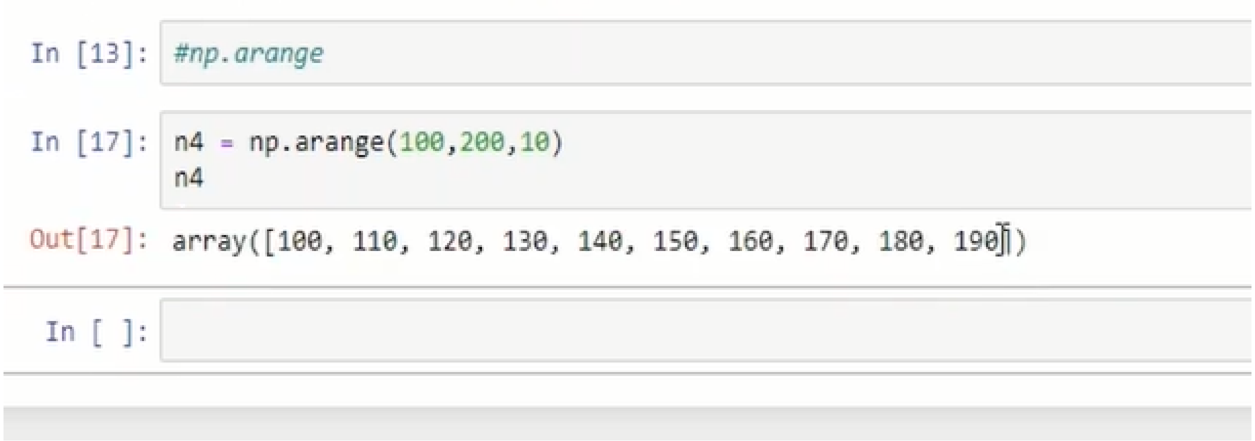
Initializing NumPy Array

Initializing NumPy array within a range

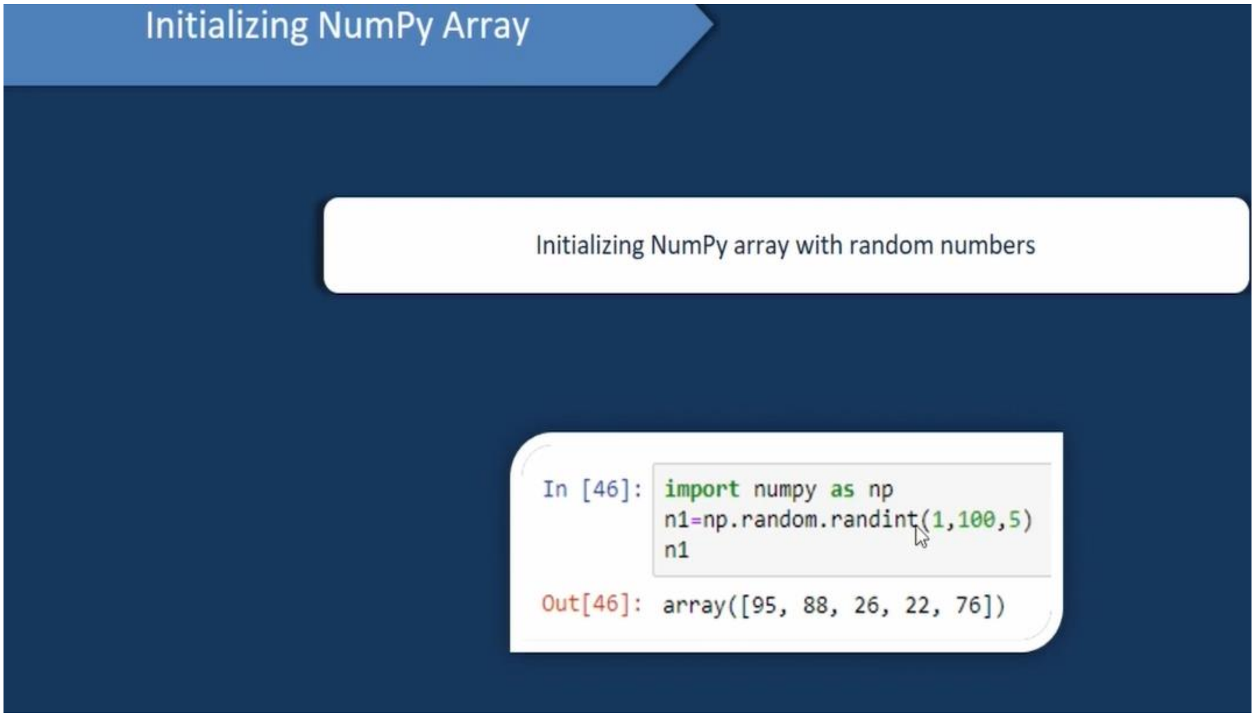
```
In [34]: import numpy as np
         n1=np.arange(10,20)
         n1

Out[34]: array([10, 11, 12, 13, 14, 15, 16, 17, 18, 19])
```



```
In [13]: #np.arange

In [15]: n4 = np.arange(100,201)
         n4

Out[15]: array([100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112,
                113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125,
                126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138,
                139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151,
                152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164,
                165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177,
                178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190,
                191, 192, 193, 194, 195, 196, 197, 198, 199, 200])

In [ ]:
```



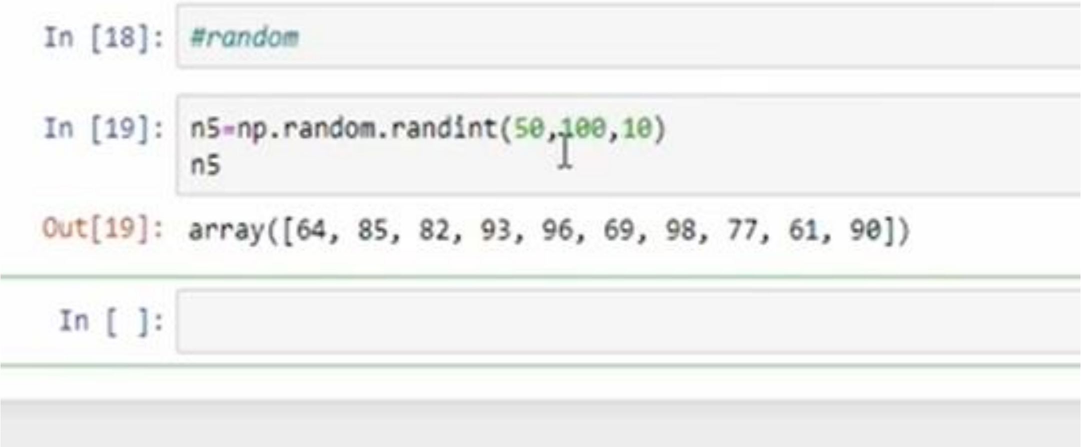
```
In [13]: #np.arange

In [17]: n4 = np.arange(100,200,10)
         n4

Out[17]: array([100, 110, 120, 130, 140, 150, 160, 170, 180, 190])

In [ ]:
```



Initializing NumPy Array

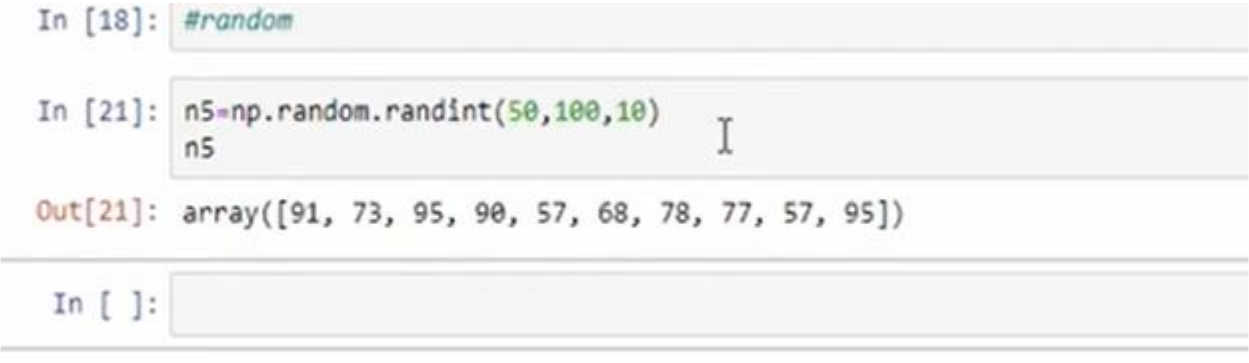
Initializing NumPy array with random numbers

```
In [46]: import numpy as np
         n1=np.random.randint(1,100,5)
         n1

Out[46]: array([95, 88, 26, 22, 76])
```

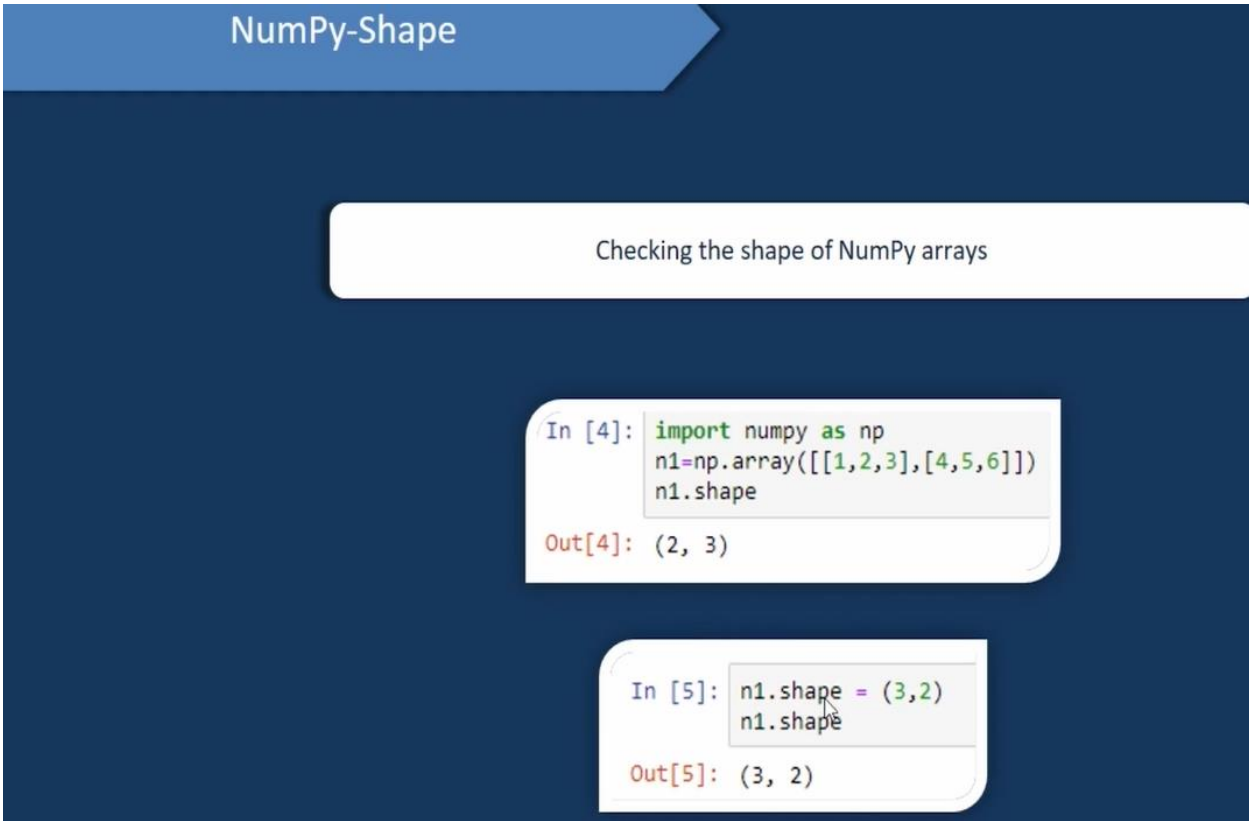


```
In [18]: #random

In [19]: n5=np.random.randint(50,100,10)
         n5

Out[19]: array([64, 85, 82, 93, 96, 69, 98, 77, 61, 90])

In [ ]:
```

If we run again, we will get different set of values.



```
In [18]: #random

In [21]: n5=np.random.randint(50,100,10)
         n5

Out[21]: array([91, 73, 95, 90, 57, 68, 78, 77, 57, 95])

In [ ]:
```

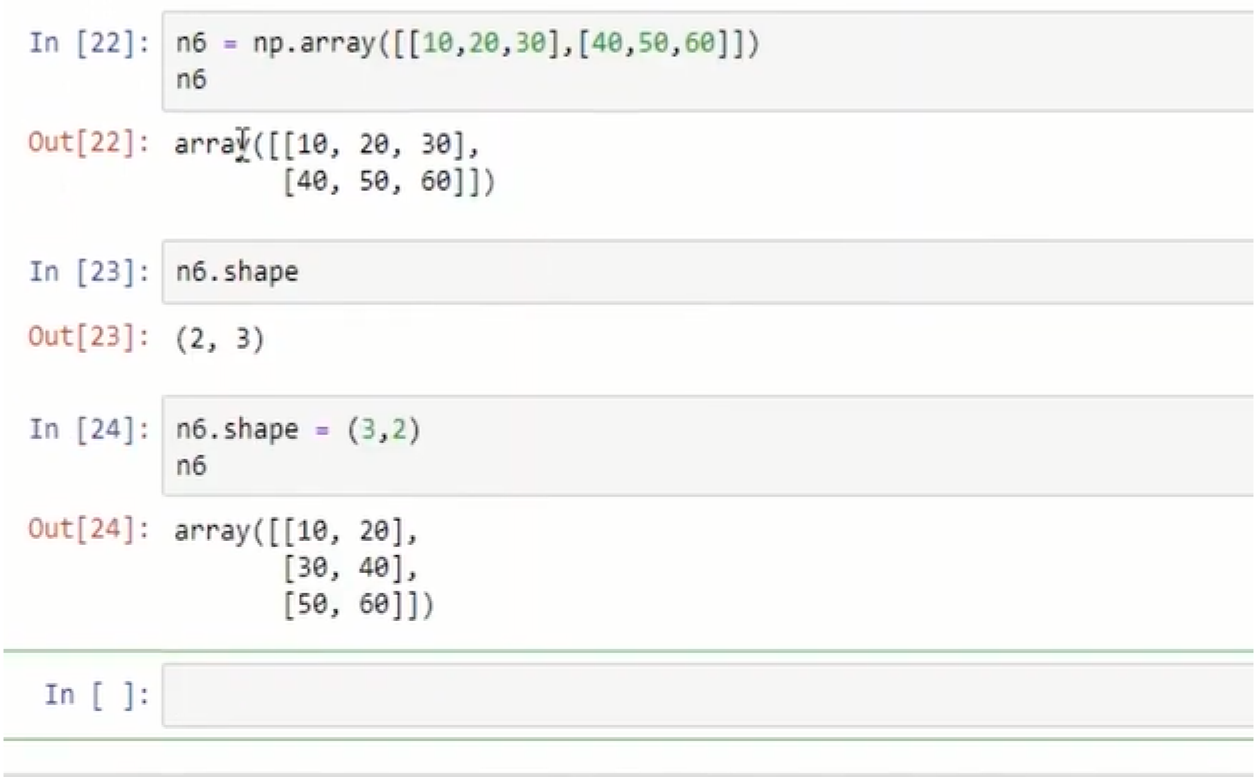


NumPy-Shape

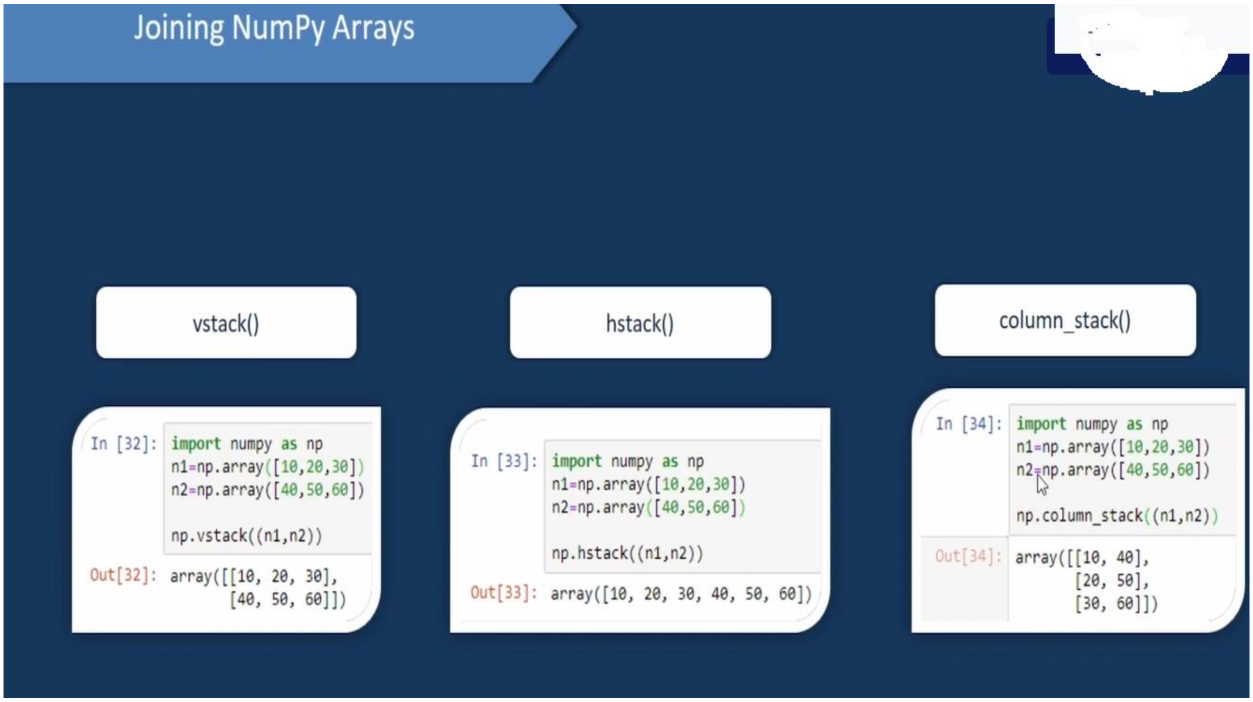
Checking the shape of NumPy arrays

```
In [4]: import numpy as np
        n1=np.array([[1,2,3],[4,5,6]])
        n1.shape

Out[4]: (2, 3)
```

```
In [5]: n1.shape = (3,2)
        n1.shape

Out[5]: (3, 2)
```



```
In [22]: n6 = np.array([[10,20,30],[40,50,60]])
         n6

Out[22]: array([[10, 20, 30],
                [40, 50, 60]])

In [23]: n6.shape

Out[23]: (2, 3)

In [24]: n6.shape = (3,2)
         n6

Out[24]: array([[10, 20],
                [30, 40],
                [50, 60]])

In [ ]:
```



Joining NumPy Arrays

vstack()

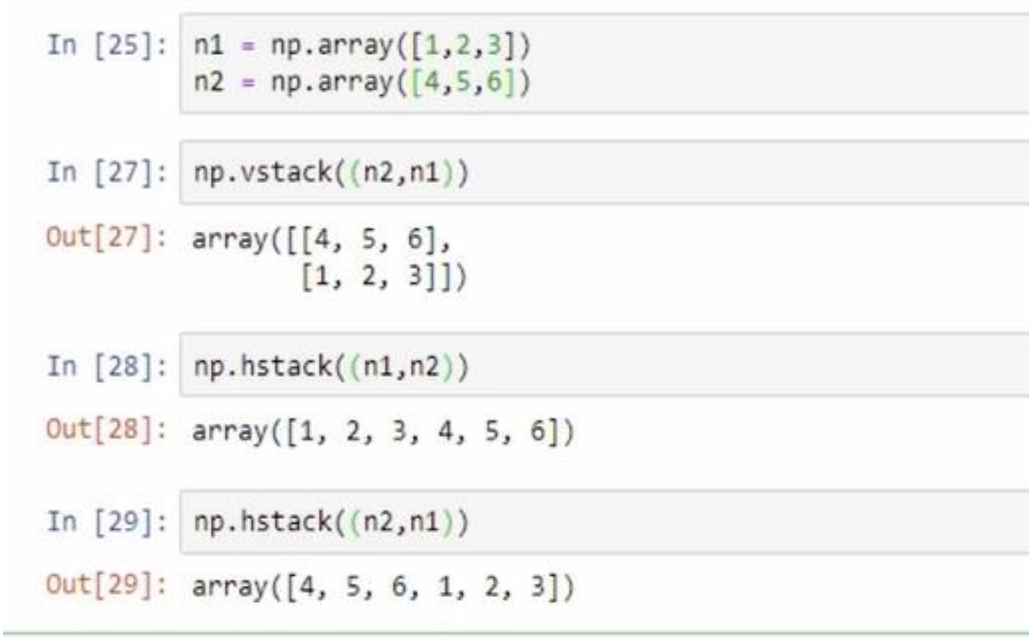
```
In [32]: import numpy as np
         n1=np.array([10,20,30])
         n2=np.array([40,50,60])

         np.vstack((n1,n2))

Out[32]: array([[10, 20, 30],
                [40, 50, 60]])
```

hstack()

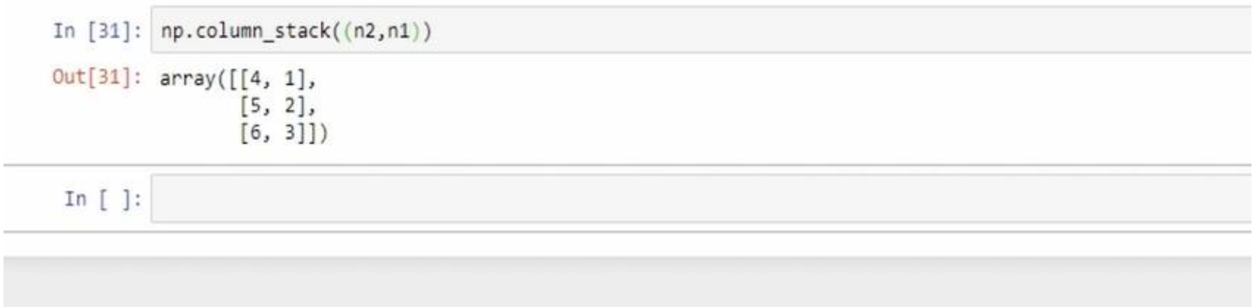
```
In [33]: import numpy as np
         n1=np.array([10,20,30])
         n2=np.array([40,50,60])

         np.hstack((n1,n2))

Out[33]: array([10, 20, 30, 40, 50, 60])
```

column_stack()

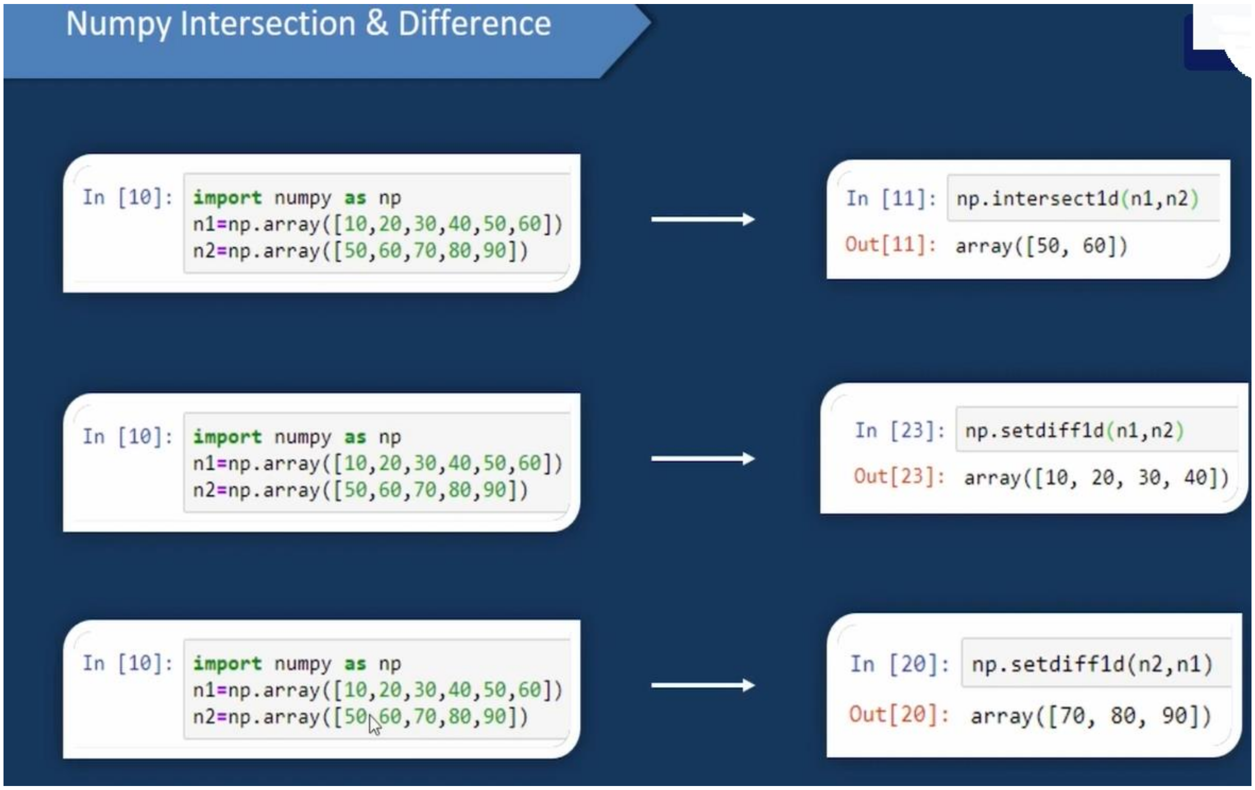
```
In [34]: import numpy as np
         n1=np.array([10,20,30])
         n2=np.array([40,50,60])

         np.column_stack((n1,n2))

Out[34]: array([[10, 40],
                [20, 50],
                [30, 60]])
```



```
In [25]: n1 = np.array([1,2,3])
         n2 = np.array([4,5,6])

In [27]: np.vstack((n2,n1))

Out[27]: array([[4, 5, 6],
                [1, 2, 3]])

In [28]: np.hstack((n1,n2))

Out[28]: array([1, 2, 3, 4, 5, 6])

In [29]: np.hstack((n2,n1))

Out[29]: array([4, 5, 6, 1, 2, 3])
```



```
In [31]: np.column_stack((n2,n1))

Out[31]: array([[4, 1],
                [5, 2],
                [6, 3]])

In [ ]:
```



Numpy Intersection & Difference

```
In [10]: import numpy as np
         n1=np.array([10,20,30,40,50,60])
         n2=np.array([50,60,70,80,90])
```

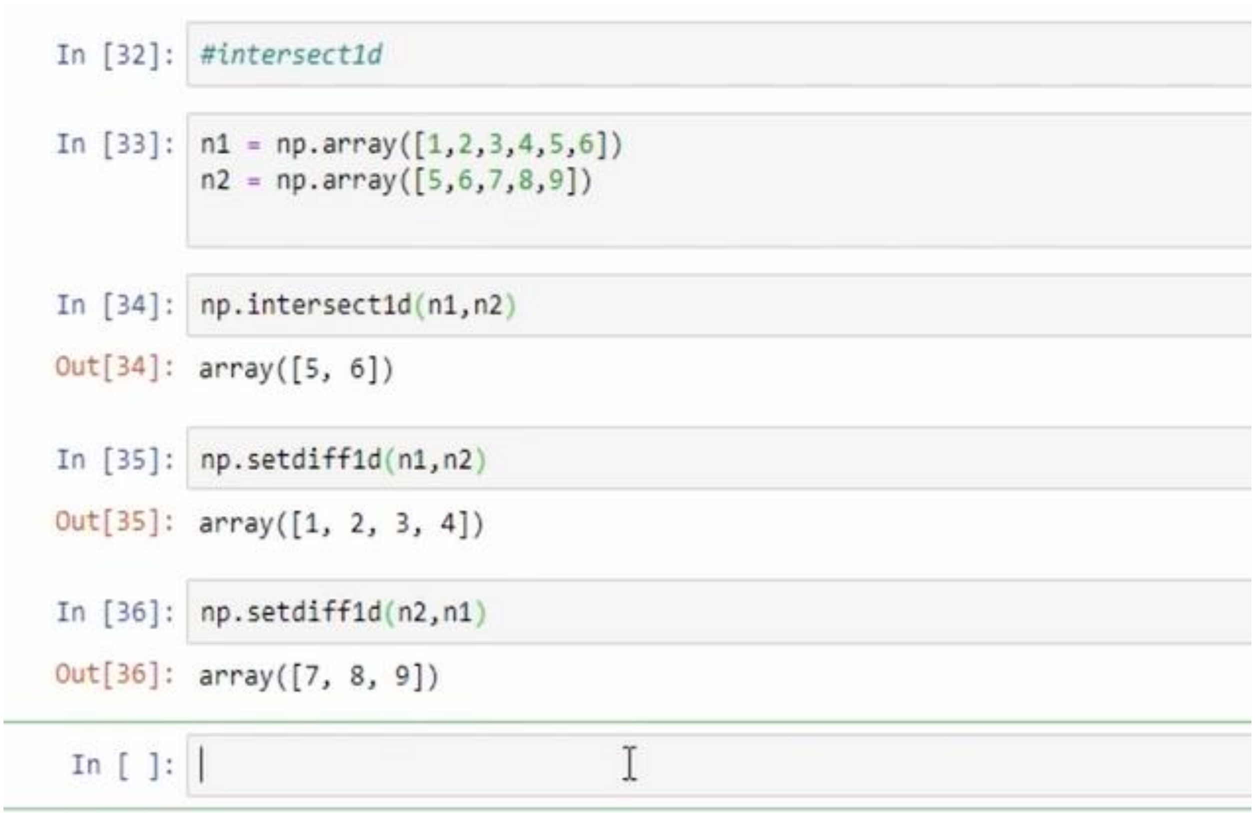
```
In [11]: np.intersect1d(n1,n2)
Out[11]: array([50, 60])
```

```
In [10]: import numpy as np
         n1=np.array([10,20,30,40,50,60])
         n2=np.array([50,60,70,80,90])
```

```
In [23]: np.setdiff1d(n1,n2)
Out[23]: array([10, 20, 30, 40])
```

```
In [10]: import numpy as np
         n1=np.array([10,20,30,40,50,60])
         n2=np.array([50,60,70,80,90])
```

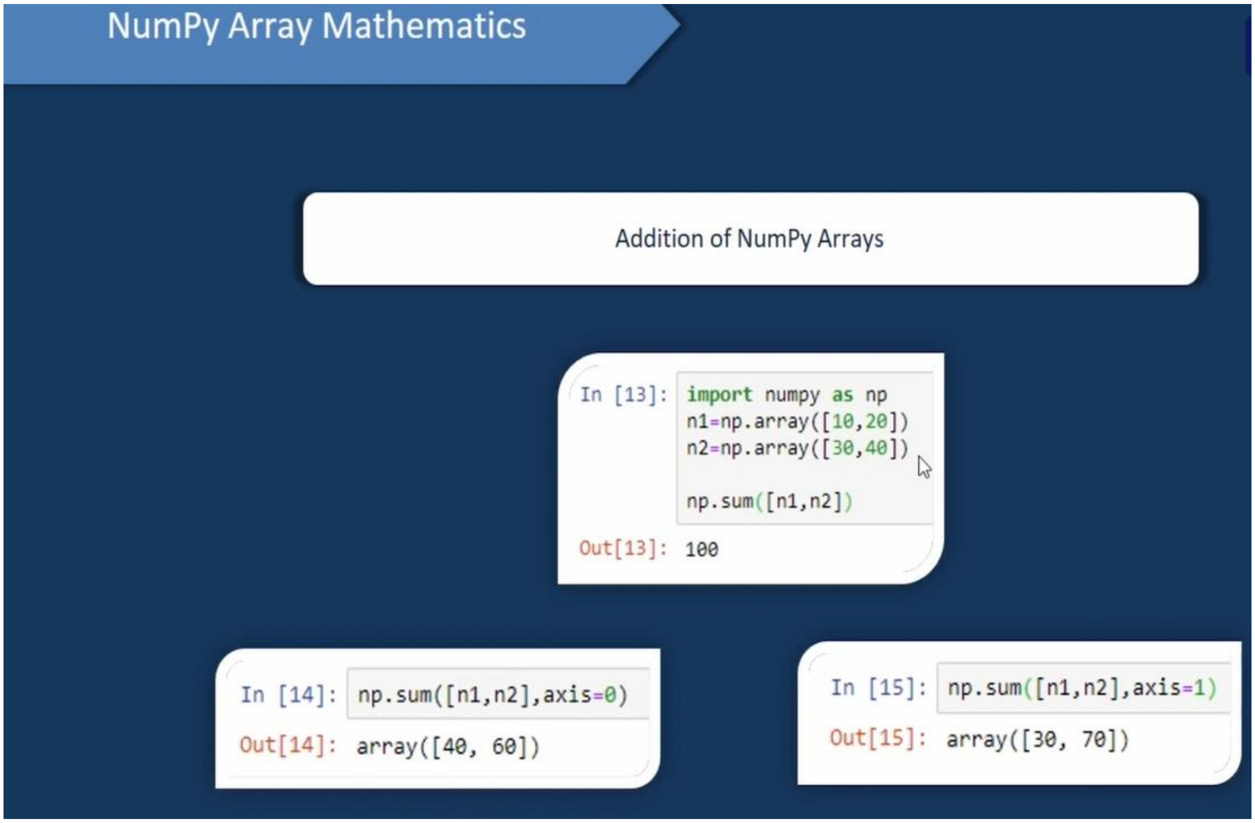
```
In [20]: np.setdiff1d(n2,n1)
Out[20]: array([70, 80, 90])
```



```
In [32]: #intersect1d

In [33]: n1 = np.array([1,2,3,4,5,6])
         n2 = np.array([5,6,7,8,9])

In [34]: np.intersect1d(n1,n2)
Out[34]: array([5, 6])

In [35]: np.setdiff1d(n1,n2)
Out[35]: array([1, 2, 3, 4])

In [36]: np.setdiff1d(n2,n1)
Out[36]: array([7, 8, 9])

In [ ]:
```

np.setiff1d(n1,n2)-> Unique elements present in n1

np.setiff1d(n2,n1)-> Unique elements present in n2



NumPy Array Mathematics

Addition of NumPy Arrays

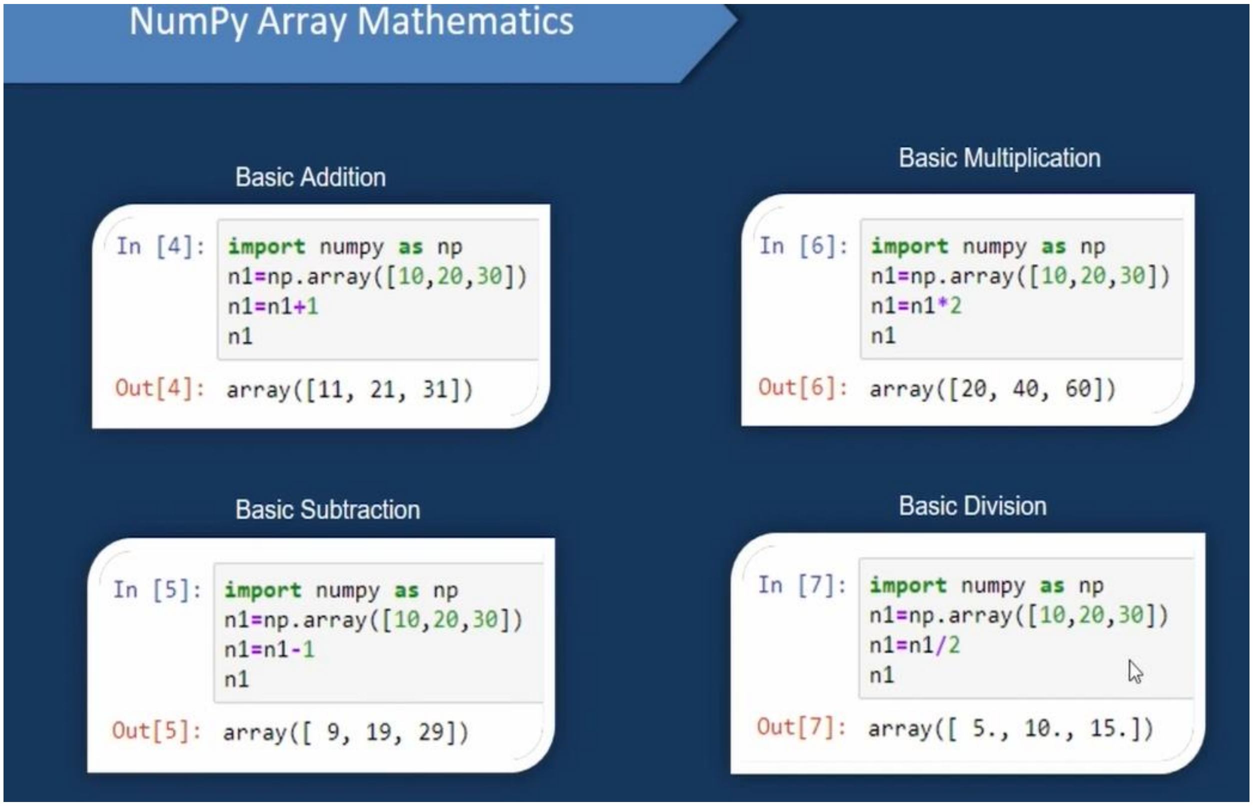
```
In [13]: import numpy as np
         n1=np.array([10,20])
         n2=np.array([30,40])

         np.sum([n1,n2])
Out[13]: 100
```

```
In [14]: np.sum([n1,n2],axis=0)
Out[14]: array([40, 60])
```

```
In [15]: np.sum([n1,n2],axis=1)
Out[15]: array([30, 70])
```

axis=0, sum the column

axis=1, sum the rows



NumPy Array Mathematics

Basic Addition

```
In [4]:  import numpy as np
         n1=np.array([10,20,30])
         n1=n1+1
         n1

Out[4]:  array([11, 21, 31])
```

Basic Multiplication

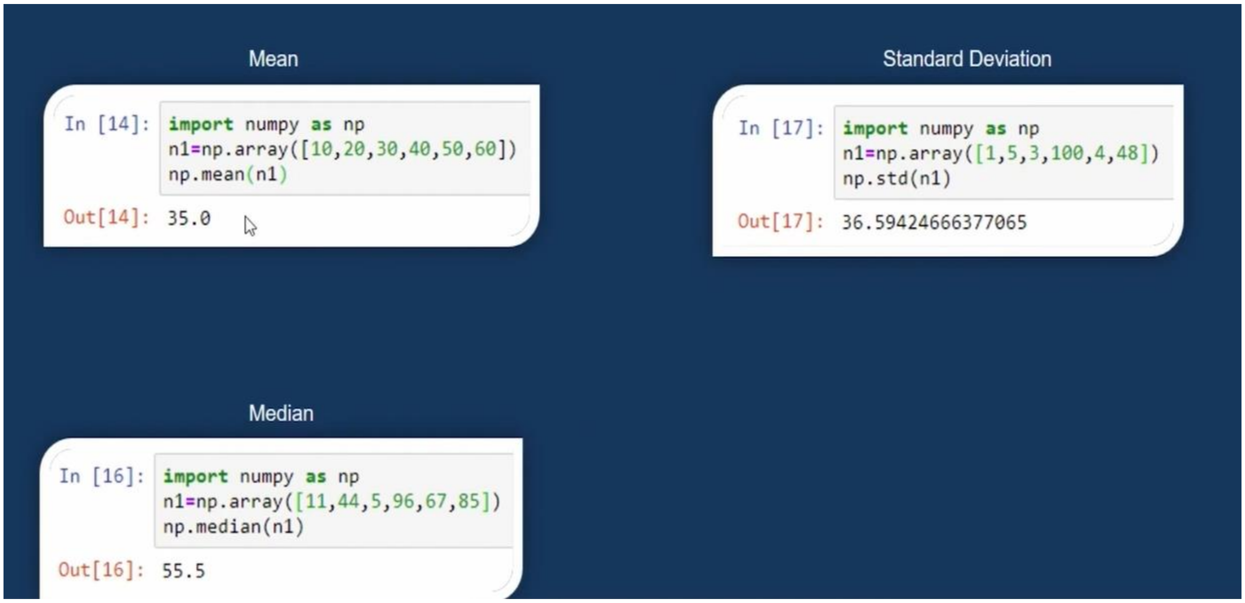
```
In [6]:  import numpy as np
         n1=np.array([10,20,30])
         n1=n1*2
         n1

Out[6]:  array([20, 40, 60])
```

Basic Subtraction

```
In [5]:  import numpy as np
         n1=np.array([10,20,30])
         n1=n1-1
         n1

Out[5]:  array([ 9, 19, 29])
```

Basic Division

```
In [7]:  import numpy as np
         n1=np.array([10,20,30])
         n1=n1/2
         n1

Out[7]:  array([ 5., 10., 15.])
```



Mean

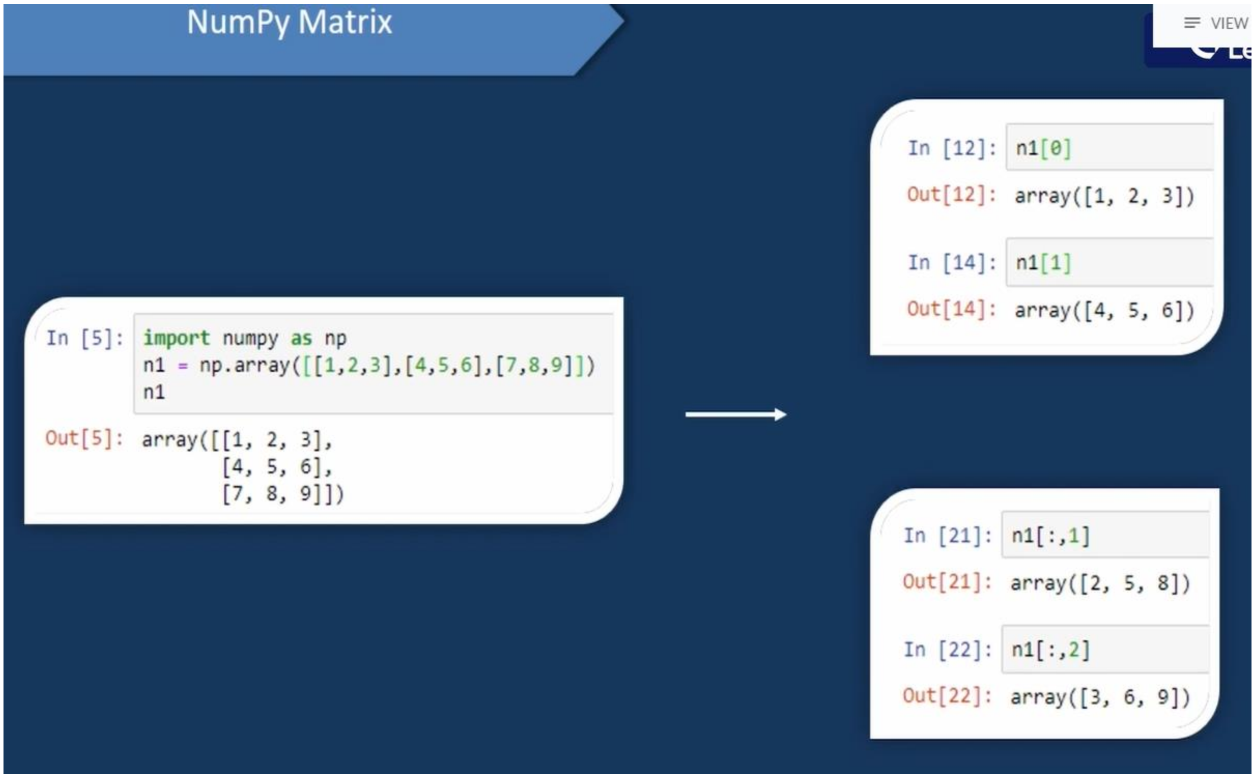
```
In [14]:  import numpy as np
          n1=np.array([10,20,30,40,50,60])
          np.mean(n1)

Out[14]:  35.0
```

Standard Deviation

```
In [17]:  import numpy as np
          n1=np.array([1,5,3,100,4,48])
          np.std(n1)

Out[17]:  36.59424666377065
```

Median

```
In [16]:  import numpy as np
          n1=np.array([11,44,5,96,67,85])
          np.median(n1)

Out[16]:  55.5
```

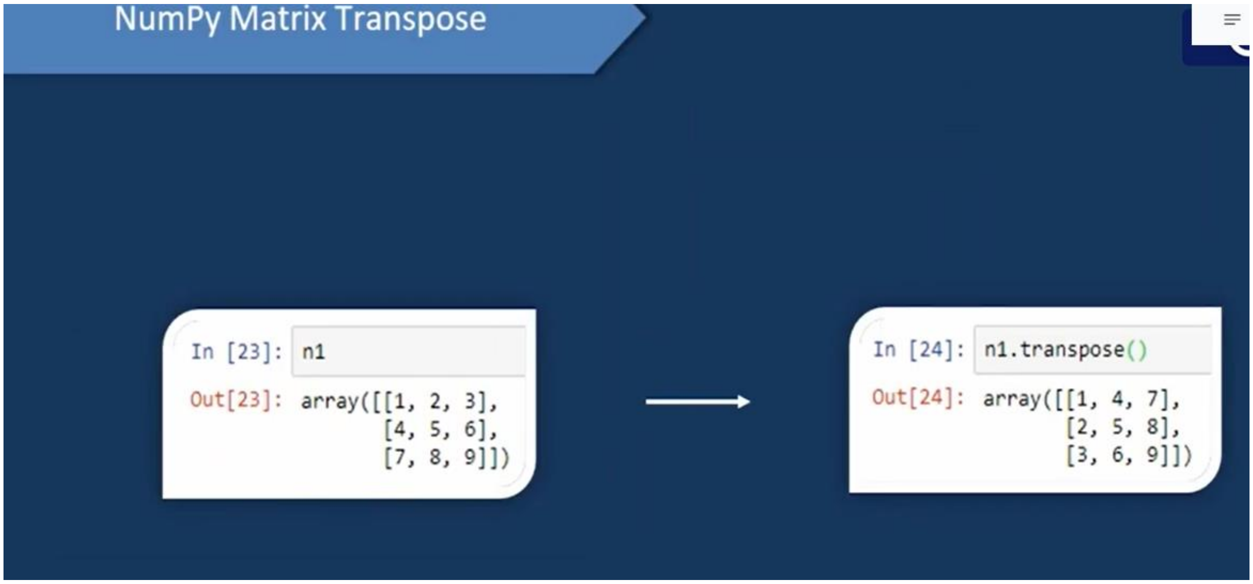


NumPy Matrix

```
In [5]:  import numpy as np
         n1 = np.array([[1,2,3],[4,5,6],[7,8,9]])
         n1

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

```
In [12]:  n1[0]

Out[12]:  array([1, 2, 3])

In [14]:  n1[1]

Out[14]:  array([4, 5, 6])
```

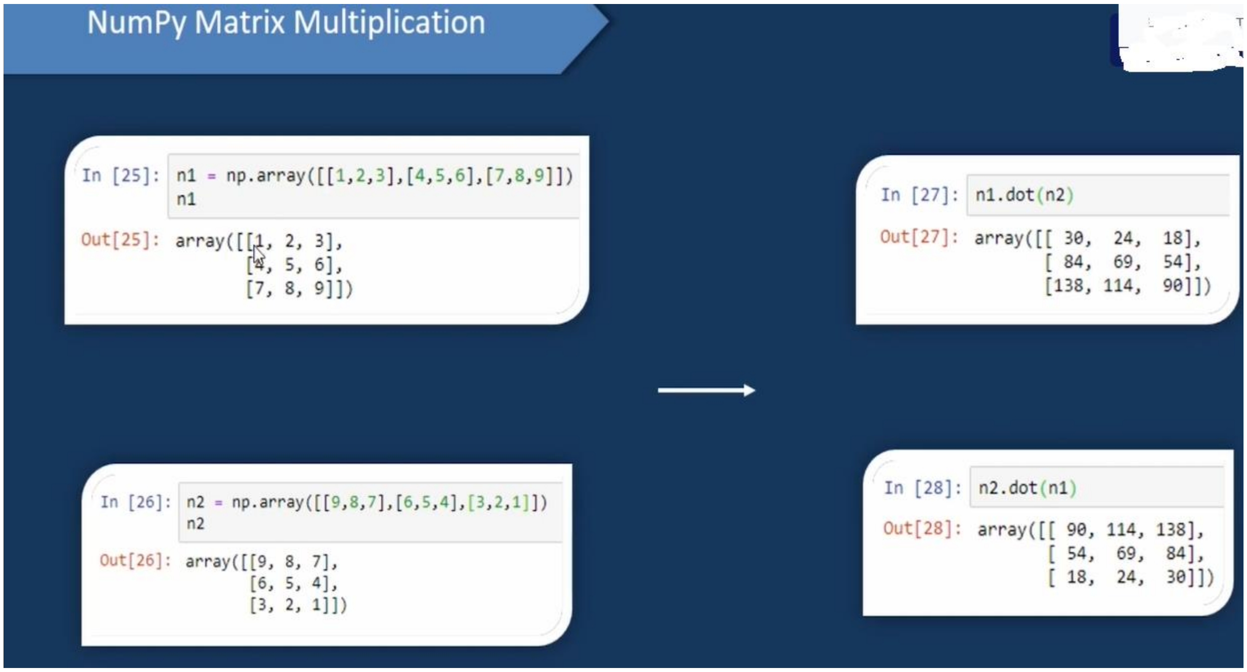
```
In [21]:  n1[:,1]

Out[21]:  array([2, 5, 8])

In [22]:  n1[:,2]

Out[22]:  array([3, 6, 9])
```



NumPy Matrix Transpose

```
In [23]:  n1

Out[23]:  array([[1, 2, 3],
                 [4, 5, 6],
                 [7, 8, 9]])
```

```
In [24]:  n1.transpose()

Out[24]:  array([[1, 4, 7],
                 [2, 5, 8],
                 [3, 6, 9]])
```



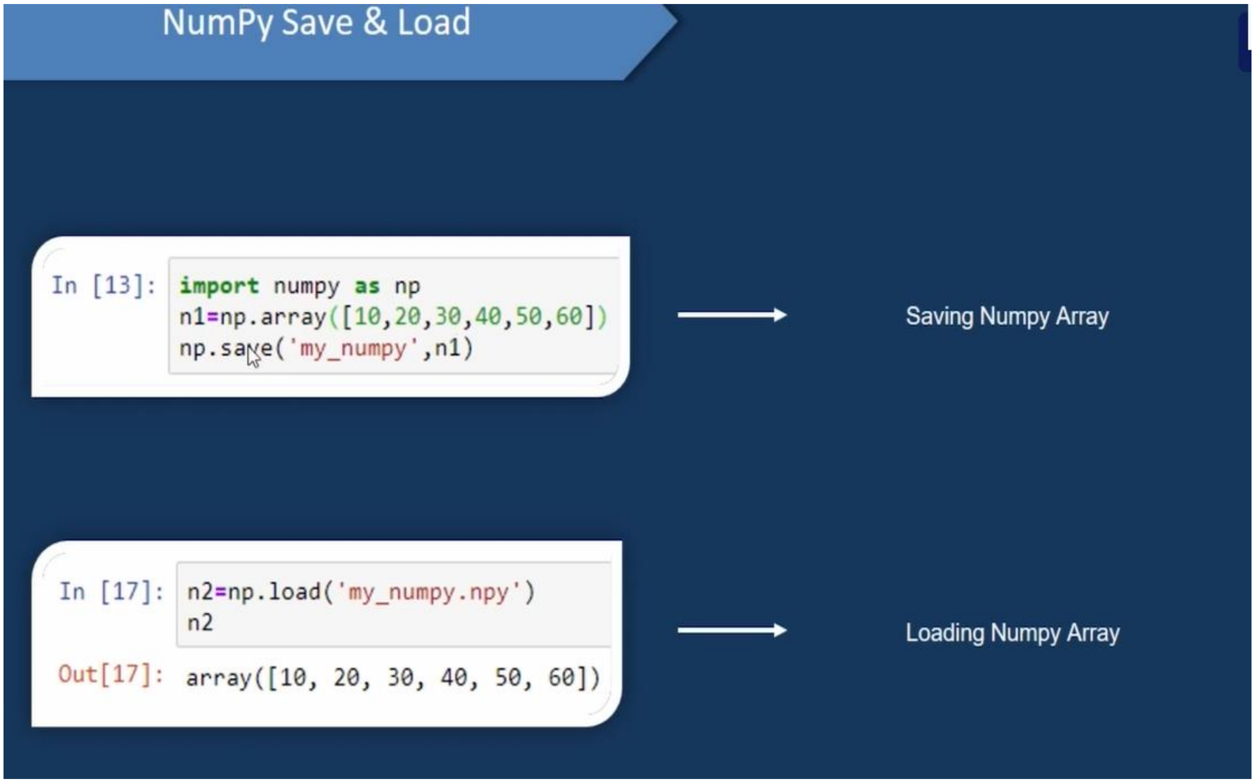
NumPy Matrix Multiplication

```
In [25]: n1 = np.array([[1,2,3],[4,5,6],[7,8,9]])
         n1

Out[25]: array([[1, 2, 3],
                [4, 5, 6],
                [7, 8, 9]])
```

```
In [26]: n2 = np.array([[9,8,7],[6,5,4],[3,2,1]])
         n2

Out[26]: array([[9, 8, 7],
                [6, 5, 4],
                [3, 2, 1]])
```

```
In [27]: n1.dot(n2)

Out[27]: array([[ 30,  24,  18],
                [ 84,  69,  54],
                [138, 114,  90]])
```

```
In [28]: n2.dot(n1)

Out[28]: array([[ 90, 114, 138],
                [ 54,  69,  84],
                [ 18,  24,  30]])
```



NumPy Save & Load

```
In [13]: import numpy as np
         n1=np.array([10,20,30,40,50,60])
         np.save('my_numpy',n1)
```

Saving Numpy Array

```
In [17]: n2=np.load('my_numpy.npy')
         n2

Out[17]: array([10, 20, 30, 40, 50, 60])
```

Loading Numpy Array