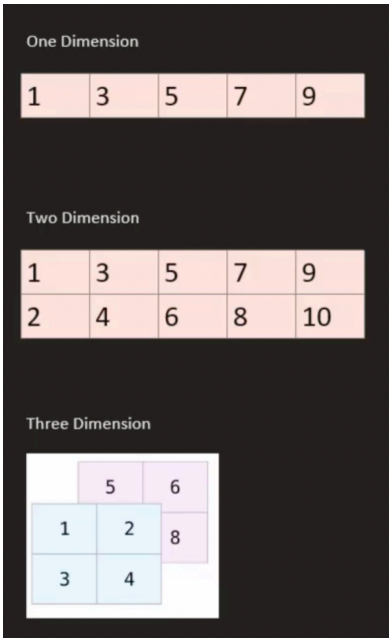
Array Creation

- We can create an array of various dimensions 1D, 2D, 3D, 5D…
- We can also take scalar values to create numpy array
- Example of different dimensions of array



- Creating a scalar array , 2D array , 3D arrays using Numpy
- We first need to import numpy to use it

```
>>> import numpy as np
>>>
>>> ar = np.array(12)
>>>
>>> ar
array(12)
>>>
```

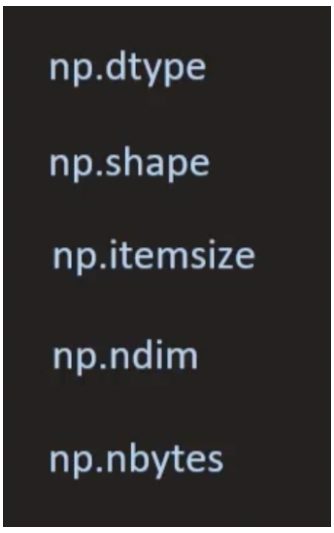
```
>>> ar1 = np.array([1,3,5,7,9])
>>> ar1
array([1, 3, 5, 7, 9])
>>> ar2 = np.array([[1,3,5,7,9],[2,4,6,8,10]])
>>> ar2
array([[ 1,  3,  5,  7,  9],
       [ 2,  4,  6,  8, 10]])
>>> ar3 = np.array([[[1,2],[3,4]],[[5,6],[7,8]]])
>>> ar3
array([[[1, 2],
        [3, 4]],

       [[5, 6],
        [7, 8]]])
>>>
```

- We can mention our own dimensions in an array by using ***ndim*** attribute

```
>>> import numpy as np
>>>
>>> ar1 = np.array([1,3,5,7,9])
>>> ar1
array([1, 3, 5, 7, 9])
>>> ar1.ndim
1
>>>
>>> ar1 = np.array([1,3,5,7,9],ndmin=3)
>>>
>>> ar1
array([[[1, 3, 5, 7, 9]]])
>>>
>>> ar1.ndim
3
>>>
```

- Numpy attributes



- Lets see them with examples

```
>>> import numpy as np
>>>
>>> ar1 = np.array([1,3,5,7,9])
>>> ar2 = np.array([[1,2,3],[4,5,6]])
>>> ar3 = np.array([[[1,1],[2,2]],[[3,3],[4,4]]])
>>> ar4 = np.array([1,2,3,4],ndmin=4)
>>>
>>> ar1.ndim
1
>>> ar3.ndim
3
>>> ar3.shape
(2, 2, 2)
>>> ar4.shape
(1, 1, 1, 4)
>>>
>>> ar1.dtype
dtype('int32')
>>> ar1.itemsize
4
>>>
```

- We can create array of float type as well

```
>>> ar5 = np.array([1.1,2.2,3.3])
>>> ar5.dtype
dtype('float64')
>>>
```