```
In [1]: print("Numpy Practice series part 1")
```

```
Numpy Practice series part 1
```

```
In [2]: import numpy as np
```

```
In [3]: arr1 = np.arange(10)
```

```
In [4]: print(arr1)
```

```
[0 1 2 3 4 5 6 7 8 9]
```

```
In [5]: arr2 = np.arange(1,11).reshape(5,6)
        print(arr2)
```

```
[[ 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 28 29 30]]
```

```
In [6]: arr3 = np.arange(32,62).reshape(5,6)
        print(arr3)
```

```
[[32 33 34 35 36 37]
 [38 39 40 41 42 43]
 [44 45 46 47 48 49]
 [50 51 52 53 54 55]
 [56 57 58 59 60 61]]
```

now checking dimension

```
In [7]: arr1.ndim
```

```
Out[7]: 1
```

```
In [8]: arr2.ndim
```

```
Out[8]: 2
```

```
In [9]: arr3.ndim
```

```
Out[9]: 2
```

```
In [10]: arr4 = np.array([[61, 62, 63, 64, 65],[66, 67, 68, 69, 70],[71,72,73,74,75],[76,77,78,79,80],[81,82,83,84,85] ])
         print(arr4)
```

```
[[61 62 63 64 65]
 [66 67 68 69 70]
 [71 72 73 74 75]
 [76 77 78 79 80]
 [81 82 83 84 85]]
```

```
In [11]: arr4.ndim
```

```
Out[11]: 2
```

type and dtype

```
In [12]: print(type(arr1))
         print(type(arr2))
         print(type(arr3))
         print(type(arr4))
```

```
<class 'numpy.ndarray'>
<class 'numpy.ndarray'>
<class 'numpy.ndarray'>
<class 'numpy.ndarray'>
```

```
In [13]: print(arr1.dtype)
         print(arr2.dtype)
         print(arr3.dtype)
         print(arr4.dtype)
```

```
int32
int32
int32
int32
```

creating 3d array ,4d array,5d array and check dtype and dim

```
In [14]: a = np.array([[[[[1,2,3],[1,2,3],[1,2,3],[1,2,3],[1,2,3]]]]])
```

```
In [15]: print(a)
         a.dtype
```

```
[[[[[1 2 3]
    [1 2 3]
    [1 2 3]
    [1 2 3]
    [1 2 3]]]]]
dtype('int32')
```

```
In [16]: print(a.ndim)
         print(type(a))
```

```
5
<class 'numpy.ndarray'>
```

```
In [17]: a = np.array([[[[1,2,3],[1,2,3],[1,2,3],[1,2,3]]]])
```

```
In [18]: print(a)
         a.dtype
```

```
[[[[1 2 3]
   [1 2 3]
   [1 2 3]
   [1 2 3]]]]
dtype('int32')
```

```
In [19]: print(a.ndim)
         print(type(a))
```

```
4
<class 'numpy.ndarray'>
```

```
In [20]: a= np.array([[[1,2,3],[1,2,3],[1,2,3]]])
```

```
In [21]: print(a)
         a.dtype
```

```
[[[1 2 3]
  [1 2 3]
  [1 2 3]]]
dtype('int32')
```

```
In [22]: print(a)
         print(a.ndim)
```

```
[[[1 2 3]
  [1 2 3]
  [1 2 3]]]
3
```

```
In [23]: print(type(a))
```

```
<class 'numpy.ndarray'>
```

```
In [ ]:
```

Numpy Practice Part 2

Concatenating Arrays

```
In [24]: arr1 = np.arange(1,13).reshape(3,4)
         print(arr1)
```

```
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]]
```

```
In [25]: arr2 = np.arange(13,25).reshape(3,4)
         print(arr2)
```

```
[[13 14 15 16]
 [17 18 19 20]
 [21 22 23 24]]
```

```
In [26]: arr3 = np.concatenate((arr1,arr2))
         print(arr3)
```

```
[[ 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 [27]: arr3
```

```
Out[27]: 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]])
```

```
In [28]: arr3 = np.concatenate((arr1,arr2),axis = 0)
         print(arr3)
```

```
[[ 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 [29]: arr3 = np.concatenate((arr1,arr2),axis=1)
         print(arr3)
```

```
[[ 1  2  3  4 13 14 15 16]
 [ 5  6  7  8 17 18 19 20]
 [ 9 10 11 12 21 22 23 24]]
```

joining array using Stack Functions

```
In [30]: arr4 = np.stack((arr1,arr2))
         print(arr4)
```

```
[[[ 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 [31]: arr4 = np.stack((arr1,arr2),axis=0)
         print(arr4)
         print(arr4.ndim)
```

```
[[[ 1  2  3  4]
  [ 5  6  7  8]
  [ 9 10 11 12]]

 [[13 14 15 16]
  [17 18 19 20]
  [21 22 23 24]]]
3
```

```
In [32]: arr4 = np.stack((arr1,arr2),axis=1)
         print(arr4)
```

```
[[[ 1  2  3  4]
  [13 14 15 16]]

 [[ 5  6  7  8]
  [17 18 19 20]]

 [[ 9 10 11 12]
  [21 22 23 24]]]
```

```
In [33]: arr4.ndim
```

```
Out[33]: 3
```

Numpy Array Indexing and slicing

```
In [34]: arr1 = np.arange(1,11)
         print(arr1)
```

```
[ 1  2  3  4  5  6  7  8  9 10]
```

```
In [37]: print((arr1[0])) # note in python indexing start from 0 not from 1
```

```
1
```

```
In [38]: print((arr1[4]))
```

```
5
```

```
In [39]: arr1.ndim
```

```
Out[39]: 1
```

```
In [45]: print(arr1[3],arr1[4])    # multiple indexing [ if anyone want to index more than one
```

```
4 5
```

```
In [46]: print(arr1[4],arr1[6],arr1[7],arr1[8])
```

```
5 7 8 9
```

```
In [47]: # lets suppose we want to add two particular index then we can do this
         print(arr1)
         print(arr1[2]+arr1[3])
```

```
[ 1  2  3  4  5  6  7  8  9 10]
7
```

```
In [49]: print(arr1[4]+arr1[5])   #similarly we can add , subtract , multiply and divide
```

```
11
```

```
In [64]: print(arr1[-2])
```

```
9
```

Indexing 2-d array and 3 d array

```
In [53]: arr2 = np.array([[1,2,3,4],[5,6,7,8]])
         print(arr2)
```

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

```
In [54]: arr2.ndim
```

```
Out[54]: 2
```

```
In [61]: arr2 = np.arange(1,9).reshape(2,4)
         print(arr2)
```

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

```
In [62]: # we can have 2 d array by this two type
```

```
In [63]: arr2
```

```
Out[63]: array([[1, 2, 3, 4],
                [5, 6, 7, 8]])
```

```
In [70]: print(arr2[0,0])  # here [1,2] 1 indicate row and 2 indicate column
```

```
1
```

```
In [71]: print(arr2[0,2])
```

```
3
```

```
In [72]: print(arr2[1,2])
```

```
7
```

indexing of 3-d array

```
In [77]: arr3 = np.arange(1,13).reshape(4,3)
         print(arr3)
```

```
[[ 1  2  3]
 [ 4  5  6]
 [ 7  8  9]
 [10 11 12]]
```

```
In [78]: arr3.ndim
```

```
Out[78]: 2
```

```
In [80]: arr3 = np.arange(13,25).reshape(4,3)
         print(arr3)
```

```
[[13 14 15]
 [16 17 18]
 [19 20 21]
 [22 23 24]]
```

```
In [85]: arr4 = np.stack((arr3,arr3))
```

```
In [86]: arr4
```

```
Out[86]: array([[[13, 14, 15],
                 [16, 17, 18],
                 [19, 20, 21],
                 [22, 23, 24]],

                [[13, 14, 15],
                 [16, 17, 18],
                 [19, 20, 21],
                 [22, 23, 24]]])
```

```
In [87]: arr4.ndim
```

```
Out[87]: 3
```

```
In [95]: print(arr4[0,3,2])
```

```
24
```

```
In [96]: print(arr4[1,3,2])
```

```
24
```

slicing

```
In [102]: print(arr1)
          arr1.ndim
```

```
[ 1  2  3  4  5  6  7  8  9 10]
Out[102]: 1
```

```
In [104]: print(arr2)
          arr2.ndim
```

```
[[1 2 3 4]
 [5 6 7 8]]
Out[104]: 2
```

```
In [107]: print(arr1[1:2])
```

```
[2]
```

```
In [109]: print(arr1[1:6]) #here 1 in exlusive and 6 in inclusive
```

```
[2 3 4 5 6]
```

```
In [111]: print(arr1[4:])
```

```
[ 5  6  7  8  9 10]
```

```
In [118]: print(arr2)
          print(arr2[:1])
```

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

```
In [125]: #print(arr2)
          print(arr2[1:5,2])
```

```
[7]
```

```
In [126]: print(arr4[0:1:4])
```

```
[[13 14 15]
 [16 17 18]
```

```
[19 20 21]
[22 23 24]]
```

In [ ]: