Assignment_3_Numpy

March 14, 2022

1 Python Chilla 10 hours video

1.1 Basics of Python

1.1.1 This file contains the basic concept of python by Ammar Bhai

1.2 Numpy Practice Assignment

```
[ ]: import numpy as np
     a = np.array([[5,5,5,4],[1,2,3,4],[1,2,3,4]])
```

```
[ ]: food = np.array(["pakora","samosa","raita"])
     food
```

```
[ ]: array(['pakora', 'samosa', 'raita'], dtype='<U6')
```

```
[ ]: food[1]
```

```
[ ]: 'samosa'
```

```
[ ]: a
```

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

```
[ ]: type(a)
```

```
[ ]: numpy.ndarray
```

```
[ ]: len(a)
```

```
[ ]: 3
```

- 3 or more dimension arrays are known as tensor just term as tensor

```
[ ]: c= np.arange(48).reshape(4,3,4)
```

```
[ ]: c
[ ]: 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, 25, 26, 27],
             [28, 29, 30, 31],
             [32, 33, 34, 35]],

            [[36, 37, 38, 39],
             [40, 41, 42, 43],
             [44, 45, 46, 47]]])
```

```
[ ]: import numpy as np
     np.zeros(6)
     np.ones(7)
```

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

```
[ ]: # Empty
     np.empty(5)
```

```
[ ]: array([2.12199579e-314, 2.34840891e+251, 4.38730294e-321, 3.79442416e-321,
            5.98181661e-154])
```

```
[ ]: # Specify
     np.arange(10)
```

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

```
[ ]: np.arange(2,20)
```

```
[ ]: array([ 2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
            19])
```

```
[ ]: # Secific Interval
     np.arange(2,20,5)
```

```
[ ]: array([ 2,  7, 12, 17])
```

```
[ ]: np.linspace(1,10,num=50)
```

```
[ ]: array([ 1.        ,  1.18367347,  1.36734694,  1.55102041,  1.73469388,
             1.91836735,  2.10204082,  2.28571429,  2.46938776,  2.65306122,
             2.83673469,  3.02040816,  3.20408163,  3.3877551 ,  3.57142857,
             3.75510204,  3.93877551,  4.12244898,  4.30612245,  4.48979592,
             4.67346939,  4.85714286,  5.04081633,  5.2244898 ,  5.40816327,
             5.59183673,  5.7755102 ,  5.95918367,  6.14285714,  6.32653061,
             6.51020408,  6.69387755,  6.87755102,  7.06122449,  7.24489796,
             7.42857143,  7.6122449 ,  7.79591837,  7.97959184,  8.16326531,
             8.34693878,  8.53061224,  8.71428571,  8.89795918,  9.08163265,
             9.26530612,  9.44897959,  9.63265306,  9.81632653, 10.        ])
```

```
[ ]: # Specify data type
     np.ones(10,dtype=np.int64)
```

```
[ ]: array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1], dtype=int64)
```

```
[ ]: a = np.array([5,5,5,4])
     a
```

```
[ ]: array([5, 5, 5, 4])
```

1.3 Array Functions

```
[ ]: a = np.array([5,3,2,4])
     a.sort()
     a
```

```
[ ]: array([2, 3, 4, 5])
```

```
[ ]: b=np.array([2,3,45,6])
     c=np.concatenate((a,b))
     c
```

```
[ ]: array([ 2,  3,  4,  5,  2,  3, 45,  6])
```

```
[ ]: a = np.array([[5,5,5,4],[5,5,5,4],[5,5,5,4]])
     b=np.array([[2,3,45,6],[5,5,5,4],[5,5,5,4]])
     c=np.concatenate((a,b))
     c
```

```
[ ]: array([[ 5,  5,  5,  4],
            [ 5,  5,  5,  4],
            [ 5,  5,  5,  4],
            [ 2,  3, 45,  6],
            [ 5,  5,  5,  4],
            [ 5,  5,  5,  4]])
```

```
[ ]: a.ndim
```

```
[ ]: 2
[ ]: d=np.array([[[1,2,3],[1,6,7],[5,6,7]]])
```

```
[ ]: d
```

```
[ ]: array([[[1, 2, 3],
             [1, 6, 7],
             [5, 6, 7]]])
```

```
[ ]: d.ndim
```

```
[ ]: 3
```

```
[ ]: d.size
```

```
[ ]: 9
```

```
[ ]: d.shape
```

```
[ ]: (1, 3, 3)
```

```
[ ]: d.reshape(1,9)
```

```
[ ]: array([[1, 2, 3, 1, 6, 7, 5, 6, 7]])
```

```
[ ]: # Reshape
     np.reshape(d,newshape=(1,9),order='c')
```

```
[ ]: array([[1, 2, 3, 1, 6, 7, 5, 6, 7]])
```

```
[ ]: # convert 1D into 2D
     a=np.array([1,2,3,4])
```

```
[ ]: a.reshape(2,2)
```

```
[ ]: array([[1, 2],
            [3, 4]])
```

```
[ ]: a=np.array([1,2,3,4,3,4,5,6])
     a[2:6]
```

```
[ ]: array([3, 4, 3, 4])
```

```
[ ]: a*3
```

```
[ ]: array([ 3,  6,  9, 12,  9, 12, 15, 18])
```

```
[ ]: a+6
```

```
[ ]: array([ 7,  8,  9, 10,  9, 10, 11, 12])
[ ]: a.mean()
```

```
[ ]: 3.5
```

```
[ ]: a.max()
```

```
[ ]: 6
```

```
[ ]: a.min()
```

```
[ ]: 1
```

```
[ ]: a = np.array([[1 , 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]])
     print(a[a < 5])
```

```
[1 2 3 4]
```

```
[ ]: print(a[a > 5])
```

```
[ 6  7  8  9 10 11 12]
```

```
[ ]: # Counting unique Numbers
     a = np.array([11, 11, 12, 13, 14, 15, 16, 17, 12, 13, 11, 14, 18, 19, 20])
     print(np.unique(a))
```

```
[11 12 13 14 15 16 17 18 19 20]
```

```
[ ]: a.transpose()
     a
```

```
[ ]: array([11, 11, 12, 13, 14, 15, 16, 17, 12, 13, 11, 14, 18, 19, 20])
```

```
[ ]: d.transpose()
```

```
[ ]: array([[[1],
             [1],
             [5]],

            [[2],
             [6],
             [6]],

            [[3],
             [7],
             [7]]])
```

```
[ ]: np.flip(a)
```

```
[ ]: array([20, 19, 18, 14, 11, 13, 12, 17, 16, 15, 14, 13, 12, 11, 11])
[ ]: np.flip(d)
```

```
[ ]: array([[[7, 6, 5],
             [7, 6, 1],
             [3, 2, 1]]])
```

```
[ ]:
```