Numpy Operation

November 4, 2019

1 Operations On Numpy Array

```
In [1]: import numpy

In [2]: Numpy_Array = numpy.arange(25,96)

In [3]: Numpy_Array

Out[3]: array([25, 26, 27, 28, 29, 30, 31, 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, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75,
               76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92,
               93, 94, 95])
```

Indexing and Slicing on an Numpy Array

```
In [7]: Numpy_Array.shape

Out[7]: (71,)

In [8]: Numpy_Array[26]

Out[8]: 51

In [9]: Numpy_Array[0]

Out[9]: 25

In [11]: Numpy_Array[70]

Out[11]: 95

In [12]: Numpy_Array[5:71]

Out[12]: array([30, 31, 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, 62, 63,
                64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80,
                81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95])
```

```
In [13]: Numpy_Array[0:5]

Out[13]: array([25, 26, 27, 28, 29])

In [14]: Numpy_Array[:60]

Out[14]: array([25, 26, 27, 28, 29, 30, 31, 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, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75,
                76, 77, 78, 79, 80, 81, 82, 83, 84])

In [16]: Numpy_Array[-1]

Out[16]: 95

In [18]: Numpy_Array[-9]

Out[18]: 87

In [19]: Numpy_Array[:-1]

Out[19]: array([25, 26, 27, 28, 29, 30, 31, 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, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75,
                76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92,
                93, 94])

In [20]: Numpy_Array[-1:25]

Out[20]: array([], dtype=int64)
```

Operations on Numpy Arrays

```
In [21]: Numpy_Array>50

Out[21]: array([False, False, False, False, False, False, False, False, False,
                False, False, False, False, False, False, False, False, False,
                False, False, False, False, False, False, False, False,  True,
                 True,  True,  True,  True,  True,  True,  True,  True,  True,
                 True,  True,  True,  True,  True,  True,  True,  True,  True,
                 True,  True,  True,  True,  True,  True,  True,  True,  True,
                 True,  True,  True,  True,  True,  True,  True,  True,  True,
                 True,  True,  True,  True,  True,  True,  True,  True])

In [22]: sum(Numpy_Array>40)

Out[22]: 55

In [23]: Numpy_Array+85
```

```
Out[23]: array([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])

In [24]: Numpy_Array*12

Out[24]: array([ 300,  312,  324,  336,  348,  360,  372,  384,  396,  408,  420,
                 432,  444,  456,  468,  480,  492,  504,  516,  528,  540,  552,
                 564,  576,  588,  600,  612,  624,  636,  648,  660,  672,  684,
                 696,  708,  720,  732,  744,  756,  768,  780,  792,  804,  816,
                 828,  840,  852,  864,  876,  888,  900,  912,  924,  936,  948,
                 960,  972,  984,  996, 1008, 1020, 1032, 1044, 1056, 1068, 1080,
                1092, 1104, 1116, 1128, 1140])

In [25]: Numpy_Array/23

Out[25]: array([1.08695652, 1.13043478, 1.17391304, 1.2173913 , 1.26086957,
                1.30434783, 1.34782609, 1.39130435, 1.43478261, 1.47826087,
                1.52173913, 1.56521739, 1.60869565, 1.65217391, 1.69565217,
                1.73913043, 1.7826087 , 1.82608696, 1.86956522, 1.91304348,
                1.95652174, 2.        , 2.04347826, 2.08695652, 2.13043478,
                2.17391304, 2.2173913 , 2.26086957, 2.30434783, 2.34782609,
                2.39130435, 2.43478261, 2.47826087, 2.52173913, 2.56521739,
                2.60869565, 2.65217391, 2.69565217, 2.73913043, 2.7826087 ,
                2.82608696, 2.86956522, 2.91304348, 2.95652174, 3.        ,
                3.04347826, 3.08695652, 3.13043478, 3.17391304, 3.2173913 ,
                3.26086957, 3.30434783, 3.34782609, 3.39130435, 3.43478261,
                3.47826087, 3.52173913, 3.56521739, 3.60869565, 3.65217391,
                3.69565217, 3.73913043, 3.7826087 , 3.82608696, 3.86956522,
                3.91304348, 3.95652174, 4.        , 4.04347826, 4.08695652,
                4.13043478])
```

Indexing and Slicing in an Multidimensional_Numpy_Array

```
In [30]: Multidimensional_Numpy_Array = numpy.random.randint(12 , 58 , size=(4,4))

In [31]: Multidimensional_Numpy_Array

Out[31]: array([[21, 15, 27, 15],
                [18, 12, 13, 15],
                [32, 47, 46, 50],
                [19, 41, 25, 17]])

In [32]: Multidimensional_Numpy_Array[0]

Out[32]: array([21, 15, 27, 15])
```

```
In [36]: Multidimensional_Numpy_Array[1,3]

Out[36]: 15

In [43]: Multidimensional_Numpy_Array[1:3 , 3]

Out[43]: array([15, 50])

In [38]: Multidimensional_Numpy_Array[: , 3]

Out[38]: array([15, 15, 50, 17])

In [39]: Multidimensional_Numpy_Array[:-1 , 2]

Out[39]: array([27, 13, 46])

In [40]: Multidimensional_Numpy_Array[-1: , 3]

Out[40]: array([17])

In [41]: Multidimensional_Numpy_Array[2 , 1:]

Out[41]: array([47, 46, 50])

In [42]: Multidimensional_Numpy_Array[3,:]

Out[42]: array([19, 41, 25, 17])

In [44]: Multidimensional_Numpy_Array[2 ,1:4 ]

Out[44]: array([47, 46, 50])

In [45]: Multidimensional_Numpy_Array[2:4 , :2]

Out[45]: array([[32, 47],
                [19, 41]])
```

Mathamatical Operations on Numpy_Array

```
In [46]: numpy.cos(Numpy_Array)

Out[46]: array([ 0.99120281,  0.64691932, -0.29213881, -0.96260587, -0.74805753,
                 0.15425145,  0.91474236,  0.83422336, -0.01327675, -0.84857027,
                -0.90369221, -0.12796369,  0.76541405,  0.95507364,  0.26664293,
                -0.66693806, -0.98733928, -0.39998531,  0.5551133 ,  0.99984331,
                 0.52532199, -0.43217794, -0.99233547, -0.64014434,  0.30059254,
                 0.96496603,  0.7421542 , -0.16299078, -0.91828279, -0.82930983,
                 0.02212676,  0.85322011,  0.89986683,  0.11918014, -0.77108022,
                -0.95241298, -0.25810164,  0.67350716,  0.98589658,  0.39185723,
                -0.56245385, -0.99964746, -0.5177698 ,  0.44014302,  0.99339038,
                 0.6333192 , -0.30902273, -0.96725059, -0.73619272,  0.17171734,
                 0.92175127,  0.82433133, -0.03097503, -0.85780309, -0.89597095,
                -0.11038724,  0.77668598,  0.9496777 ,  0.24954012, -0.6800235 ,
                -0.98437664, -0.38369844,  0.56975033,  0.99937328,  0.51017704,
                -0.44807362, -0.99436746, -0.62644445,  0.3174287 ,  0.96945937,
                 0.73017356])
```

```
In [47]: numpy.cosh(Numpy_Array)

Out[47]: array([3.60024497e+10, 9.78648047e+10, 2.66024120e+11, 7.23128532e+11,
                1.96566715e+12, 5.34323729e+12, 1.45244248e+13, 3.94814801e+13,
                1.07321790e+14, 2.91730871e+14, 7.93006726e+14, 2.15561577e+15,
                5.85957119e+15, 1.59279659e+16, 4.32967002e+16, 1.17692633e+17,
                3.19921747e+17, 8.69637471e+17, 2.36391973e+18, 6.42580006e+18,
                1.74671355e+19, 4.74805971e+19, 1.29065644e+20, 3.50836796e+20,
                9.53673286e+20, 2.59235276e+21, 7.04674541e+21, 1.91550400e+22,
                5.20687972e+22, 1.41537665e+23, 3.84739263e+23, 1.04582975e+24,
                2.84286000e+24, 7.72769468e+24, 2.10060520e+25, 5.71003695e+25,
                1.55214897e+26, 4.21917833e+26, 1.14689158e+27, 3.11757454e+27,
                8.47444622e+27, 2.30359332e+28, 6.26181585e+28, 1.70213802e+29,
                4.62689086e+29, 1.25771934e+30, 3.41883561e+30, 9.29335873e+30,
                2.52619682e+31, 6.86691490e+31, 1.86662100e+32, 5.07400194e+32,
                1.37925673e+33, 3.74920850e+33, 1.01914053e+34, 2.77031119e+34,
                7.53048657e+34, 2.04699848e+35, 5.56431877e+35, 1.51253866e+36,
                4.11150636e+36, 1.11762330e+37, 3.03801511e+37, 8.25818127e+37,
                2.24480641e+38, 6.10201647e+38, 1.65870005e+39, 4.50881420e+39,
                1.22562277e+40, 3.33158811e+40, 9.05619541e+40])

In [48]: numpy.tan(Numpy_Array)

Out[48]: array([-1.33526407e-01,  1.17875355e+00, -3.27370380e+00, -2.81429605e-01,
                 8.87142844e-01, -6.40533120e+00, -4.41695568e-01,  6.61006041e-01,
                -7.53130148e+01, -6.23498963e-01,  4.73814720e-01,  7.75047091e+00,
                -8.40771255e-01,  3.10309661e-01,  3.61455441e+00, -1.11721493e+00,
                 1.60656699e-01,  2.29138799e+00, -1.49838734e+00,  1.77046993e-02,
                 1.61977519e+00, -2.08661353e+00, -1.24527568e-01,  1.20012724e+00,
                -3.17290855e+00, -2.71900612e-01,  9.03086149e-01, -6.05327238e+00,
                -4.31158197e-01,  6.73800101e-01, -4.51830879e+01, -6.11273688e-01,
                 4.84699227e-01,  8.33085685e+00, -8.25774009e-01,  3.20040389e-01,
                 3.74316794e+00, -1.09750978e+00,  1.69749752e-01,  2.34786031e+00,
                -1.47003826e+00,  2.65605178e-02,  1.65231726e+00, -2.04008160e+00,
                -1.15548546e-01,  1.22195992e+00, -3.07762040e+00, -2.62417378e-01,
                 9.19286404e-01, -5.73702254e+00, -4.20700951e-01,  6.86747689e-01,
                -3.22685758e+01, -5.99179998e-01,  4.95677533e-01,  9.00365495e+00,
                -8.10994416e-01,  3.29826407e-01,  3.88059631e+00, -1.07818381e+00,
                 1.78870172e-01,  2.40672971e+00, -1.44241747e+00,  3.54205013e-02,
                 1.68582537e+00, -1.99520041e+00, -1.06587872e-01,  1.24427006e+00,
                -2.98738626e+00, -2.52978097e-01,  9.35752472e-01])

In [49]: numpy.sum(Numpy_Array)

Out[49]: 4260

In [53]: numpy.mean(Numpy_Array)

Out[53]: 60.0
```

```
In [54]: numpy.median(Numpy_Array)

Out[54]: 60.0

In [56]: numpy.mod(Numpy_Array , 5)

Out[56]: array([0, 1, 2, 3, 4, 0, 1, 2, 3, 4, 0, 1, 2, 3, 4, 0, 1, 2, 3, 4, 0, 1,
                2, 3, 4, 0, 1, 2, 3, 4, 0, 1, 2, 3, 4, 0, 1, 2, 3, 4, 0, 1, 2, 3,
                4, 0, 1, 2, 3, 4, 0, 1, 2, 3, 4, 0, 1, 2, 3, 4, 0, 1, 2, 3, 4, 0,
                1, 2, 3, 4, 0])

In [ ]:
```