numpy Math

In [4]:

```
import numpy as np

import os
x = np.arange(12).reshape(4, 3)
print("Original array:")
print(x)
header = 'col1 col2 col3'
np.savetxt('temp.txt', x, fmt="%d", header=header)
print("After loading, content of the text file:")
result = np.loadtxt('temp.txt')
print(result)
```

```
Original array:
[[ 0  1  2]
 [ 3  4  5]
 [ 6  7  8]
 [ 9 10 11]]
After loading, content of the text file:
[[ 0.  1.  2.]
 [ 3.  4.  5.]
 [ 6.  7.  8.]
 [ 9. 10. 11.]]
```

In [11]:

```
f = np.loadtxt("narray.txt")
```

In [12]:

```
f
```

Out[12]:

```
array([[2., 4., 6., 7.],
       [1., 2., 3., 5.],
       [3., 2., 7., 8.],
       [1., 5., 6., 4.]])
```

In [13]:

```
f = np.loadtxt("narray.txt", dtype = "int")
```

In [14]:

```
f
```

Out[14]:

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

In [15]:

```
f.dtype
```

Out[15]:

```
dtype('int64')
```

2 NUMPY Maths

2.1 Adding, subtracting, multiplying, transposing arrays

In [16]:

```
# Generating 5*5 size array using random function for integer datatype

import numpy as np
x =  np.random.randint(100, size = (5,5))
y =  np.random.randint(100, size = (5,5))
```

In [17]:

```
x
```

Out[17]:

```
array([[ 0,  0, 32, 85, 42],
       [91, 44, 39, 88, 20],
       [32, 47, 98, 44, 38],
       [83, 97, 24, 99, 99],
       [94, 67, 33, 12, 40]])
```

In [18]:

```
y
```

Out[18]:

```
array([[15, 64, 18, 21, 50],
       [81, 66, 94,  4, 44],
       [69,  9,  3, 50,  4],
       [32, 58, 49, 45, 30],
       [16, 84, 22,  1, 79]])
```

In [19]:

```
## Add two matrices x+y or np.add(x, y)
x+ y
```

Out[19]:

```
array([[ 15,  64,  50, 106,  92],
       [172, 110, 133,  92,  64],
       [101,  56, 101,  94,  42],
       [115, 155,  73, 144, 129],
       [110, 151,  55,  13, 119]])
```

In [21]:

```
np.add(x, y)
```

Out[21]:

```
array([[ 15,  64,  50, 106,  92],
       [172, 110, 133,  92,  64],
       [101,  56, 101,  94,  42],
       [115, 155,  73, 144, 129],
       [110, 151,  55,  13, 119]])
```

In [22]:

```
# Matrix Transpose

x.T
```

Out[22]:

```
array([[ 0, 91, 32, 83, 94],
       [ 0, 44, 47, 97, 67],
       [32, 39, 98, 24, 33],
       [85, 88, 44, 99, 12],
       [42, 20, 38, 99, 40]])
```

3. Calculating column sums and row sums

In [23]:

```
x = np.random.randint(10, size = (4,4))
```

In [24]:

```
x
```

Out[24]:

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

In [25]:

```
print (x.sum(), x.mean(), x.std())
```

```
69 4.3125 2.567069097239106
```

In [26]:

```
np.sum(x, axis = 0)
```

Out[26]:

```
array([12, 25, 13, 19])
```

In [27]:

```
x+ [10,14,12,12]
```

Out[27]:

```
array([[11, 17, 13, 16],
       [13, 21, 21, 17],
       [14, 22, 15, 19],
       [14, 21, 12, 15]])
```

In [28]:

```
l = np.array([[10,20], [20,30]])
```

In [29]:

```
l+10
```

Out[29]:

```
array([[20, 30],
       [30, 40]])
```

In [30]:

```
x
```

Out[30]:

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

In [31]:

```
np.std(x)
```

Out[31]:

```
2.567069097239106
```

Advance Matrix Operations

In [32]:

```
from numpy import linalg
import numpy as np
```

6 Solving a set of linear equations

2x + 2y+3z = 5 3x + y +4z = 7 4x + 3y =10

In [33]:

```
a = np.array([[2,2,3], [3,1,4],[4,3,0]])
b = np.array([5,7,10])
x = np.linalg.solve(a, b)
x
```

Out[33]:

```
array([ 2.30434783,  0.26086957, -0.04347826])
```

7 Matrix Inversion

In [34]:

```
a = np.array([[1, 2], [3, 4]])
np.linalg.inv( a )
```

Out[34]:

```
array([[-2. ,  1. ],
       [ 1.5, -0.5]])
```

8 Calculating an eigen value and vector for a matrix

In [35]:

```
m1 = np.diag((1, 2, 3))
m1
```

Out[35]:

```
array([[1, 0, 0],
       [0, 2, 0],
       [0, 0, 3]])
```

In [36]:

```
 eigval, eigvec = np.linalg.eig( m1 )
```

In [37]:

```
eigval
```

Out[37]:

```
array([1., 2., 3.])
```

In [38]:

```
eigvec
```

Out[38]:

```
array([[1., 0., 0.],
       [0., 1., 0.],
       [0., 0., 1.]])
```

In [ ]: