Matrix Algebra in Python

```
print("week 04")

print("Matrix Algebra in Python")
```

```
week 04
Matrix Algebra in Python
```

"numpy" is a library, which a powerful tool for numerical simulation.

```
# import numpy library
import numpy as np
```

```
# Create a matrix using numpy
A = np.array([[1, 2], [3, 4]])
print("Matrix A= \n",A)
```

```
Matrix A=
 [[1 2]
 [3 4]]
```

```
# Create another matrix B(say)
B= np.array([[2, 1],[0, 1]])
print("Matrix B= \n",B)
```

```
Matrix B=
 [[2 1]
 [0 1]]
```

Addition of the Matrices

Now, we want to add A and B Matrix in the next cell.

```
# Say, C = A + B
C = A + B
print("Matrix C=A+B = \n",C)
```

```
Matrix C=A+B =
 [[3 3]
 [3 5]]
```

Subtract of the matrices

```
# Say, D = A - B
D = A - B
print("Matrix D=A-B = \n",D)
```

```
Matrix D=A-B =
 [[-1  1]
 [ 3  3]]
```

Multiplication of the matrices

```
# Why we can't do like this way:
# Say, M = A * B
M = A * B
print("Matrix M=A*B = \n", M)
```

```
Matrix M=A*B =
 [[2 2]
 [0 4]]
```

```
# We have to use dot for the matrix multiplication
M_2 = np.dot(A, B)
print("Matrix M_2=A.B = \n", M_2)
```

```
Matrix M_2=A.B = 
 [[2 3]
 [6 7]]
```

Transpose and Inverse Matrix

```
# Create a transpose of matrix A
A_T = A.T
print("The traspose of matrix A is \n", A_T)
```

```
The traspose of matrix A is 
 [[1 3]
 [2 4]]
```

```
# Create inverse matrix of A
A_inv = np.linalg.inv(A)
print("The inverse of matrix A is \n", A_inv)
```

```
The inverse of matrix A is 
 [[-2.   1. ]
 [ 1.5 -0.5]]
```

Determinant and Trace of a Matrix

```
# Determinant of a Matrix A
A_det = np.linalg.det(A)
print("The determinant of a matrix A is \n", int(A_det))
```

```
The determinant of a matrix A is 
 -2
```

```
# Trace of a matrix A
A_trace = np.trace(A)
print("The trace of matrix A is \n", A_trace)
```

```
The trace of matrix A is 
 5
```

Eigenvalues and Eigenvectors

```
# We want see thethe eigenvalues and eigenvectors of a matrix A
eigenvalues, eigenvectors = np.linalg.eig(A)

print("The eigenvalues of a matrix A: ", eigenvalues)
print("The eigenvectors of a matrix A: ", eigenvectors)
```

```
The eigenvalues of a matrix A:  [-0.37228132  5.37228132]
The eigenvectors of a matrix A:  [[-0.82456484 -0.41597356]
 [ 0.56576746 -0.90937671]]
```

Rank of a matrix

```
A_rank = np.linalg.matrix_rank(A)

print("The rank of a matrix A_rank: ", A_rank)
```

```
The rank of a matrix A_rank:  2
```

```
# Create a matrix using numpy
Y = np.array([[1, 2], [3, 4], [5, 6]])
print("Matrix Y= \n", Y)
```

```
Matrix Y=
 [[1 2]
 [3 4]
 [5 6]]
```

```
Y_rank = np.linalg.matrix_rank(Y)

print("The rank of a matrix Y_rank: ", Y_rank)
```

```
The rank of a matrix Y_rank:  2
```

LU Decomposition

```
from scipy.linalg import lu
P, L, U = lu(A)

print("P: \n", P)
```

```
P:
 [[0. 1.]
 [1. 0.]]
```

```
print("L: \n", L)
```

```
L:
 [[1.         0.        ]
 [0.33333333 1.        ]]
```

```
print("U: \n", U)
```

```
U:
 [[3.         4.        ]
 [0.         0.66666667]]
```

Singular value decomposition

```
U, s, V = np.linalg.svd(A)

print("U :", U)
```

```
U : [[-0.40455358 -0.9145143 ]
 [-0.9145143   0.40455358]]
```

```
print("s :", s)
```

```
s : [5.4649857  0.36596619]
```

```
print("V :", V)
```

```
V : [[-0.57604844 -0.81741556]
 [ 0.81741556 -0.57604844]]
```

```
# ********************************************************************************************************************** #
```