: Python and Linear Algebra :

: Assignment 2 :

Matrix :

A matrix is a two-dimensional array of scalars with one or more columns and one or more rows.

The notation for a matrix is often an uppercase letter, such as A, and entries are referred to by their two-dimensional subscript of row (i) and column (j), such as a i,j . For example:

A = (a 1,1 a 1,2
a 2,1 a 2,2
a 3,1 a 3,2)

Matrix in Python using a two-dimensional NumPy array.

```
# create matrix
from numpy import array
A = array([[1, 2, ?], [4, ?, 6]])
print(?)
```

Matrix Addition :

Two matrices with the same dimensions can be added together to create a new third matrix.
The **scalar** elements in the resulting matrix are calculated as the addition of the elements in each of the matrices being added.

```
# add matrices
from numpy import array
A = array([[1, 2, 3], [4, 5, 6]])
print(?)
B = array([[1, 2, 3], [4, 5, 6]])
print(?)
C = A + B
print(?)
```

Matrix Dot Product :

The rule for matrix multiplication is as follows:

The number of columns (n) in the first matrix (A) must equal the number of rows (m) in the second matrix (B).

Example, matrix A has the dimensions m rows and n columns and matrix B has the dimensions n and k. The n columns in A and n rows b are equal. The result is a new matrix with m rows and k columns.

$$C(m, k) = A(m, n) \times B(n, k)$$

```
# matrix dot product
from numpy import array
A = array([[1, 2], [3, 4], [5, 6]])
print(?)
B = array([[1, 2], [3, 4]])
print(B)
C = A.dot(B)
print(?)
```

Task 1 : For this lesson, you must implement more matrix arithmetic operations such as subtraction, division, vector- matrix multiplication

: Assignment 3 :

Matrix Types :

Transpose -

A defined matrix can be transposed, which creates a new matrix with the number of columns and rows flipped. This is denoted by the superscript T next to the matrix.

Task 2 : **Write the equation of transpose**.

```
# transpose matrix
```

```
from numpy import array
A = array([[1, 2], [3, 4], [5, 6]])
print(A)
C = A.T
print(?)
```

Inversion -

The operation of inverting a matrix is indicated by a -1 superscript next to the matrix;

Task 2 : **Write the equation of inversion**.

The result of the operation is referred to as the inverse of the original matrix; for example, B is the inverse of A.

Not all matrices are **invertible**.

```
# invert matrix
from numpy import array
from numpy.linalg import inv
# define matrix
A = array([[1.0, 2.0], [3.0, 4.0]])
print(A)
# invert matrix
B = inv(A)
print(?)
```

Square Matrix -

A square matrix is a matrix where the number of rows (n) equals the number of columns (m).

The square matrix is contrasted with the rectangular matrix where the number of rows and columns are not equal.

Task 3 : **Write the equation of the square matrix**. Give an example.

Symmetric Matrix -

A symmetric matrix is a type of square matrix where the top-right triangle is the same as the bottom-left triangle. To be symmetric, the axis of symmetry is always the main diagonal of the matrix, from the top left to the bottom right. A symmetric matrix is always square and equal to its own transpose.

Task 4 : **Write the equation of the symmetric matrix**. Give an example.

Triangular Matrix -

A triangular matrix is a type of square matrix that has all values in the upper-right or lower-left of the matrix with the remaining elements filled with zero values. A triangular matrix with values only above the main diagonal is called an upper triangular matrix. Whereas, a triangular matrix with values only below the main diagonal is called a lower triangular matrix.

Task 5 : **Write the equation of the symmetric matrix**. Give an example.

Diagonal Matrix -

A diagonal matrix is one where values outside of the main diagonal have a zero value, where the main diagonal is taken from the top left of the matrix to the bottom right. A diagonal matrix is often denoted with the variable D and may be represented as a full matrix or as a vector of values on the main diagonal.

Task 6 : **Write the equation of the diagonal matrix**. Give an example.