In [2]:
```
import numpy as np
```

In [2]:
```
# Create an array:
#a = np.array([1,2,3], dtype='int16')
a = np.array([1,2,3])
print(a)
```

```
[1 2 3]
```

In [4]:
```
# 2 dimensional array: 2 rows and 3 columns:
b = np.array([[1,2,3],[4,5,6]])
print(b)
```

```
[[1 2 3]
 [4 5 6]]
```

In [5]:
```
# Get dimension of numpy array:
# Just 1 row and 1 column
a.ndim
```

Out[5]:
```
1
```

In [6]:
```
# Get dimension of numpy array:
# 2 dimensions: 2 rows and 3 columns
b.ndim
```

Out[6]:
```
2
```

In [7]:
```
# Check how many memory that an numpy array take up:
a.dtype
```

Out[7]:
```
dtype('int64')
```

In [ ]:
```

```

In [ ]:
```
# Selecting element in a matrix:
```

In [3]:
```
# Create a vector as a row:
vector_row = np.array([1,2,3,4,5,6])
print('Vector row:\n', vector_row)
```

```
Vector row:
 [1 2 3 4 5 6]
```

In [5]:
```
# Create a matrix:
matrix = np.array([[1,2,3],[4,5,6],[7,8,9]])
print('Matrix:\n',matrix)
```

```
Matrix:
 [[1 2 3]
 [4 5 6]
 [7 8 9]]
```

In [6]:
```
# Select 1st element of row vector:
print('1st element of the row vector is',vector_row[0])
```

```
1st element of the row vector is 1
```

In [7]:
```
# Select 2nd element of row vector:
print('2nd element of the row vector is',vector_row[1])
```

```
2nd element of the row vector is 2
```

In [8]:
```
# Select the 3rd element of row vector:
print('3rd element of row vector is',vector_row[2])
```

```
3rd element of row vector is 3
```

In [14]:
```
# Select all element of a vector:
print('All element of the row vector:\n',vector_row[:])
```

```
All element of the row vector:
 [1 2 3 4 5 6]
```

In [15]:
```
# Select first 3 elements of a vector:
print('The first 3 elements of the vector are:\n',vector_row[:3])
```

```
The first 3 elements of the vector are:
 [1 2 3]
```

In [16]:
```
# Select last element of a vector:
print('The last element of the vector is:\n',vector_row[-1])
```

```
The last element of the vector is:
 6
```

In [18]:
```
# Select everything after 2nd element:
print('The last 3 element of the vector are:\n',vector_row[3:])
```

```
The last 3 element of the vector are:
 [4 5 6]
```

In [ ]:
```
Operation with matrix
```

In [10]:
```
# Select 1st row and 1st column of a matrix:
print('1st row and 1st column of the matrix is:',matrix[0,0])
```

```
1st row and 1st column of the matrix is: 1
```

In [11]:
```
# Select 2nd row and 2nd column of a matrix:
print('2nd row and 2nd column of the matrix is:',matrix[1,1])
```

```
2nd row and 2nd column of the matrix is: 5
```

In [12]:
```
# Select 3rd row and 3rd column of a matrix:
print('3rd row and 3rd column of the matrix is:',matrix[2,2])
```

```
3rd row and 3rd column of the matrix is: 9
```

In [13]:
```
# Select 1st row and 2nd column of a matrix:
print('1st row and 2nd column of the matrix is:',matrix[0,1])
```

```
1st row and 2nd column of the matrix is: 2
```

In [19]:
```
# Select all element of a matrix:
print('All element of the matrix:\n',matrix[:,:])
```

```
All element of the matrix:
 [[1 2 3]
 [4 5 6]
 [7 8 9]]
```

In [23]:
```
# Select all column and first 2 rows of a matrix:
print('All column and 2nd row of the matrix:\n',matrix[:2,:])
```

```
All column and 2nd row of the matrix:
 [[1 2 3]
 [4 5 6]]
```

In [26]:
```
# Select 2nd column of a matrix:
print('2nd column of the matrix:\n',matrix[:,1])
```

```
2nd column of the matrix:
 [2 5 8]
```

In [29]:
```
# Select last column:
print('1st way to show the last column of the matrix:\n',matrix[:,-1])
print('2nd way to show the last column of the matrix:\n',matrix[:,2])
```

```
1st way to show the last column of the matrix:
 [3 6 9]
2nd way to show the last column of the matrix:
 [3 6 9]
```

In [ ]:
```

```

In [ ]:
```
Describing a matrix:
```

In [3]:
```
# Create a matrix:
matrix_1 = np.array([[1,2,3],[3,4,5],[5,6,8]])
print(matrix_1)
```

```
[[1 2 3]
 [3 4 5]
 [5 6 8]]
```

In [32]:
```
# Numebr of row and column: .shape function
# This matrix has 3 rows and 3 columns
print('number of row and column of the matrix:\n',matrix_1.shape)
```

```
number of row and column of the matrix:
 (3, 3)
```

In [34]:
```
# Number of element in a matrix(size - row*column): .size function
print('Size of the matrix:',matrix_1.size)
```

```
Size of the matrix: 9
```

In [35]:
```
# Number of dimension: .ndim function
print('Number of dimension of the matrix:\n',matrix_1.ndim)
```

```
Number of dimension of the matrix:
 2
```

In [36]:
```
# Create a function that adds 100 to something:
add_100 = lambda i: i+100
```

In [37]:
```
# Convert it into a vectorized function:
vectorized_add_100 = np.vectorize(add_100)
```

In [38]:
```
# Apply function add_100 to a matrix:
print('Add 100 to all element of the matrix:\n',vectorized_add_100(matrix_1))
```

```
Add 100 to all element of the matrix:
 [[101 102 103]
 [103 104 105]
 [105 106 108]]
```

In [39]:
```
# Create a function to multiply by 2:
multi_2 = lambda i: i*2
```

In [40]:
```
# Convert:
vectorized_multi_2 = np.vectorize(multi_2)
```

In [41]:
```
# New matrix:
print('New matrix:\n',vectorized_multi_2(matrix_1))
```

```
New matrix:
 [[ 2  4  6]
 [ 6  8 10]
 [10 12 16]]
```

In [43]:
```
# Find max and min value in a matrix:
```

In [44]:
```
# Max:
print('maximum value of the matrix is:',matrix_1.max())
```

```
maximum value of the matrix is: 8
```

In [45]:
```
# Min:
print('minimum value of the matrix is:',matrix_1.min())
```

```
minimum value of the matrix is: 1
```

In [ ]:
```
# Min in each column:
print('Min in each column of the matrix')
```

In [ ]:
```
# Calculate average, variance, standard deviation
```

In [ ]:
```
# Mean: .mean() function
```

In [4]:
```
print('Mean:\n',matrix_1.mean())
```

```
Mean:
 4.111111111111111
```

In [ ]:
```
# Standard deviation: .std() function
```

In [5]:
```
print('Standard deviation:\n',matrix_1.std())
```

```
Standard deviation:
```

```
2.0245407953653998
```

In [ ]:
```
# Variance:
```

In [6]:
```
print('Variance:\n',matrix_1.var())
```

```
Variance:
 4.098765432098765
```

In [ ]:
```
# Reshaping arrays:
```

In [8]:
```
# Reshape: .reshape()
print('New matrix_1:\n',matrix_1.reshape(9,1))
```

```
New matrix_1:
 [[1]
 [2]
 [3]
 [3]
 [4]
 [5]
 [5]
 [6]
 [8]]
```

In [10]:
```
print('New matrix_1\n',matrix_1.reshape(9))
```

```
New matrix_1
 [1 2 3 3 4 5 5 6 8]
```

In [20]:
```
print('New_matrix_1\n',matrix_1.flatten())
```

```
New_matrix_1
 [1 2 3 3 4 5 5 6 8]
```

In [13]:
```
# Transposing a matrix: .T
print('Transposed matrix:\n',matrix_1.T)
```

```
Transposed matrix:
 [[1 3 5]
 [2 4 6]
 [3 5 8]]
```

In [15]:
```
# Find determant and Rank of a matrix:
print('Deterimant:\n',np.linalg.det(matrix_1))
```

```
Deterimant:
 -1.999999999999999
```

In [16]:
```
# Calculate the rank:
print('Rank:\n',np.linalg.matrix_rank(matrix_1))
```

```
Rank:
 3
```

In [17]:
```
# Diagonal of a matrix:
print('Principal diagonal:',matrix_1.diagonal())
```

```
Principal diagonal: [1 4 8]
```

In [19]:
```
# Diagonal one above the principal diagonal:
print('One above:\n',matrix_1.diagonal(offset=1))
```

```
One above:
 [2 5]
```

In [21]:
```
# Diagonal one below principal diagonal:
print('One below:\n',matrix_1.diagonal(offset=-1))
```

```
One below:
 [3 6]
```

In [22]:
```
# Calculate the trace of a matrix: .trace()
print('Trace:\n',matrix_1.trace())
```

```
Trace:
 13
```

In [23]:
```
# Calculate eigenvalues
eigenvalues, eigenvectors = np.linalg.eig(matrix_1)
print('Eigenvalues:\n',eigenvalues)
print('Eigenvectors:\n',eigenvectors)
```

```
Eigenvalues:
 [13.50727705 -0.71450818  0.20723113]
Eigenvectors:
 [[-0.2769271  -0.88967924  0.22901304]
 [-0.51426329  0.09178213 -0.83678426]
 [-0.81169246  0.44726602  0.49733804]]
```

In [24]:
```
# Calculateing Dot Products
# Create vector_1:
vector_1 = np.array([1,2,3])
# Create vector_2:
vector_2 = np.array([4,5,6])
```

In [25]:
```
# Calculate Dot product:
print('Dot Product:\n',np.dot(vector_1,vector_2))
```

```
Dot Product:
 32
```

In [26]:
```
# Calculate Dot product:
print('Dot product:\n',vector_1 @ vector_2)
```

```
Dot product:
 32
```

In [ ]:
```
# Adding, subtract and multiply matrix:
```

In [27]:
```
# Matrix_2:
matrix_2 = np.array([[1,2,3],[4,5,6],[7,8,9]])
```

In [28]:
```
# Matrix_3:
matrix_3 = np.array([[7,8,9],[4,5,6],[1,2,3]])
```

In [29]:
```
# Add 2 matrix:
print('Sum:\n', np.add(matrix_2,matrix_3))
```

```
Sum:
```

```
[[ 8 10 12]
 [ 8 10 12]
 [ 8 10 12]]
```

In [30]:
```
# Subtract 2 matrix:
print('Subtract:\n',np.subtract(matrix_2,matrix_3))
```

```
Subtract:
 [[-6 -6 -6]
 [ 0  0  0]
 [ 6  6  6]]
```

In [31]:
```
# Multiply:
print('Multuplication:\n', matrix_2*matrix_3)
```

```
Multuplication:
 [[ 7 16 27]
 [16 25 36]
 [ 7 16 27]]
```

In [32]:
```
# Inverting a matrix:
print('Inverse matrix:\n', np.linalg.inv(matrix_2))
```

```
Inverse matrix:
 [[ 3.15251974e+15 -6.30503948e+15  3.15251974e+15]
 [-6.30503948e+15  1.26100790e+16 -6.30503948e+15]
 [ 3.15251974e+15 -6.30503948e+15  3.15251974e+15]]
```

In [ ]:
```
```