Numpy Array creation

In [30]:

```
# 1D array
import numpy as np
array_1d = np.array([1,2,3])
print(array_1d)
```

```
[1 2 3]
```

In [29]:

```
# 2D array
import numpy as np
array_2d = np.array([[1,2],[3,4]])
print(array_2d)
```

```
[[1 2]
 [3 4]]
```

In [41]:

```
# 3D array
import numpy as np
arr_3D = np.array([[[1,3,4,5], [3,4,5,7]]])
arr_3D
```

Out[41]:

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

In [34]:

```
#Using dtype parameter
import numpy as np
c = np.array([1,2,3], dtype=complex)
print(c)
```

```
[1.+0.j 2.+0.j 3.+0.j]
```

In [31]:

```
# Creating an array with ones
ones_array = np.ones((2, 3))
print("\nOnes Array:")
print(ones_array)
```

```

Ones Array:
[[1. 1. 1.]
 [1. 1. 1.]]
```

In [35]:

```
# Creating an array with zeros
zeros_array = np.zeros((3, 4))
print("\nZeros Array:")
print(zeros_array)
```

```

Zeros Array:
[[0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]]
```

In [20]:

```
#Create an array with range of elements
import numpy as np
```

```
e = np.arange(10)
print(e)
```

```
[0 1 2 3 4 5 6 7 8 9]
```

In [36]:

```
# Creating an array with a range of values
range_array = np.arange(0, 10, 2)  # Start, stop, step
print("\nRange Array:")
print(range_array)
```

```
Range Array:
[0 2 4 6 8]
```

In [26]:

```
# Creating an array with evenly spaced values
linspace_array = np.linspace(0, 1, 5)  # Start, stop, num
print("\nLinspace Array:")
print(linspace_array)
```

```
Linspace Array:
[0.   0.25 0.5  0.75 1.  ]
```

Creation of numpy array using existing data

Syntax: numpy.asarray(data, dtype=None, order=None)

data = Input data in the form of list, list of tuples, tuples

Dtype = Default data type of input data is applied

Order = C(row major and its default) or F(column major)

In [23]:

```
# convert list to ndarray
import numpy as np
x = [1,2,3]
a = np.asarray(x)
print(a)
```

```
[1 2 3]
```

In [25]:

```
# ndarray from tuple
import numpy as np
x = (4,5,6)
a = np.asarray(x)
print(a) # output [1 2 3]
```

```
[4 5 6]
```

Numpy Array operations

In [37]:

```
# Element-wise addition
arr1 = np.array([1, 2, 3])
```

```
arr2 = np.array([4, 5, 6])
result_addition = arr1 + arr2
print("Element-wise Addition:")
print(result_addition)

# Element-wise multiplication
result_multiply = arr1 * arr2
print("\nElement-wise Multiplication:")
print(result_multiply)

# Matrix multiplication
matrix1 = np.array([[1, 2], [3, 4]])
matrix2 = np.array([[5, 6], [7, 8]])
result_matrix_multiply = np.dot(matrix1, matrix2)
print("\nMatrix Multiplication:")
print(result_matrix_multiply)
```

```
Element-wise Addition:
[5 7 9]

Element-wise Multiplication:
[ 4 10 18]

Matrix Multiplication:
[[19 22]
 [43 50]]
```

In [39]:

```
# Creating a 1D array
arr_1d = np.array([1, 2, 3, 4, 5])
print("1D Array:")
print(arr_1d)

# Creating a 2D array
arr_2d = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
print("\n2D Array:")
print(arr_2d)
```

```
1D Array:
[1 2 3 4 5]

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

In [40]:

```
# Accessing elements
print("First element:", arr1[0])
print("Element at row 1, column 2:", arr_2d[1, 2])

# Slicing
print("\nSlicing 1D array:", arr1[1:3])
print("Slicing 2D array:")
print(arr_2d[0:2, 1:3])
```

```
First element: 1
Element at row 1, column 2: 6

Slicing 1D array: [2 3]
Slicing 2D array:
[[2 3]
 [5 6]]
```

```
In [ ]:
```