

## ◆ What is NumPy?

**NumPy (Numerical Python)** is an open-source Python library used for **numerical and scientific computing**.

It provides support for **multi-dimensional arrays**, matrices, and powerful **mathematical functions**.

NumPy is faster and more memory-efficient than Python lists because it is implemented internally in **C language**. It allows **vectorized operations**, which means calculations are performed on entire arrays without using loops.

NumPy is the **core foundation** of Data Science, Machine Learning, and Artificial Intelligence in Python.

---

## ◆ Why do we use NumPy?

We use NumPy because:

- It performs **fast numerical calculations**
  - It handles **large datasets efficiently**
  - It supports **array and matrix operations**
  - It is essential for **AI, ML, and Data Analysis**
- 

## ◆ Features of NumPy

- N-dimensional array object (`ndarray`)
- Fast and efficient computation
- Vectorized operations
- Broadcasting support

- Linear algebra functions
  - Mathematical and statistical functions
  - Memory-efficient storage
- 

### ◆ Advantages of NumPy

- Faster than Python lists
  - Uses less memory
  - Easy mathematical operations
  - No need for loops
  - Ideal for AI and ML applications
  - Works well with other libraries like Pandas and Matplotlib
- 

### ◆ NumPy Array vs Python List

Python List	NumPy Array
Slow for calculations	Very fast
Uses more memory	Uses less memory
Loop required for math	No loop needed
Not suitable for AI	Best for AI/ML
General purpose	Numerical computing

---

### ◆ Example: List vs NumPy Python

List

```
l = [1, 2, 3] print(l
```

\* 2) Output:

```
[1, 2, 3, 1, 2, 3]
```

## **NumPy Array**

```
import numpy as np
```

```
a = np.array([1,
```

```
2, 3]) print(a * 2) Output:
```

```
[2 4 6]
```

```
1
2 import numpy as np
3 # Python list
4 l = [1, 2, 3, 4]
5
6 # NumPy array
7 a = np.array([1, 2, 3, 4])
8
9 print(a)
10 print(l)
11 print(a*2)
12 print(l*2)
13
14
15 l1 = [1, 2, 3]
16 l2 = [4, 5, 6]
17
18 result = []
19 for i in range(len(l1)):
20     result.append(l1[i] + l2[i])
21
22 print(result)
23
24 a = np.array([1,2,3])
25 b = np.array([4,5,6])
26
27 print(a + b)
28
```

◆ Question 1: Create a NumPy array Create an array of numbers from 1 to 10

✓ Solution:

```
import numpy as np
```

```
arr = np.arange(1, 11) print(arr)
```

---

## ◆ Question 2: Find shape and datatype

### Check shape and datatype of an array

#### ✓ Solution:

```
arr = np.array([10, 20, 30, 40])
```

```
print(arr.shape)
```

```
print(arr.dtype)
```

---

## ◆ Question 3: Create a 2D array

### Create a 2×3 matrix

#### ✓ Solution:

```
arr = np.array([[1, 2, 3],  
                [4, 5, 6]]) print(arr)
```

---

## ◆ Question 4: Find max, min, and mean Find

### maximum, minimum, and average of an array

#### ✓ Solution:

```
arr = np.array([5, 10, 15, 20])
```

```
print("Max:", arr.max())
```

```
print("Min:", arr.min())
```

```
print("Mean:", arr.mean())
```

## ◆ STEP 1: Array Creation

### 1 1D, 2D, 3D Arrays

```
import numpy as np

a = np.array([1, 2, 3])      # 1D
b = np.array([[1, 2], [3, 4]]) # 2D
c = np.array([[[1, 2]]])    # 3D
```

## ★ QUESTION --

Create a 2D array of size  $3 \times 3$  ?

◆ STEP 2: Special Arrays  
(VERY IMPORTANT)

```
np.zeros((2,3))
np.ones((3,3))
np.arange(1,10,2)
np.linspace(1,5,5)
np.eye(3)
```