### Splitting (Divide Arrays)

**x = np.arange(1, 10)**

**print(np.split(x, 3))**  # 3 equal parts

# [array([1, 2, 3]), array([4, 5, 6]), array([7, 8, 9])]

## 🔹 6. Broadcasting

Broadcasting allows operations on arrays of **different shapes**.

**a = np.array([1, 2, 3])**

**b = 5**

**print(a + b)**  # [6 7 8] (scalar added to every element)

👉 Works with different shapes:

**mat = np.array([[1, 2, 3],**

**[4, 5, 6]])**

**print(mat + a)**

# [[ 2 4 6]

# [ 5 7 9]]

## 📝 Mini Exercise (Classwork)

1. Create a 1D array of numbers 1–12.  
   * Reshape into 3×4.
   * Calculate **row-wise mean**.
2. Create a 2D array (3×3) with numbers 1–9.  
   * Find max of each column.
   * Transpose the matrix.
3. Add a scalar (10) to an array [1, 2, 3, 4, 5] using broadcasting.

# 📘 NumPy – Day 3 Notes

**Topic:** Random, Boolean Indexing, Conditional Ops, Linear Algebra

## 🔹 1. NumPy Random Module

Used to generate random numbers.

# Random float between 0 and 1

**print(np.random.rand(5))**

# Random integers

**print(np.random.randint(1, 10, size=5))** # 5 numbers between 1 and 9

# Random normal distribution (mean=0, std=1)

**print(np.random.randn(5))**

# Random 2D array

**print(np.random.randint(1, 100, size=(3, 3)))**

# Fixing results (Reproducibility)

**np.random.seed(42)**

**print(np.random.randint(1, 10, 5))**

## 🔹 2. Boolean Indexing & Filtering

**arr = np.array([10, 20, 30, 40, 50])**

# Condition based selection

**print(arr[arr > 25])**  # [30 40 50]

# Multiple conditions

**print(arr[(arr > 15) & (arr < 45)])**  # [20 30 40]

## 🔹 3. Conditional Operations

**a = np.array([10, 20, 30, 40, 50])**

#np.where(condition, value\_if\_true, value\_if\_false)

**print(np.where(a > 25, "High", "Low"))**

# ['Low' 'Low' 'High' 'High' 'High']

# Check if any/all elements satisfy condition

**print(np.any(a > 40))**  # True

**print(np.all(a > 5))**  # True

## 🔹 5. Linear Algebra with NumPy

## 1. Matrix Addition

**import numpy as np**

**A = np.array([[1, 2],**

**[3, 4]])**

**B = np.array([[5, 6],**

**[7, 8]])**

**C = A + B**

**print("Matrix Addition:\n", C)**

**Output:**

Matrix Addition:

[[ 6 8]

[10 12]]

## 2. Matrix Subtraction

**C = A - B**

**print("Matrix Subtraction:\n", C)**

**Output:**

Matrix Subtraction:

[[-4 -4]

[-4 -4]]

## 3. Element-wise Multiplication

**C = A \* B**

**print("Element-wise Multiplication:\n", C)**

**Output:**

Element-wise Multiplication:

[[ 5 12]

[21 32]]

## 4. Matrix Multiplication (@ or dot)

**C = A @ B # or np.dot(A, B)**

**print("Matrix Multiplication:\n", C)**

**Output:**

Matrix Multiplication:

[[19 22]

[43 50]]

## 5. Transpose

**print("Transpose of A:\n", A.T)**

**Output:**

Transpose of A:

[[1 3]

[2 4]]

## 6. Determinant

**det\_A = np.linalg.det(A)**

**print("Determinant of A:", det\_A)**

**Output:**

Determinant of A: -2.0000000000000004

**Formula:**

det(A)=ad−bc

## 7. Inverse

**inv\_A = np.linalg.inv(A)**

**print("Inverse of A:\n", inv\_A)**

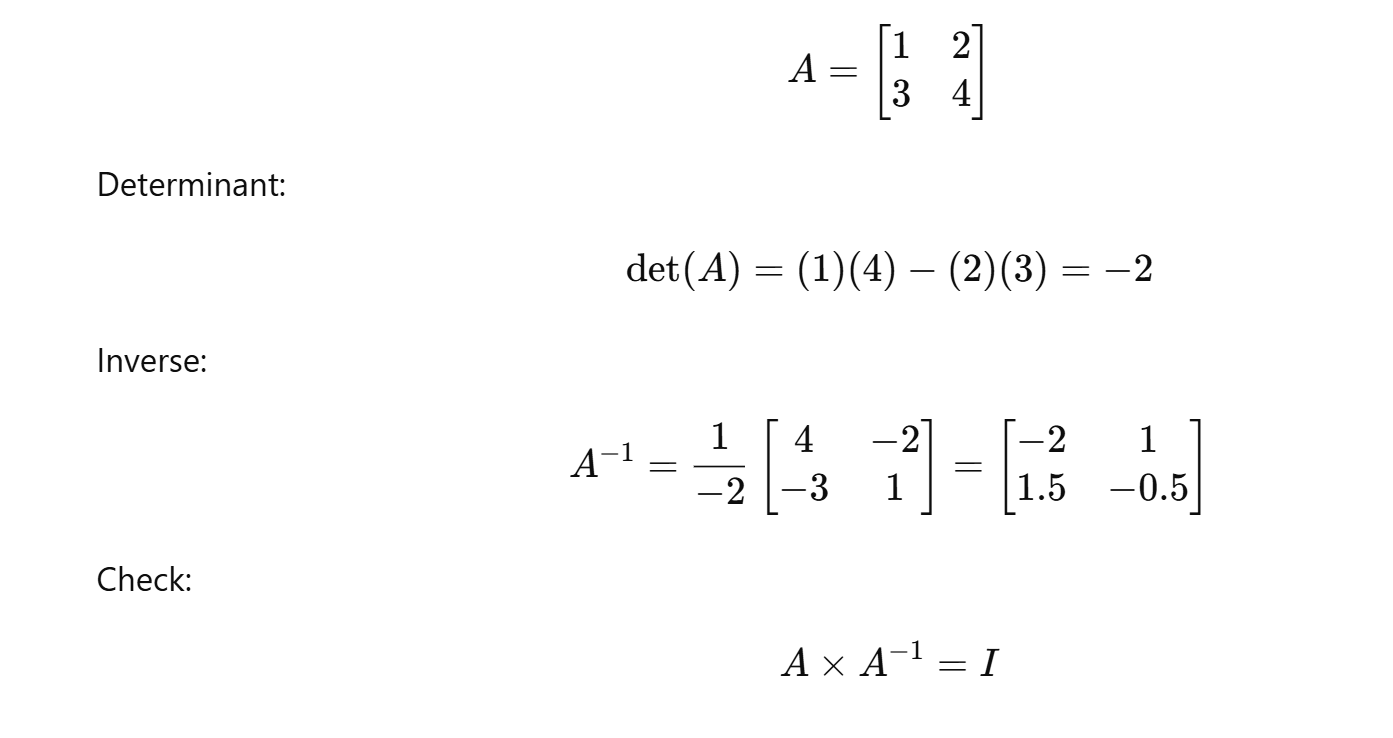
**Output:**

Inverse of A:

[[-2. 1. ]

[ 1.5 -0.5]]

**Formula :**



## 8. Trace

**Trace** = sum of diagonals (like a quick summary of the matrix).

**trace\_A = np.trace(A)**

**print("Trace of A:", trace\_A)**

**Output:**

Trace of A: 5

## 📝 Classwork Exercise

1. Create two 3×3 matrices X and Y.  
   * Perform **addition, subtraction, element-wise multiplication, and matrix multiplication**.
2. Find **transpose, determinant, inverse, rank, and trace** of X.
3. Verify that X @ inv(X) = Identity Matrix.

## 

1. Generate 10 random integers between 50–100.  
   * Find mean & std.
   * Filter values above mean.
2. Create a 3×3 matrix with random values.  
   * Find determinant & inverse.
3. Given arr = [5, 15, 25, 35, 45], replace numbers > 30 with -1 using np.where.