**🐍 DAY 14 – HOME ASSIGNMENTS: Introduction to NumPy**

**🔹 Part A: Creating NumPy Arrays**

1. Import NumPy and:
   * Create a 1D array with numbers from 1–10.
   * Create a 2D array of shape (3, 3) with numbers from 1–9.
2. Create an array of:
   * All zeros (shape = (3, 3))
   * All ones (shape = (2, 5))
   * All random numbers (shape = (4,))

**🔹 Part B: Inspecting Arrays**

1. Given:

arr = np.arange(12).reshape(3, 4)

* + Print its **shape**, **size**, **ndim**, and **dtype**.

1. Change the dtype of an array from float64 to int32.

**🔹 Part C: Indexing and Slicing**

1. Create an array:

arr = np.arange(1, 21).reshape(4, 5)

* + Print the first row.
  + Print the last column.
  + Slice the subarray of the first 2 rows and first 3 columns.

**🔹 Part D: Array Operations**

1. Elementwise operations:
   * Create two arrays of the same shape (e.g., a = [[1,2],[3,4]], b = [[5,6],[7,8]])
   * Perform elementwise:
     + Addition
     + Subtraction
     + Multiplication
     + Division
2. Find:
   * Sum of all elements
   * Min, max, and average of an array
3. Use NumPy to compute:
   * Square root of every element in an array
   * Square of every element in an array

**🔹 Part E: Reshaping and Stacking**

1. Create a 1D array with numbers from 1–12 and:
   * Reshape it into a 3×4 array.
   * Flatten it back to a 1D array.
2. Stack arrays:
   * Create a = [1, 2, 3] and b = [4, 5, 6]
   * Stack them **vertically** and **horizontally**.

**🔹 Part F: Advanced NumPy Practice**

1. Create an array of 100 random numbers and:
   * Find the top 5 maximum numbers
   * Find the average
   * Find the indices of all numbers > 50
2. Create a 5x5 array of random numbers between 10–100 and:
   * Find its transpose
   * Find the sum of its diagonal
   * Extract the diagonal elements
3. Compare Lists vs NumPy:
   * Create a Python list and a NumPy array of the same data.
   * Compare their memory usage using sys.getsizeof() and .nbytes.
   * Compare their performance for an operation like adding 1 to every element.
4. **Prime Numbers with NumPy**:
   * Create an array of numbers from 1–100.
   * Use boolean masking to filter out only the prime numbers.
5. **Moving Averages**:
   * Given an array of daily temperatures, write a NumPy script to compute a 3‑day moving average.