NumPy

Arrays, Indexing, Slicing

Concept: NumPy Arrays

- NumPy arrays are like **supercharged Python lists**, designed for mathematical and matrix operations.
- They are **faster** and use **less memory** than lists.

□ Example 1: Creating Arrays

```
import numpy as np

# 1D array
arr1 = np.array([1, 2, 3, 4, 5])
print("1D Array:", arr1)

# 2D array (matrix)
arr2 = np.array([[1, 2, 3], [4, 5, 6]])
print("\n2D Array:\n", arr2)

# Array of zeros
zeros = np.zeros((3,3))
print("\nZeros Array:\n", zeros)

# Array of ones
ones = np.ones((2,4))
print("\nOnes Array:\n", ones)

# Range
rng = np.arange(0, 10, 2)
print("\nArray with step:", rng)
```

□ Example 2: Indexing & Slicing

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

print(arr[0])      # First element
print(arr[-1])     # Last element
print(arr[1:4])    # Slice from index 1 to 3
print(arr[:3])     # First 3 elements
print(arr[::2])    # Every second element
```

□ Example 3: 2D Indexing

```
mat = np.array([[10, 20, 30],
                [40, 50, 60],
                [70, 80, 90]])

print(mat[0, 0])    # First row, first column
print(mat[1, 2])    # Second row, third column
print(mat[:, 1])    # All rows, 2nd column
print(mat[1, :])    # Entire 2nd row
```

□ Scenario

Imagine you're analyzing student scores stored in a **2D array** where rows represent students and columns represent test scores.

- You can slice to get **all scores of a student** or **all students' scores in a subject**.

```
scores = np.array([
    [85, 90, 78],
    [88, 76, 92],
    [90, 91, 85]
])

# Get all scores of student 1
print("Student 1 scores:", scores[0, :])

# Get all students' Math scores (2nd column)
print("Math scores:", scores[:, 1])
```

□ Assignment

1. Create a NumPy array of numbers from 1 to 100. Extract:
    - All even numbers
    - All multiples of 5
    - Last 10 numbers
2. Create a 3x3 matrix and:
    - Get the 2nd row
    - Get the 3rd column
    - Slice the first 2 rows & 2 columns

□ Quiz

1. How do you create a 5x5 array of all ones?
2. What does `arr[::2]` do?

3. If `mat = np.array([[1,2],[3,4]])`, what is `mat[1,0]`?
4. Difference between `np.arange(5)` and `np.linspace(0,1,5)`?

□ Capstone Mini Project (Arrays, Indexing, Slicing)

Student Grade System

- Create a 5x5 matrix representing grades of 5 students across 5 subjects.
- Extract:
  - Grades of student 3
  - All students' grades in subject 4
  - Average grade per student
  - Highest score in subject 2

Vectorization & Broadcasting

□ Concept

- **Vectorization**: Performing operations on entire arrays instead of loops.
- **Broadcasting**: Numpy's ability to **stretch smaller arrays** to match larger arrays during arithmetic.

□ Example 1: Vectorization

Instead of looping through lists:

```
# Normal Python
lst = [1, 2, 3, 4]
doubled = [x*2 for x in lst]
print(doubled)

# NumPy
arr = np.array([1, 2, 3, 4])
print(arr * 2)   # Element-wise multiplication
```

□ Example 2: Broadcasting

```
# Add scalar to array
arr = np.array([10, 20, 30, 40])
print(arr + 5)   # Adds 5 to every element
```

```
# 2D Broadcasting
mat = np.array([[1,2,3],
                [4,5,6]])

vec = np.array([10, 20, 30])

# Each row of mat gets vec added
print(mat + vec)
```

□ Scenario

A company has **monthly sales data** for 3 products. They decide to increase prices by **10%**. Instead of looping, broadcasting does it in one line.

```
sales = np.array([[100, 200, 150],
                  [120, 210, 160],
                  [130, 220, 170]])

increase = 1.10  # 10% increase
new_sales = sales * increase
print(new_sales)
```

□ Assignment

1. Create a NumPy array of 10 numbers. Multiply all numbers by 3 without using a loop.
2. Given `mat = [[1,2],[3,4],[5,6]]`, add `[10,20]` to every row using broadcasting.
3. Compute the square of numbers from 1–20 using vectorization.

□ Quiz

1. What is broadcasting in NumPy?
2. How is `arr + 5` different from looping through the array?
3. If `arr = np.array([1,2,3])` and `arr2 = np.array([10])`, what is `arr + arr2`?

□ Capstone Mini Project (Vectorization & Broadcasting)

Weather Data Analysis

- Generate a NumPy array of temperatures (°C) for 7 days and 3 cities.
- Convert all temperatures to Fahrenheit using vectorization.
- Increase all temperatures of city 2 by 2°C using broadcasting.
- Find the hottest day for each city.