**2. Array indexing and  slicing**

# Using a Python list

fruits = ["apple", "banana", "cherry", "date", "strawberry"]

print("Original list:", fruits)

# Indexing

print("First fruit:", fruits[0])

print("Last fruit:", fruits[-1])

# Slicing

print("Fruits from index 1 to 3:", fruits[1:4])

print("Fruits from start to index 2:", fruits[:3])

print("Fruits from index 2 to end:", fruits[2:])

print("Every second fruit:", fruits[::2])

# Using NumPy array

import numpy as np

numbers = np.array([10, 20, 30, 40, 50, 60])

print("\nOriginal NumPy array:", numbers)

# Indexing

print("Element at index 2:", numbers[2])

# Slicing

print("Slice from index 1 to 4:", numbers[1:5])

print("Slice with step 2:", numbers[::2])

print("Reversed array:", numbers[::-1])

**output:**

Original list: ['apple', 'banana', 'cherry', 'date', 'strawberry']

First fruit: apple

Last fruit: strawberry

Fruits from index 1 to 3: ['banana', 'cherry', 'date']

Fruits from start to index 2: ['apple', 'banana', 'cherry']

Fruits from index 2 to end: ['cherry', 'date', 'strawberry']

Every second fruit: ['apple', 'cherry', 'strawberry']

Original NumPy array: [10 20 30 40 50 60]

Element at index 2: 30

Slice from index 1 to 4: [20 30 40 50]

Slice with step 2: [10 30 50]

Reversed array: [60 50 40 30 20 10]

**3. Reshaping and flattening**

# Reshaping the array

reshaped\_array = array.reshape(1, 9)

print("Reshaped array:\n", reshaped\_array)

**op:**[[1 2 3 4 5 6 7 8 9]]

# Flattening the array

flattened\_array = array.flatten()

print("Flattened array:", flattened\_array)

**op:**[1 2 3 4 5 6 7 8 9]

**4. Mathematical operations: Element-wise operations**

import numpy as np

# Create a NumPy array

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

print("Original array:", arr)

# Square each element

squared = arr \*\* 3

print("Squared:", squared)

# Raise each element to a specific power

power = np.power(arr, 3)  # Cube each element

print("Cubed (Power of 3):", power)

# Square root of each element

sqrt = np.sqrt(arr)

print("Square root:", sqrt)

# Add a constant to each element

added = arr + 7

print("Array + 7:", added)

# Multiply each element by a constant

multiplied = arr \* 6

print("Array \* 6:", multiplied)

**output:**

Original array: [1 2 3 4 5]

Squared: [  1   8  27  64 125]

Cubed (Power of 3): [  1   8  27  64 125]

Square root: [1.         1.41421356 1.73205081 2.         2.23606798]

Array + 7: [ 8  9 10 11 12]

Array \* 6: [ 6 12 18 24 30]

**Statistical &aggregation functions**

import numpy as np

# Sample data

data = np.array([15, 20, 25, 40, 35, 35, 50, 55, 60, 70, 65])

print("Data:", data)

# Aggregation & Statistical Functions

print("Sum:", np.sum(data))

print("Mean:", np.mean(data))

print("Median:", np.median(data))

print("Standard Deviation (std):", np.std(data))

print("Variance (var):", np.var(data))

print("Maximum:", np.max(data))

print("Minimum:", np.min(data))

print("Product of all elements:", np.prod(data))

print("25th Percentile:", np.percentile(data, 15))

print("Mode:", stats.mode(data, keepdims=False))

**Output:**

Data: [15 20 25 40 35 35 50 55 60 70 65]

Sum: 470

Mean: 42.72727272727273

Median: 40.0

Standard Deviation (std): 17.756386349658737

Variance (var): 315.2892561983471

Maximum: 70

Minimum: 15

Product of all elements: 275900625000000000

25th Percentile: 22.5