**Department of CSE (ICB)**

**Class: SY, Semester: IV**

**Subject: Python Programming**

**Experiment No.: 4**

**AIM : Implement Arrays / Numpy Array (1D, 2D) applications.**

**Exercise 1:**

Arrays are data structures that store multiple values of the same data type in a contiguous memory block.

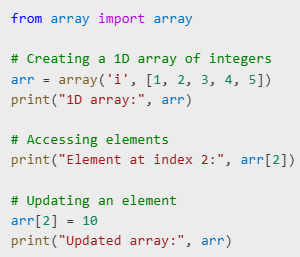
Python does not have built-in support for arrays like other languages (e.g., C or Java). Instead, Python uses lists for similar functionality. However, for high-performance applications, the NumPy library provides efficient, multidimensional arrays called NumPy arrays.

NumPy (Numerical Python) is a library used for numerical and scientific computing. It offers powerful tools for creating and manipulating arrays with greater efficiency than native Python lists.

1D and 2D Arrays in Python

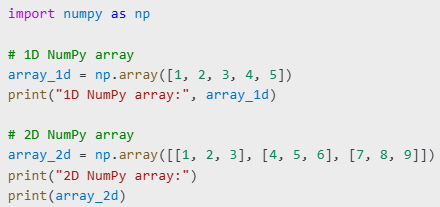
Using Python's Built-in array Module

Python provides an array module in its standard library.



The array module is limited to single-dimensional arrays and does not support advanced operations. Therefore, NumPy is preferred for multi-dimensional arrays.

Creating NumPy Arrays



**1. Data Storage and Retrieval**

Application: Storing a collection of numbers or strings in memory for quick access.

* Example: Store student grades or temperatures recorded over a week.

# Storing temperatures for a week

temperatures = np.array([72, 75, 78, 79, 73, 74, 77])

print("Average temperature:", np.mean(temperatures))

A close-up of a computer code

Description automatically generated

**2. Mathematical and Statistical Computations**

Application: Efficient numerical computations on large data sets.

* Example: Calculating mean, median, standard deviation.

A screenshot of a computer code

Description automatically generated

**3. Image Processing**

Application: Images are stored as 2D arrays of pixel values.

* Example: Converting an image to grayscale.



**4. Data Transformation**

Application: Reshaping and transposing data arrays.

A screenshot of a computer code

Description automatically generated

**5. Linear Algebra**

Application: Matrix operations for scientific computing.

A screenshot of a computer code

Description automatically generated

**6. Sorting and Searching**

Application: Finding elements and sorting data.

A screenshot of a computer code

Description automatically generated

**Conclusion**

Arrays and NumPy arrays are powerful tools for efficient data manipulation and computation in Python. NumPy’s advanced functionality makes it a standard choice for scientific computing, machine learning, and data analysis.

**Exercises:**

**Create two 1D arrays, a = [1, 2, 3] and b = [4, 5, 6]. Perform the following operations:**

1. Add the arrays element-wise.
2. Multiply the arrays element-wise.
3. Find the dot product of a and b.
4. Create an array of 10 zeros using numpy.zeros().
5. Create an array of 5 ones and multiply each element by 3.
6. Create an array with numbers from 1 to 20 using numpy.arange().
7. Create a 1D array with 6 equally spaced values between 0 and 1 using numpy.linspace().
8. Reverse the array from exercise 6.
9. Create two 2D arrays

A number and a line

Description automatically generated with medium confidence

**Perform the following:**

1. Add A and B.
2. Multiply A and B element-wise.
3. Compute the matrix product of A and B using np.dot() or @.
4. Create a 2D array of size 4x4 filled with random integers between 0 and 100 using numpy.random.randint().
5. Extract the first two rows and first two columns from the array created in exercise 10
6. Create a 1D array with values from 1 to 12 and reshape it into a 3x4 matrix.
7. Flatten a 3x3 matrix into a 1D array.