## 🔹 What is an Array?

An **array** is a collection of elements stored at **contiguous memory locations**.

* It stores multiple items of the **same type**.
* Each item can be accessed using an **index** (0-based indexing in most languages).

👉 Think of it as a row of boxes, each with a number (index) written on it.

## 🔹 Array Basics

* **Fixed size** (in most languages like Java).
* **Fast access** (O(1) time) using index.
* **Insert/Delete** can be costly (O(n)), since elements may need to shift.

## 🔹 Example Problem

Let’s start simple:  
**Problem:** Store 5 numbers in an array, print them, and calculate their sum.

### ✅ Python

# Array in Python (list is dynamic but works like array)

arr = [10, 20, 30, 40, 50]

# Print elements

for num in arr:

print(num, end=" ")

# Sum of elements

total = sum(arr)

print("\nSum =", total)

### ✅ Java

public class ArrayExample {

public static void main(String[] args) {

int[] arr = {10, 20, 30, 40, 50};

// Print elements

for (int num : arr) {

System.out.print(num + " ");

}

// Sum of elements

int total = 0;

for (int num : arr) {

total += num;

}

System.out.println("\nSum = " + total);

}

}

### ✅ JavaScript

let arr = [10, 20, 30, 40, 50];

// Print elements

arr.forEach(num => process.stdout.write(num + " "));

// Sum of elements

let total = arr.reduce((a, b) => a + b, 0);

console.log("\nSum =", total);

## 🔹 Key Operations with Arrays

We’ll cover step by step:

1. Traversal (visiting all elements) ✅
2. Insertion (add element at index)
3. Deletion (remove element)
4. Searching (linear & binary)
5. Sorting

**Find the n number**

**Problem with array1.equals(array2)**

In Java, **arrays are objects**, and the equals() method for arrays **does NOT compare contents**. It behaves like == and checks if both arrays **refer to the same object in memory**.

So even if two arrays have the same elements:

Using Arrays.equals() (built-in method

import java.util.Arrays;

if(Arrays.equals(array1, array2)){

System.out.println("Arrays are equal");

}else{

System.out.println("Arrays are not equal");

}

Alright 👍 let’s now do the **same thing but for 1 to 100**.

**Step 1: What we expect**

Normally, we should have all numbers from **1 to 100**:

[1,2,3,4,...,100][1, 2, 3, 4, ..., 100][1,2,3,4,...,100]

That means there are **100 numbers in total**.

**Step 2: But one is missing**

Example:

[1,2,3,4,6,7,...,100][1, 2, 3, 4, 6, 7, ..., 100][1,2,3,4,6,7,...,100]

Here, **5 is missing**.

**Step 3: Formula method**

1. **Expected sum** of numbers from 1 to 100 is:

100(101)2=5050\frac{100(101)}{2} = 50502100(101)​=5050

1. **Actual sum** → add all numbers in your given array.

Example: if the array is missing 5, then sum will be:

504550455045

1. **Missing number = Expected sum − Actual sum**

5050−5045=55050 - 5045 = 55050−5045=5

✅ So the missing number is **5**.

This works no matter which number is missing.