What is an Array? Why Do We Need an Array? 📝

An **array** is a data structure that stores a fixed-size, sequential collection of elements of the same data type. Think of it as a single cabinet with multiple drawers, where each drawer can hold only one type of item (e.g., all numbers, all names). We need arrays because they allow us to manage and access multiple related pieces of data under a single variable name, rather than declaring a separate variable for each item.

Real-Time Use Case:

In a student management system, instead of creating individual variables like student1Grade, student2Grade, student3Grade, we can use a single array, studentGrades, to store all the grades together. This makes it easy to loop through all the grades and perform operations like calculating the average.

What is a Single-Dimensional Array?

A single-dimensional array is the simplest form of an array. It represents a linear list of elements. Each element in the array is accessed by its index, which starts at 0.

Code Example:

Java

```
// An array to store the days of the week
String[] daysOfWeek = new String[7]; // Declares an array that can hold 7 strings


// Assigning values to the array elements
daysOfWeek[0] = "Sunday";
daysOfWeek[1] = "Monday";


// Accessing an element
String firstDay = daysOfWeek[0]; // firstDay will be "Sunday"
```

Different Ways of Initializing Single-Dimensional Arrays

You can initialize a single-dimensional array in several ways:

1. **Declaration, Instantiation, and Initialization in Separate Steps:**

Java

```
String[] fruits; // Declaration
fruits = new String[3]; // Instantiation
fruits[0] = "Apple"; // Initialization
fruits[1] = "Banana";
fruits[2] = "Cherry";
```

2. **Declaration, Instantiation, and Initialization in One Line:**

Java

```
String[] fruits = {"Apple", "Banana", "Cherry"};
```

3. **Using new and a size, but without initial values:**

Java

```
int[] numbers = new int[5]; // Creates an array of 5 integers, all initialized to 0
```

Default Values of Array Elements

When you create an array using the new keyword but don't explicitly initialize its elements, Java automatically assigns a **default value** to each element. This default value depends on the array's data type.

Data Type	Default Value
int, short, byte, long	0
float, double	0.0
boolean	false
char	'\u0000' (null character)
Objects (e.g., String)	null

Code Example:

Java

```
public class ArrayDefaultValues {
   public static void main(String[] args) {
      int[] numbers = new int[3];
      String[] names = new String[2];
      boolean[] flags = new boolean[1];

      System.out.println(numbers[0]); // Output: 0
      System.out.println(names[0]);   // Output: null
      System.out.println(flags[0]);   // Output: false
   }
}
```

Interview Questions on Arrays

1. **What is an array in Java, and what is its key limitation?**
    - **Answer:** An array is a fixed-size data structure that stores elements of the same data type. Its key limitation is that its size cannot be changed after it is created.
2. **How do you declare and initialize a single-dimensional array of integers with a size of 10?**
    - **Answer:** int[] myNumbers = new int[10];
3. **What is the default value of an element in an array of type String?**
    - **Answer:** The default value is null.
4. **How do you access the third element of an array named myArray?**
    - **Answer:** myArray[2], because array indices are zero-based.
5. **Why do we get an ArrayIndexOutOfBoundsException?**
    - **Answer:** This exception is thrown when you try to access an array element using an index that is outside the valid range (from 0 to array.length - 1).