

Data Structures and Algorithms

BY : DR. JAWAD ALZAMILY

Java Arrays

Java Arrays

An array is a collection of similar types of **data**.

For example, if we want to store the names of 100 people then we can create an array of the string type that can store 100 names.

```
String[] array = new String[100];
```

Here, the above array cannot store more than 100 names. The number of values in a Java array is always fixed

Java Arrays

How to declare an array in Java?

In Java, here is how we can declare an array.

dataType[] arrayName;

dataType : it can be primitive data types like int, char, double, byte, etc. or Java objects

arrayName : it is an identifier

For example,

double[] data;

Here, data is an array that can hold values of type double.

Java Arrays

How to Initialize Arrays in Java?

In Java, we can initialize arrays during declaration. For example,

//declare and initialize an array

```
int[] age = {12, 4, 5, 2, 5};
```

Java Arrays

In the Java array, each memory location is associated with a number. The number is known as an array index. We can also initialize arrays in Java, using the index number. For example,

// declare an array

```
int[] age = new int[5];
```

// initialize array

```
age[0] = 12;
```

```
age[1] = 4;
```

```
age[2] = 5;
```

age[0]	age[1]	age[2]	age[3]	age[4]
12	4	5	2	5

Java Arrays initialization



Java Arrays - Example: Access Array Elements

How to Access Elements of an Array in Java?

```
class Main {  
    public static void main(String[] args) {  
  
        // create an array  
        int[] age = {12, 4, 5, 2, 5};  
  
        // access each array elements  
        System.out.println("Accessing Elements of Array:");  
        System.out.println("First Element: " + age[0]);  
        System.out.println("Second Element: " + age[1]);  
        System.out.println("Third Element: " + age[2]);  
        System.out.println("Fourth Element: " + age[3]);  
        System.out.println("Fifth Element: " + age[4]);  
    }  
}
```

```
Accessing Elements of Array:  
First Element: 12  
Second Element: 4  
Third Element: 5  
Fourth Element: 2  
Fifth Element: 5
```



Java Arrays - Example: Using For Loop

```
class Main {  
    public static void main(String[] args) {  
  
        // create an array  
        int[] age = {12, 4, 5};  
  
        // loop through the array  
        // using for loop  
        System.out.println("Using for Loop:");  
        for(int i = 0; i < age.length; i++) {  
            System.out.println(age[i]);  
        }  
    }  
}
```

```
Using for Loop:  
12  
4  
5
```


Java Arrays - Example: Using the for-each Loop

```
class Main {  
    public static void main(String[] args) {  
  
        // create an array  
        int[] age = {12, 4, 5};  
  
        // loop through the array  
        // using for loop  
        System.out.println("Using for-each Loop:");  
        for(int a : age) {  
            System.out.println(a);  
        }  
    }  
}
```

```
Using for-each Loop:  
12  
4  
5
```

Java Arrays - Example: Compute Sum and Average of Array Elements

```
class Main {  
    public static void main(String[] args) {  
  
        int[] numbers = {2, -9, 0, 5, 12, -25, 22, 9, 8, 12};  
        int sum = 0;  
        Double average;  
  
        // access all elements using for each loop  
        // add each element in sum  
        for (int number: numbers) {  
            sum += number;  
        }  
  
        // get the total number of elements  
        int arrayLength = numbers.length;  
  
        // calculate the average  
        // convert the average from int to double  
        average = ((double)sum / (double)arrayLength);  
  
        System.out.println("Sum = " + sum);  
        System.out.println("Average = " + average);  
    }  
}
```

```
Sum = 36  
Average = 3.6
```