Java Arrays

**Java array** is an object which contains elements of a similar data type. Additionally, The elements of an array are stored in a contiguous memory location. It is a data structure where we store similar elements. We can store only a fixed set of elements in a Java array.

Array in Java is index-based, the first element of the array is stored at the 0th index, 2nd element is stored on 1st index and so on.

Single Dimensional Array in Java

**Syntax to Declare an Array in Java**

* dataType[] arr; (or)
* dataType []arr; (or)
* dataType arr[];

**Instantiation of an Array in Java**

* arrayRefVar=**new** datatype[size];
* **int** a[]={33,3,4,5};//declaration, instantiation and initialization

For-each Loop for Java Array

We can also print the Java array using [**for-each loop**](https://www.javatpoint.com/for-each-loop). The Java for-each loop prints the array elements one by one. It holds an array element in a variable, then executes the body of the loop.

The syntax of the for-each loop is given below:

1. **for**(data\_type variable:array){
2. //body of the loop
3. }

## ArrayIndexOutOfBoundsException

The Java Virtual Machine (JVM) throws an ArrayIndexOutOfBoundsException if length of the array in negative, equal to the array size or greater than the array size while traversing the array.

Multidimensional Array in Java

In such case, data is stored in row and column based index (also known as matrix form).

**Syntax to Declare Multidimensional Array in Java**

1. dataType[][] arrayRefVar; (or)
2. dataType [][]arrayRefVar; (or)
3. dataType arrayRefVar[][]; (or)
4. dataType []arrayRefVar[];

**Example to instantiate Multidimensional Array in Java**

1. **int**[][] arr=**new** **int**[3][3];//3 row and 3 column

## Copying a Java Array

We can copy an array to another by the arraycopy() method of System class.

//Java Program to copy a source array into a destination array in Java

class TestArrayCopyDemo {

    public static void main(String[] args) {

        //declaring a source array

        char[] copyFrom = { 'd', 'e', 'c', 'a', 'f', 'f', 'e',

                'i', 'n', 'a', 't', 'e', 'd' };

        //declaring a destination array

        char[] copyTo = new char[7];

        //copying array using System.arraycopy() method

        System.arraycopy(copyFrom, 2, copyTo, 0, 7);

        //printing the destination array

        System.out.println(String.valueOf(copyTo));

    }

}

in

output: caffeine

## Cloning an Array in Java

Since, Java array implements the Cloneable interface, we can create the clone of the Java array. If we create the clone of a single-dimensional array, it creates the deep copy of the Java array. It means, it will copy the actual value. But, if we create the clone of a multidimensional array, it creates the shallow copy of the Java array which means it copies the references

//Java Program to clone the array

class Testarray1{

public static void main(String args[]){

int arr[]={33,3,4,5};

System.out.println("Printing original array:");

for(int i:arr)

System.out.println(i);

System.out.println("Printing clone of the array:");

int carr[]=arr.clone();

for(int i:carr)

System.out.println(i);

System.out.println("Are both equal?");

System.out.println(arr==carr);

}}