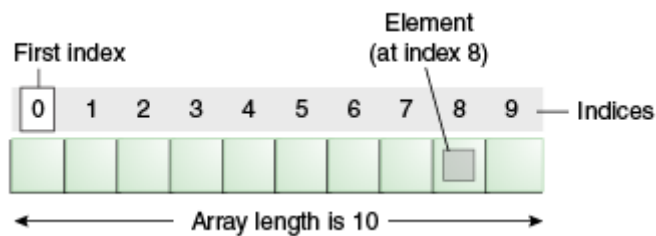


Java Array

Normally, array is a collection of similar type of elements that have contiguous memory location.

Java array is an object that contains elements of similar data type. It is a data structure where we store similar elements. We can store only fixed set of elements in a java array.

Array in java is index based, first element of the array is stored at 0 index.



Advantage of Java Array

- **Code Optimization:** It makes the code optimized, we can retrieve or sort the data easily.
- **Random access:** We can get any data located at any index position.

Disadvantage of Java Array

- **Size Limit:** We can store only fixed size of elements in the array. It doesn't grow its size at runtime. To solve this problem, collection framework is used in java.

Types of Array in java

There are two types of array.

- Single Dimensional Array
- Multidimensional Array

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];
```

Example of single dimensional java array

Let's see the simple example of java array, where we are going to declare, instantiate, initialize and traverse an array.

```
class Testarray{  
    public static void main(String args[]){  
  
        int a[]=new int[5];//declaration and instantiation  
        a[0]=10;//initialization  
        a[1]=20;  
        a[2]=70;  
        a[3]=40;  
        a[4]=50;  
  
        //printing array  
        for(int i=0;i<a.length;i++)//length is the property of array  
            System.out.println(a[i]);  
  
    }  
}
```

Test it Now

```
Output: 10  
        20  
        70  
        40  
        50
```

Declaration, Instantiation and Initialization of Java Array

We can declare, instantiate and initialize the java array together by:

```
int a[]={33,3,4,5};//declaration, instantiation and initialization
```

Let's see the simple example to print this array.

```
class Testarray1{  
    public static void main(String args[]){  
  
        int a[]={33,3,4,5};//declaration, instantiation and initialization
```

```
//printing array
for(int i=0;i<a.length;i++)//length is the property of array
System.out.println(a[i]);

}}
```

Test it Now

Output:33

3

4

5

Passing Array to method in java

We can pass the java array to method so that we can reuse the same logic on any array.

Let's see the simple example to get minimum number of an array using method.

```
class Testarray2{
static void min(int arr[]){
int min=arr[0];
for(int i=1;i<arr.length;i++)
if(min>arr[i])
    min=arr[i];

System.out.println(min);
}

public static void main(String args[]){

int a[]={33,3,4,5};
min(a);//passing array to method
```

```
}}
```

Test it Now

Output:3

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

```
dataType[][] arrayRefVar; (or)  
dataType [][]arrayRefVar; (or)  
dataType arrayRefVar[][]; (or)  
dataType []arrayRefVar[];
```

Example to instantiate Multidimensional Array in java

```
int[][] arr=new int[3][3]; //3 row and 3 column
```

Example to initialize Multidimensional Array in java

```
arr[0][0]=1;  
arr[0][1]=2;  
arr[0][2]=3;  
arr[1][0]=4;  
arr[1][1]=5;  
arr[1][2]=6;  
arr[2][0]=7;  
arr[2][1]=8;  
arr[2][2]=9;
```

Example of Multidimensional java array

Let's see the simple example to declare, instantiate, initialize and print the 2Dimensional array.

```
class Testarray3{
public static void main(String args[]){

//declaring and initializing 2D array
int arr[][]={{1,2,3},{2,4,5},{4,4,5}};

//printing 2D array
for(int i=0;i<3;i++){
    for(int j=0;j<3;j++){
        System.out.print(arr[i][j]+" ");
    }
    System.out.println();
}

}}
```

Test it Now

```
Output:1 2 3
       2 4 5
       4 4 5
```

What is the class name of java array?

In java, array is an object. For array object, an proxy class is created whose name can be obtained by `getClass().getName()` method on the object.

```
class Testarray4{
public static void main(String args[]){

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

Class c=arr.getClass();
```

```
String name=c.getName();

System.out.println(name);

}}
```

Test it Now

Output:I

Copying a java array

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

Syntax of arraycopy method

```
public static void arraycopy(
Object src, int srcPos, Object dest, int destPos, int length
)
```

Example of arraycopy method

```
class TestArrayCopyDemo {
    public static void main(String[] args) {
        char[] copyFrom = { 'd', 'e', 'c', 'a', 'f', 'f', 'e',
                           'i', 'n', 'a', 't', 'e', 'd' };
        char[] copyTo = new char[7];

        System.arraycopy(copyFrom, 2, copyTo, 0, 7);
        System.out.println(new String(copyTo));
    }
}
```

Test it Now

Output:caffein

Addition of 2 matrices in java

Let's see a simple example that adds two matrices.

```
class Testarray5{  
    public static void main(String args[]){  
        //creating two matrices  
        int a[][]={{1,3,4},{3,4,5}};  
        int b[][]={{1,3,4},{3,4,5}};  
  
        //creating another matrix to store the sum of two matrices  
        int c[][]=new int[2][3];  
  
        //adding and printing addition of 2 matrices  
        for(int i=0;i<2;i++){  
            for(int j=0;j<3;j++){  
                c[i][j]=a[i][j]+b[i][j];  
                System.out.print(c[i][j]+" ");  
            }  
            System.out.println();//new line  
        }  
    }  
}
```

Test it Now

```
Output:2 6 8  
        6 8 10
```

[< prev](#)[next >](#)

Share this page



Latest 4 Tutorials



CouchDB



Docker



Rails



RichFaces