

KAMNIS

Java Arrays

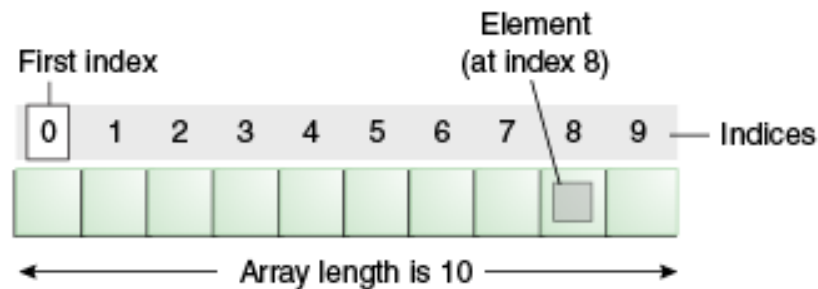
Dr. Ferdin Joe John Joseph

KAMNOETVIDYA

SCIENCE ACADEMY

Arrays

- An array is a container object that holds a fixed number of values of a single type. The length of an array is established when the array is created. After creation, its length is fixed.



An array of 10 elements.

Example 1

```
class ArrayDemo {  
    public static void main(String[] args) {  
        // declares an array of integers  
        int[] anArray;  
        // allocates memory for 10 integers  
        anArray = new int[10];  
        // initialize first element  
        anArray[0] = 100;  
        // initialize second element  
        anArray[1] = 200;  
        // and so forth  
        anArray[2] = 300;  
        anArray[3] = 400;  
        anArray[4] = 500;  
        anArray[5] = 600;  
        anArray[6] = 700;  
        anArray[7] = 800;  
        anArray[8] = 900;  
        anArray[9] = 1000;  
    }  
}
```

```
System.out.println("Element at index 0: " + anArray[0]);  
    System.out.println("Element at index 1: " + anArray[1]);  
    System.out.println("Element at index 2: "+ anArray[2]);  
    System.out.println("Element at index 3: "+ anArray[3]);  
    System.out.println("Element at index 4: "+ anArray[4]);  
    System.out.println("Element at index 5: "+ anArray[5]);  
    System.out.println("Element at index 6: "+ anArray[6]);  
    System.out.println("Element at index 7: "+ anArray[7]);  
    System.out.println("Element at index 8: "+ anArray[8]);  
    System.out.println("Element at index 9: "+ anArray[9]);  
    }  
}
```

Syntax

```
Datatype[] variable_name;    variable_name=new Datatype[size];  
Int[] anArray                anArray=new int[10];
```

Declaring a Variable to Refer to an Array

- `byte[] anArrayOfBytes;`
- `short[] anArrayOfShorts;`
- `long[] anArrayOfLongs;`
- `float[] anArrayOfFloats;`
- `double[] anArrayOfDoubles;`
- `boolean[] anArrayOfBooleans;`
- `char[] anArrayOfChars;`
- `String[] anArrayOfStrings;`

Creating, Initializing, and Accessing an Array

```
int[] anArray = {  
    100, 200, 300,  
    400, 500, 600,  
    700, 800, 900, 1000  
};
```

Single Dimensional

Example 2 – Multi Dimensional

```
class MultiDimArrayDemo {  
    public static void main(String[] args) {  
        String[][] names = {  
            {"Mr. ", "Mrs. ", "Ms. "},  
            {"Smith", "Jones"}  
        };  
        // Mr. Smith  
        System.out.println(names[0][0] + names[1][0]);  
        // Ms. Jones  
        System.out.println(names[0][2] + names[1][1]);  
    }  
}
```

2 Dimensional Matrix - Case

syntax

```
int [ ] [ ] variable_name = new int [ rows ] [ columns ] ;
```

```
// Declares a 2-D array
```

```
int [ ] [ ] scores = new int [ 4 ] [ 5 ] ;
```

```
int [ ] [ ] scores = { { 20, 18, 22, 20, 16 },  
                        { 18, 20, 18, 21, 20 },  
                        { 16, 18, 16, 20, 24 },  
                        { 25, 24, 22, 24, 25 }  
};
```

	columns				
rows	[0][0]	[0][1]	[0][2]	[0][3]	[0][4]
	[1][0]	[1][1]	[1][2]	[1][3]	[1][4]
	[2][0]	[2][1]	[2][2]	[2][3]	[2][4]
	[3][0]	[3][1]	[3][2]	[3][3]	[3][4]

Getting input in a two dimensional array

```
for ( row = 0; row < 4; row ++ )  
{  
    for ( column = 0; column < 5; column + + )  
    {  
        scores [ row ] [ column ] = Console.readInt ( "Enter score " + column + "for contestant " + row );  
    }  
}
```


Example 3 – Sum of two arrays

```
import java.util.Scanner;
class AddTwoMatrix
{
    public static void main(String args[])
    {
        int m, n, c, d;
        Scanner in = new Scanner(System.in);
        System.out.println("Enter the number of rows and
columns of matrix");
        m = in.nextInt();
        n = in.nextInt();
        int first[][] = new int[m][n];
        int second[][] = new int[m][n];
        int sum[][] = new int[m][n];
        System.out.println("Enter the elements of first matrix");
        for ( c = 0 ; c < m ; c++ )
            for ( d = 0 ; d < n ; d++ )
                first[c][d] = in.nextInt();
```

```
        System.out.println("Enter the elements of second matrix");
            for ( c = 0 ; c < m ; c++ )
                for ( d = 0 ; d < n ; d++ )
                    second[c][d] = in.nextInt();
        for ( c = 0 ; c < m ; c++ )
            for ( d = 0 ; d < n ; d++ )
                sum[c][d] = first[c][d] + second[c][d]; //replace '+'
with '-' to subtract matrices
        System.out.println("Sum of entered matrices:-");
        for ( c = 0 ; c < m ; c++ )
        {
            for ( d = 0 ; d < n ; d++ )
                System.out.print(sum[c][d]+"\\t");
            System.out.println();
        }
    }
}
```

Exercise

- 1) Create two matrices like example 3 and perform matrix multiplication
- 2) Create a 2 Dimensional matrix and perform transpose operation