



5. Arrays

Sisoft Technologies Pvt Ltd
SRC E7, Shipra Riviera Bazar, Gyan Khand-3, Indirapuram, Ghaziabad
Website: www.sisoft.in Email: info@sisoft.in
Phone: +91-9999-283-283

Array

- ◆ Arrays are used typically to group objects of the same type
- ◆ These group of objects are referred by a common name
- ◆ Arrays can be declared of any type either primitive or class.
- ◆ The declaration of an array creates space for a reference.
- ◆ Actual memory allocation is done dynamically either by a new statement or by an array initializer.

Declaring Array Variable

- Use an array into a program, you should declare a variable to reference the array, and you should specify the type of array the variable may reference. Here is the syntax for declaring an array variable.

```
dataType[] arrayRefVariable; // preferred way.  
or  
dataType arrayRefVariable[]; // works but not preferred way.
```

Example:

The following code chunks are examples of this syntax:

```
double[] myList; // preferred way.  
or  
double myList[]; // works but not preferred way.
```

Creating Array Variable Continue...

- You can build an array by using the new operator with the following given syntax:

```
arrayRefVariable = new dataType[arraySize];
```

The statement does the two things:

1. It makes an array using new dataType[ArraySize];
 2. It assigns the reference of the freshly made array to the variable arrayRefVariable.
- Declaring an array variable, making an array, and assigning the reference of the array to the variable may be combined in one statement, as represented :

```
dataType[] arrayRefVariable = new dataType[ArraySize];
```

Creating Array Variable Continue...

- Alternatively you may make an arrays as follows:

```
dataType[] arrayRefVariable = {value0, value1, ..., valuek};
```

- The array elements are reached through an index. An array indices are 0-Based. That is, they start from the 0 to arrayRefVariable.length-n.

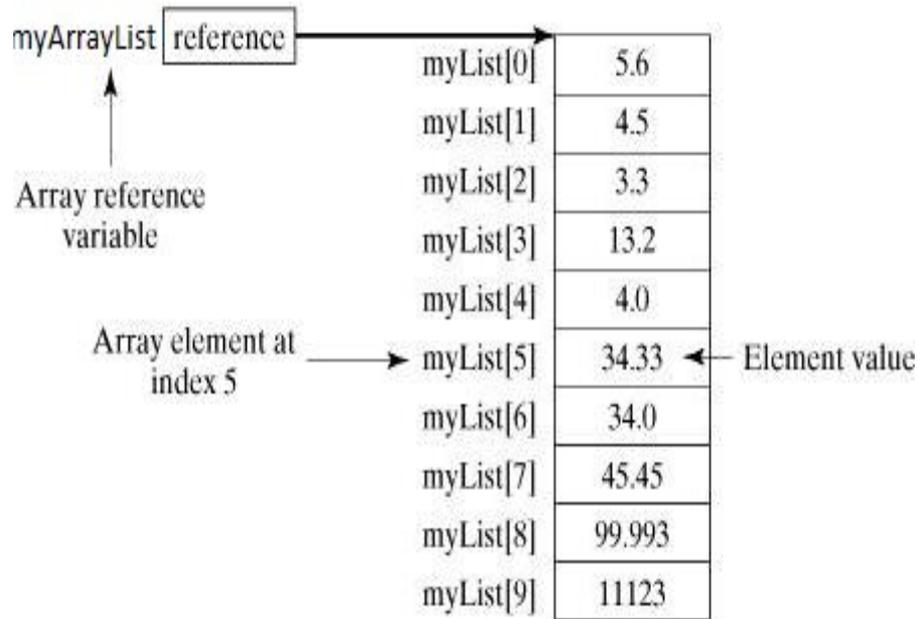
Example

- The following statement declares an array variable, myArrayList, makes an array of 10 elements of double type, and assigns its reference to myArrayList.

```
double[] myArrayList = new double[10];
```

Array Variable Continue...

- The following picture shows array myArrayList. Here, myArrayList contains 10 double-values and an indices are from 0 to 9.



Processing Array Variable Continue...

- The number of elements in an array is stored as part of the array object in the `length` attribute.
- When the processing array elements, we often use either the `for` loop or the `foreach` loop because all of the elements into an array are of the similar type, and the size of the array is known.

Example:

```
public class TestArray {
    public static void main(String[] args) {
        double[] myArrayList = {1.9, 2.9, 3.4, 3.5};

        // Print all the Array Elements
        for (int i = 0; i < myArrayList.length; i++) {
            System.out.println(myArrayList[i] + " ");
        }
    }
}
```

Processing Array Variable Continue...

```
// Summing all the Elements
    double total = 0;
    for (int i = 0; i < myArrayList.length; i++) {
        total += myArrayList[i];
    }
    System.out.println("Total is : " + total);
    // Finding the Largest Element
    double max = myArrayList[0];
    for (int i = 1; i < myArrayList.length; i++) {
        if (myArrayList[i] > max) max = myArrayList[i];
    }
    System.out.println("Max is : " + max);
}
}
```

Processing Array Variable Continue...

- The output of the preceding code is shown in the below:

```
1.9  
2.9  
3.4  
3.5  
Total is : 11.7  
Max is : 3.5
```

Processing Array Variable Continue...

- The foreach loops
 - The JDK 1.5 introduced the new for loop, known as enhanced for loop or the foreach loop, which capable you to traverse the complete array sequentially without an index variable.

Example

The following code shows all the elements in the array
myArrayList:

```
•  
• public class Test_Array {  
•     public static void main(String[] args) {  
•         double[] myArrayList = {1.9, 2.9, 3.4, 3.5};  
•         // Print all the Array Elements  
•         for (double ele: myArrayList) {  
•             System.out.println(ele);  
•         }  
•     }  
• }  
• }
```

Processing Array Variable Continue...

- The output of the preceding code is shown in the below:

```
1.9  
2.9  
3.4  
3.5
```

Arrays (Multi-Dimensional)

Declaration of Two Dimensional Array

```
dataType[] [] arrayRefVariable
```

Assign Range to Two Dimensional Array

```
varName = new type[size1][size2];
```

```
fa = new float[2][3];
```

Arrays (Two Dimensional)

Number of elements in Multi-Dimensional Array

`varName.length` ← **Number of Rows**

Number of elements in Each elements of Multi-Dimensional Array

`varName[index].length`

Initialization of Multi-Dimensional Array

```
Type varName[][] = {{e00, ... , e0n}, {e10,...,e1y}, {e20, ..., e2z}};
```

Processing Two Dimensional Array: for loop

Example:

```
int [][] myArray = new int [8][6];
for (int row=0; row < myArray.length; row++)
{
    for (int col=0; col < myArray[row].length;
        col++)
    {
        myArray[row][col] = row*col;
    }
}
```

Processing Two Dimensional Array: for-each loop

Example:

```
int [][] myArray = new int [8][6];
for (int[] x: myArray)    {
    for (int y : x)
    {
        System.out.println( y +" " );
    }
}
```

Array Resizing

- After an array is created, it cannot be resized.
- The same reference variable is used to refer to an entirely new array.
- The following code snippet displays a reference variable being used for a new array:
 - `int[] myArray = new int[6];`
 - `myArray = new int[10];`

Exercise

1. Write a program to sort an array in ascending order.
2. Write a program to sum a two matrices of 5 by 5 and display the result in the third one.
3. Write a program to display the sum of the diagonals of the 6 by 6 matrix.