Java - 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.

The first element of array start with zero.

Using an array in your program is a 3 step process:

1. Declaring you array (variable).
2. Constructing your array (size).
3. Initializing your array (values).

All elements or items in an array must be the same data type. For example Strings and integers cannot co-exist in one array. 1st item is index zero.

Code	Notes
`int [] myVariable;`	1st way to declare array
`int myVariable2 [];`	2nd way to declare array
`int [] myArray3 = new int[15];`	assign array size (15 elements or items) of the type integer
`myArray3[0] = 3;`	value of 3 is assigned to 1st item index 0 (zero)
`myArray3[1] = 7;`	value of 7 is assigned to 2nditem (index 1)
`int[] myArray = {1,2,3,4};`	declares array, size and values all in one line
`int len = myArray.length;`	.length gives the # of spaces allocated for the array
`int my2dArray [] [];`	declaring a 2 dimensional array

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

    // Print all the array elements
    for (int i = 0; i < myList.length; i++) {
      System.out.println(myList[i] + " ");
    }

    // Summing all elements
    double total = 0;
    for (int i = 0; i < myList.length; i++) {
      total += myList[i];
    }
    System.out.println("Total is " + total);

    // Finding the largest element
    double max = myList[0];
    for (int i = 1; i < myList.length; i++) {
      if (myList[i] > max) max = myList[i];
    }
    System.out.println("Max is " + max);
  }
}
```

Using for loops to access the array elements. Note that the variable “i” is used as the index for accessing the values in the array. This is similar to the “List” in python.

First item in the array is always index 0 (zero).

or   total = total + myList[i];

<u>Declaring Array:-</u>

<u>Syntax:</u>

```
elementType[] arrayName; Or elementType arrayName[];
```

<u>Example:</u>

```
int[] intArray; int intArray[];
```

Constructing Array:-
Syntax:

```
new elementType[size];
```

Example:

```
int[] intArray = new int[10]; // Defines that intArray will store 10 integer values int
intArray[] = new int[10];
```

Initializing Array:-
Syntax:

```
arrayName[element 0,1,2?.. N] = value;
```

Example:

```
intArray[0] = 10; // Assign an integer value 10 to the first element 0 of the array
intArray[1] = 20;
```

Declaring and Initializing Array:-
Syntax:

```
elementType[] arrayName = {values1,values2,? valueN};
```

Example:

```
int[] intArray = {1,2,3,4};
```

Array Example:

```
public class Main {
public static void main(String[] args) {
  String[] names = new String[3];
  names[0] = "A";     names[1] = "ABC";   names[2] = "XYZ";
  for (int i = 0; i < names.length; i++) {
    System.out.println(names[i]);
  }
  //this line should throw an exception
  //System.out.println(names[6]);
}
}
```

Array are passed by reference:

Arrays are passed to functions by reference, or as a pointer to the original. This means anything you do to the Array inside the function affects the original.

Example:

```
public class Main {

public static void passByRefrence(String a[]) {

  a[0] = "Z";

}

public static void main(String args[]) {

  String[] b = {"A","B","C"};

  System.out.println("Before Function call : "+b[0]);

  Main.passByRefrence(b);

  System.out.println("After Function call : "+b[0]);

}

}
```

Output:
Before Function call : A
After Function call : Z

Multidimensional Arrays:

Multidimensional arrays, are arrays of arrays.

Syntax:

```
elementType[][] arrayName = new elementType[size][size];
```

Example:

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

When you allocate memory for a multidimensional array, you need only specify the memory for the first (leftmost) dimension.

You can allocate the remaining dimensions separately.

In Java the length of each array in a multidimensional array is under your control.

Example:

```
int multi[][] = new int[2][]; multi[0] = new int[5]; multi[1] = new int[4];
```

Array of Objects:

It is possible to create array of objects of user created class.

Example:

```
class Employee{ int id; String name; public void setData(int id, String name){ this.id = id;
this.name = name; } public void displayData(){ System.out.println("Employee ID : "+this.id);
System.out.println("Employee Name : "+this.name); } } class Main{ public static void
main(String args[]){ Employee[] emp = new Employee[2]; emp[0].setData(1,"ABC");
emp[1].setData(2,"XYZ"); emp[0].displayData(); emp[1].displayData(); } }
```

Output:

```
Employee ID : 1
Employee Name : ABC
Employee ID : 2
Employee Name : XYZ
```