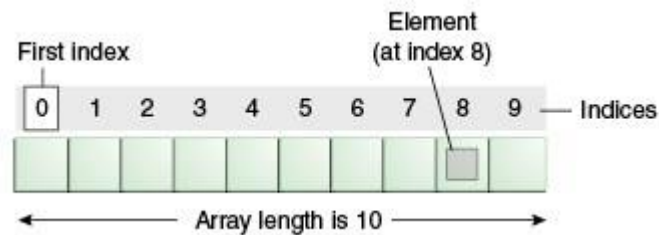


Lesson on Arrays in Java

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

Element and index



Each item in an array is called an **element**, and each element is accessed by its numerical **index**. As shown in the preceding illustration, numbering begins with 0. The 9th element, for example, would therefore be accessed at index 8.

The following program, ArrayDemo, creates an array of integers, puts some values in the array, and prints each value to standard output.

Class that makes use of an array

```
1 class ArrayDemo
2 {
3     public static void main(String[] args)
4     {
5         // declares an array of integers
6         int[] anArray;
7         // allocates memory for 10 integers
8         anArray = new int[10];
9
10        // initialize first element
11        anArray[0] = 100;
12        // initialize second element
13        anArray[1] = 200;
14        // and so forth
15        anArray[2] = 300;
16        anArray[3] = 400;
```

```

17     anArray[4] = 500;
18     anArray[5] = 600;
19     anArray[6] = 700;
20     anArray[7] = 800;
21     anArray[8] = 900;
22     anArray[9] = 1000;
23
24     System.out.println ("Element at index 0: " + anArray[0] );
25     System.out.println( "Element at index 1: " + anArray[1] );
26     System.out.println( "Element at index 2: " + anArray[2] );
27     System.out.println( "Element at index 3: " + anArray[3] );
28     System.out.println( "Element at index 4: " + anArray[4] );
29     System.out.println( "Element at index 5: " + anArray[5] );
30     System.out.println( "Element at index 6: " + anArray[6] );
31     System.out.println( "Element at index 7: " + anArray[7] );
32     System.out.println( "Element at index 8: " + anArray[8] );
33     System.out.println( "Element at index 9: " + anArray[9] );
34 }
35 }

```

The output from this program is:

```

Element at index 0: 100
Element at index 1: 200
Element at index 2: 300
Element at index 3: 400
Element at index 4: 500
Element at index 5: 600
Element at index 6: 700
Element at index 7: 800
Element at index 8: 900
Element at index 9: 1000

```

Lines 6 and 8 can be combined in into one as shown:

```
int[] anArray = new int[10];
```

You can declare arrays of other types e.g.:

```
byte[] anArrayOfBytes;
double[] anArrayOfDoubles;
```

```
boolean[] anArrayOfBooleans;  
char[] anArrayOfCharacters;  
String[] anArrayOfStrings;
```

Creating, Initializing, and Accessing an Array

You can create an array by listing its elements as shown here:

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

Here the length of the array is determined by the number of values provided between braces and separated by commas.

The array that we saw in the example is called a **one-dimensional array** (you can have arrays that are two-dimensional, three-dimensional etc).

Example: average age

Class that uses arrays to find average age

```
1 public class Arrays  
2 {  
3     static String names [ ] = new String [5];  
4     static int ages [ ] = new int [5];  
5  
6     public static void main (String args[])  
7     {  
8         int aveAge;  
9  
10        enterValues();  
11        aveAge = averageAge();  
12        System.out.println ("The average age is " + aveAge);  
13    }  
14  
15    public static void enterValues ()  
16    {  
17        for (int i=0; i<5; i++)  
18        {  
19            System.out.print ("Enter name: ");  
20            names[i] = Keyboard.readString();  
21            System.out.print ("Enter age of " + names[i] + ": ");  
            ages[i] = Keyboard.readInt();  
        }  
    }  
}
```

```

22     }
23 }
24
25 public static int averageAge ()
26 {
27     int ave;
28     int tot = 0;
29
30     for (int i = 0; i<5; i++)
31         tot = tot + ages[i];
32
33     ave = tot / 5;
34
35     return ave;
    }
}

```

Exercise

1. Consider the array below and then answer the questions.

	0	1	2	3	4	5
numbers	78	-32	9	66	-85	12

- a. What is the value of numbers [3]?
- b. What is the value of 4*numbers [2] + 8?
- c. What is the value of 5*numbers [2 + 3]?
- d. How will the array change after the following modifications?
 - i) numbers[3] = 7;
 - ii) numbers[0] = numbers[2] - numbers[5]
 - iii) for (i=0, i<5, i++) numbers[i] = numbers[i]*2 - i;

2. Look at the following array and answer the questions below.

```

String [ ] skiResorts = {
    "Whistler Blackcomb", "Squaw Valley", "Brighton",
    "Snowmass", "Sun Valley", "Taos"
};

```

- a. What is the index of Brighton?
- b. Write an expression that refers to the string “Brighton” within the array.
- c. What is the index of the last item in the array?
- d. What is the value of the expression `skiResorts [4]`?