

Java – Iterate over Array Elements

Java – Iterate over Array Elements

We know that in Java, Array is a collection of similar type of elements which are stored in an order and can be accessed using an incremental index. Also, it is required that we could traverse or iterate through elements of this array.

To iterate over elements of an array, you can use looping statements like for loop, while loop or enhanced for loop.

In this tutorial, we will go through some of the examples where we take an array with some elements in it, and traverse through those elements.

Example – Iterate over Array Elements – For Loop

In this example, we declare and initialize an integer array `nums` with some elements, to be precise six elements. What we shall do is, employ a for loop. The for loop has an `index` variable starting with an initial value of zero and increment in steps of one until we reach end of the array.

As the index of the array also ranges from zero until its length, we shall use the variable `index` to access elements of the array, one by one.

Java Program

```
/**
 * Java Example Program to iterate over Array Elements
 */
public class IterateArray {
    public static void main(String[] args) {
        int[] nums = {25, 86, 41, 97, 22, 34};
        for(int index = 0; index < nums.length; index++) {
            //access each element using index
            int num = nums[index];
            System.out.println(num);
        }
    }
}
```

Output

Compile and run the above program in command prompt or terminal. Or you may use your favorite IDE which does this job in just a click of run button. You should see the elements printed to the console as shown below.

```
25
86
41
97
22
34
```

In each execution of for loop body, we got access to an element, and we printed that element to standard output.

Example – Iterate over Array Elements – Enhanced For Loop

In this example, we are using enhanced for loop to iterate through array elements. The for loop is enhanced, because we do not initialize some index, nor update it, nor use the index to get the element from array. All of that is encapsulated in a single statement of `int num:nums` .

Java Program

```
/**
 * Java Example Program to iterate over Array Elements
 */
public class IterateArray {
    public static void main(String[] args) {
        int[] nums = {25, 86, 41, 97, 22, 34};
        for(int num: nums) {
            //access each element
            System.out.println(num);
        }
    }
}
```

Output

```
25
86
41
97
22
34
```

We traversed through all the elements of array.

Example – Iterate over Array Elements – While Loop

We know that while loop is some what primitive than for loop. If you look at the following program, it is more or

less same as that of first example. Just take care the while loop statement's syntax. And you should get access to each array element one at a time in the loop.

Java Program

```
/**
 * Java Example Program to iterate over Array Elements
 */
public class IterateArray {
    public static void main(String[] args) {
        int[] nums = {25, 86, 41, 97, 22, 34};
        int index=0;
        while(index<nums.length) {
            //access each element in array
            int num = nums[index];
            System.out.println(num);
            index++;
        }
    }
}
```

Output

```
25
86
41
97
22
34
```

Conclusion

Concluding this topic, in this [Java Tutorial](#), we learned how to traverse or iterate over the array elements in Java programming, with the help of example programs.

- ◆ [Java Tutorial](#)
- ◆ [Java Array](#)
- ◆ [Java Array - Print](#)
- ◆ [Java Array - Initialize](#)
- ◆ [Java Array of Integers](#)
- ◆ [Java Array of Strings](#)
- ◆ [Java Array of Objects](#)
- ◆ [Java Array of Arrays](#)

⇒ **Java Array - Iterate over Items**

◆ [Java Array - For Loop](#)

◆ [Java Array - While Loop](#)

◆ [Java Array Append](#)

◆ [Java Array - Check if Empty](#)

◆ [Java Array Average](#)

◆ [Java Array - Contains](#)

◆ [Java Array - ForEach](#)

◆ [Java Array - Find Index of Item](#)

◆ [Java Array Sum](#)

◆ [Java Concatenate Arrays](#)

◆ [Java Array - Find Smallest Number](#)

◆ [Java Array - Find Largest Number](#)

◆ [Java Array - Reverse](#)