



Java Iterator

[< Previous](#)[Next >](#)

Java Iterator

An **Iterator** is an object that can be used to loop through collections, like [ArrayList](#) and [HashSet](#). It is called an "iterator" because "iterating" is the technical term for looping.

To use an Iterator, you must import it from the **java.util** package.

Getting an Iterator

The **iterator()** method can be used to get an **Iterator** for any collection:

Example

```
// Import the ArrayList class and the Iterator class
import java.util.ArrayList;
import java.util.Iterator;

public class Main {
    public static void main(String[] args) {

        // Make a collection
```

```
ArrayList<String> cars = new ArrayList<String>();  
cars.add("Volvo");  
cars.add("BMW");  
cars.add("Ford");  
cars.add("Mazda");  
  
// Get the iterator  
Iterator<String> it = cars.iterator();  
  
// Print the first item  
System.out.println(it.next());  
}  
}
```

Try it Yourself »

Looping Through a Collection

To loop through a collection, use the `hasNext()` and `next()` methods of the `Iterator` :

Example

```
while(it.hasNext()) {  
    System.out.println(it.next());  
}
```

Try it Yourself »

ADVERTISEMENT

Removing Items from a Collection

Iterators are designed to easily change the collections that they loop through. The `remove()` method can remove items from a collection while looping.

Example

Use an iterator to remove numbers less than 10 from a collection:

```
import java.util.ArrayList;
import java.util.Iterator;

public class Main {
    public static void main(String[] args) {
        ArrayList<Integer> numbers = new ArrayList<Integer>();
        numbers.add(12);
        numbers.add(8);
        numbers.add(2);
        numbers.add(23);
        Iterator<Integer> it = numbers.iterator();
        while(it.hasNext()) {
            Integer i = it.next();
            if(i < 10) {
                it.remove();
            }
        }
    }
}
```

```
        System.out.println(numbers);  
    }  
}
```

[Try it Yourself »](#)

Note: Trying to remove items using a **for loop** or a **for-each loop** would not work correctly because the collection is changing size at the same time that the code is trying to loop.

[◀ Previous](#)

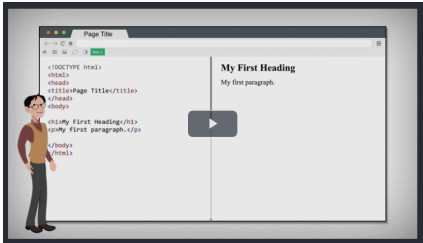
[Next ▶](#)

ADVERTISEMENT



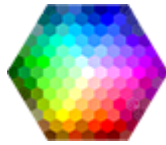
NEW

We just launched
W3Schools videos



Explore now

COLOR PICKER



Get certified
by completing
a Java
course today!



Get started

CODE GAME



Play Game

ADVERTISEMENT

ADVERTISEMENT

ADVERTISEMENT

[Report Error](#)

[Spaces](#)

[Pro](#)

[Buy Certificate](#)

Top Tutorials

[HTML Tutorial](#)
[CSS Tutorial](#)
[JavaScript Tutorial](#)
[How To Tutorial](#)
[SQL Tutorial](#)
[Python Tutorial](#)
[W3.CSS Tutorial](#)
[Bootstrap Tutorial](#)
[PHP Tutorial](#)
[Java Tutorial](#)
[C++ Tutorial](#)
[jQuery Tutorial](#)

Top References

[HTML Reference](#)
[CSS Reference](#)
[JavaScript Reference](#)
[SQL Reference](#)
[Python Reference](#)
[W3.CSS Reference](#)
[Bootstrap Reference](#)
[PHP Reference](#)
[HTML Colors](#)
[Java Reference](#)
[Angular Reference](#)
[jQuery Reference](#)

Top Examples

[HTML Examples](#)
[CSS Examples](#)
[JavaScript Examples](#)
[How To Examples](#)
[SQL Examples](#)
[Python Examples](#)
[W3.CSS Examples](#)
[Bootstrap Examples](#)
[PHP Examples](#)

Java Examples
XML Examples
jQuery Examples

Get Certified

HTML Certificate
CSS Certificate
JavaScript Certificate
Front End Certificate
SQL Certificate
Python Certificate
PHP Certificate
jQuery Certificate
Java Certificate
C++ Certificate
C# Certificate
XML Certificate

[FORUM](#) | [ABOUT](#)

W3Schools is optimized for learning and training. Examples might be simplified to improve reading and learning. Tutorials, references, and examples are constantly reviewed to avoid errors, but we cannot warrant full correctness of all content. While using W3Schools, you agree to have read and accepted our terms of use, cookie and privacy policy.

Copyright 1999-2022 by Refsnes Data. All Rights Reserved.
W3Schools is Powered by W3.CSS.

