

Iterators and for-each Loops

- Consider the interface `Iterable<T>` *← some class type*

```
package java.lang;
import java.util.Iterator;
/**/
public interface Iterable<T> {
    /**/
    Iterator<T> iterator();
}
```

A class which implements
`Iterable <T>` can use the
for-each loop syntax

- and the interface `Iterator<E>` *← some class type*

```
package java.util;
import java.util.function.Consumer;
/**/
public interface Iterator<E> {
    /**/
    default void forEachRemaining(Consumer<? super E> action);
    boolean hasNext();
    E next();
    default void remove();
}
```

default means that java
supplies a default implementation
for these two methods, i.e., it is
optional to code them in the
implementing class

- Here is a list of the public methods of `List<T>`.
`List <T>` was written by Mr. Harren.
It is similar to `ArrayList<T>`, but simpler.

```
import java.util.Iterator;
/*
 * A generic list of Objects of type T --
 *
 * Note that the interface Iterable allows
 * for the use of the for-each loop syntax
 */
public class List<T> implements Iterable<T>, Iterator<T> {
    /*
     * Create a List with initial capacity of BLOCK_SIZE = 32
     */
    public List ();
    /*
     * Add element e to the end of the list
     */
    public void add ( T e );
    /*
     * Get the element at index i
     */
    public T get ( int i );
    /*
     * Return the current length
     */
    public int length ();
    /*
     * As per interface Iterable
     */
    public Iterator<T> iterator();
    /*
     * As per interface Iterator
     */
    public boolean hasNext ();
    /*
     * As per interface Iterator
     */
    public T next ();
}
```

● for loop syntax

```
List<String> stringList=new List<String> ();
/*
 * code (not shown) which defines the entries of stringList
 */
String s;
int len=stringList.length();
for ( int i=0; i<len; ++i ) {
    s=stringList.get(i);
    /*
     * more code (not shown) which uses String s
     */
}
```

● while loop syntax

```
import java.util.Iterator;
/*
 * whatever code (not shown) which may be here
 */
List<String> stringList=new List<String> ();
/*
 * code (not shown) which defines the entries of stringList
 */
String s;
Iterator<String> stringIterator=stringList.iterator();
while ( stringIterator.hasNext() ) {
    s=stringIterator.next();
    /*
     * more code (not shown) which uses String s
     */
}
```

● for-each loop syntax (this loop prints the entries of stringList to the console)

```
List<String> stringList=new List<String> ();
/*
 * code (not shown) which defines the entries of stringList
 */
for ( String s : stringList ) {
    System.out.println(s);
}
```

← "for each String s in stringList"

● for-each loops also work with arrays

Example: The output of the code segment

```
int [] ia = new int [] { 5 , 4 , 3 , 2 , 1 };
for ( int i : ia ) {
    System.out.print(i);
}
```

is

54321