

## Section Thirty Three – Iterator

---



In this section the Developer will learn...

- How to loop through elements using Iterator



## Iterator – Introduction

---

Iterator is the replacement class for Enumeration and although very similar has an additional method which allows the Developer to remove the value after use.

The three methods the Iterator has are ...

*hasNext()*  
*next()*  
*remove()*

Make sure to import the package before creating an Iterator

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



## Iterator – Using Iterator

---

Simple looping with Iterator can be achieved by populating it with an ArrayList ...

```
ArrayList<String> textlist = new ArrayList<String>();  
textlist.add("A");  
textlist.add("B");  
textlist.add("C");  
textlist.add("D");  
textlist.add("E");
```

```
Iterator<String> looper = textlist.iterator();
```

```
while(looper.hasNext())  
    {  
        System.out.print(looper.next() + " ");  
    }
```

## Iterator – Using Iterator with remove()

---



The `remove()` method will allow the Developer to use the value then erase it ...

```
ArrayList<String> textlist = new ArrayList<String>();  
textlist.add("A");  
textlist.add("B");  
textlist.add("C");  
textlist.add("D");  
textlist.add("E");
```

```
Iterator<String> looper = textlist.iterator();
```

```
while(looper.hasNext())  
    {  
        System.out.println(looper.next() + " ");  
        looper.remove(); // Must follow a next()  
    }
```



## Iterator – Using ListIterator

---

A sub-class of Iterator is ListIterator, this inherits the previous methods but has a few additional ones, one of which is set() which allows the amendment of the current value in the loop ...

```
ArrayList<Integer> textlist = new ArrayList<Integer>();  
textlist.add(1);
```

...

```
ListIterator<Integer> looper = textlist.listIterator();  
while(looper.hasNext())  
    {  
        looper.set(looper.next() + 10);  
    }
```

```
System.out.println(textlist);
```

```
[11, 12, 13, 14, 15]
```