Difference between Iterator<E> and ListIterator<E> interface

We have seen [Iterator<E>](http://data-structure-learning.blogspot.com/2015/05/java-collections-part-8-iterator.html) and [ListIterator<E>](http://data-structure-learning.blogspot.com/2015/05/java-collections-part-9-listiterator.html) interface and how to use them. I would strongly recommend going through both of posts of Iterator<E> and ListIterator<E>.

Iterator is used to iterate over [List<E>](http://data-structure-learning.blogspot.com/2015/05/java-collections-part-5list-interface.html), Set<E> and [Queue<E>](http://data-structure-learning.blogspot.com/2015/05/queue-interface.html). ListIterator can traverse through List<E> only.

Using Iterator<E> we can traverse in only forward direction. But using ListIterator<E> we can traverse through forward as well as backward direction.

Using ListIterator<E> we can obtain index of next and previous elements. Same is not possible with Iterator<E> interface.

Using ListIterator<E> we can perform modification operations such as set, remove and add operations. Iterator<E> has remove method to remove element from Collection.

Iterator<E> always starts from first index. ListIterator<E> can be started from specified index as listIterator() is overloaded method. Below are 2 overloaded methods of listIterator().

**public** ListIterator<E> listIterator() {

**return** **new** ListItr(0);

}

**public** ListIterator<E> listIterator(**int** index) {

**if** (index < 0 || index > size)

**throw** **new** IndexOutOfBoundsException("Index: "+index);

**return** **new** ListItr(index);

}