**Different ways of iterating an ArrayList**

Q**-There are how many ways are there we can Iterate are Array List?**

1. After Java JDK 1.8 we have Lambda Expression also.
2. There are five to six ways we can Iterate or we can print all the values from Array List.
3. It will help you while writing the code because in Selenium or in backend Automation also
4. So this is something very important we have to Iterate are Array List .

**Ways A) Using Java 8 with for each loop and lambda expression :**

Steps – to be considered ----

1. Create One Array List like which is String type . it means I will store only String values on it .

ArrayList <String> tvSeries = **new** ArrayList <String>();

So I have restricted my ArrayList with String value. It means I have provided the this particular List having only String value. It means given the Generic over here. [import – java.util package]

1. I will add couple of TV series using add().
2. Now How can we Iterate with the help of different ways ?

We can Iterate with the help of using

1. for each loop and lambda expression -- to do this things simple write

tvseries then dot forEach (one object like shows is there for each TV series then lambda Expression and then start the 2nd Breaket – then end 2nd breaket and end First breaket. -🡪 tvseries.forEach(shows -> {}); then give syso(shows); //whatever the shows value is there.

**here shows is presenting this particular tvSeries Element – it means shows is presenting One by One each and Every Element . So you can give any name here other than shows.**

1. **Then you will get all the values – See the Out put**

**public** **class** IteratingWaysArrayListDemo {

**public** **static** **void** main(String[] args) {

ArrayList <String> tvSeries = **new** ArrayList <String>();

tvSeries.add("Game of Thrones");

tvSeries.add("Breaking Bad");

tvSeries.add("The Big Bankg Theory");

tvSeries.add("The Walking Dead");

tvSeries.add("Preson Break");

// Using Java 8 with for each loop and lambda expression :

tvSeries.forEach(shows ->{

System.***out***.println(shows);

});

}}

**Out put**

Game of Thrones

Breaking Bad

The Big Bang Theory

The Walking Dead

Prison Break…..

Way Number B) Using Iterator : iterator () is there .

Steps –

1. Use Iterator () ,
2. Use tvSeries object reference 🡪 dot🡪 iterator() and
3. This iterator () , it will return one Iterator of String. So it will store Iterator String using generic.
4. Use while loop and this Iterator it. one method is there named hasNext()- It means if in this particular Iterator the next value is available. You have to keep executing this particular while loop. And it will move to the next value – it.next(). And this will return a string and then printed in the console.
5. The returned iterator is [*fail-fast*](https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html#fail-fast).

// Use Iterator :

Iterator <String> it =tvSeries.iterator();

**while** (it.hasNext()) {

String str = it.next();

System.***out***.println(str);

}

C - Using Iterator and Java 8 foreachRemaing()

**Steps**

1. You need to do Whatever tvseries Array List – then apply Iterator and what exactly it will return string and we can store in same ID -🡪 it =tvSeries.iterator();
2. And then with this particular Iterator it we will use foreachRemaining()
3. In this method I will put shows which is pointing to this particular Array List foreachRemaining(shows ->{}) and then lambda Expression and start 2nd bracket[it means starting lambda] – end 2nd bracket [ending lambda] and then End first bracket ---Then semicolon.

we can use any name or show [not shows because shows means ArrayList object.] which pointing to this particular Array List

System.***out***.println("===Print using Iterator and Java 8 foreachRemaing()======");

//Print using Iterator and Java 8 foreachRemaing()

it =tvSeries.iterator();

it.forEachRemaining(str1 -> {

System.***out***.println(str1);

}); // }we can use any name or show [not shows = Array List object]

===Print using Iterator and Java 8 foreachRemaing()======

Game of Thrones

Breaking Bad

The Big Bankg Theory

The Walking Dead

Prison Break

D Using Simple for each loop—

Step / Logic

1. Use for each loop and Data type of Array List then ref name then colon ---and then name of ArrayList. and starting and ending 2nd bracket.
2. So for(String show : tvSeries ) { syso}
3. I know that in my Array List all the values are string

// Simple using for each loop

System.***out***.println("===Print using for each Loop======");

**for**(String show : tvSeries) {

System.***out***.println(show);

}

**E Simple using for loop with order or Indexing**

// Simple using for loop with order or Indexing

System.***out***.println("===Print using for loop with order or Indexing======");

**for** (**int** i = 0; i < tvSeries.size(); i++) {

System.***out***.println(tvSeries.get(i));

}

**F Using List Iterator() to traverse in both the directions**

**Logic –**

1. We will start from end of the List and Traverse back ward
2. Simple whatever your tvSeries ArrayList and dot List Iterator () – it means What is the total size of your list. So total size is tvSeries.size() and it will return – List of Iterator String type – using Generic.
3. Then start a while loop – what ever the List Iterator is there -> ltr
4. You have to got the back ward directions, if it is having the previous value.

so use hasPrevious() and then come inside the while loop , simple with the help of same Iterator [ltr] and then dot 🡪 what is the previous value is there.

And store it in a particular String Value . and then print it .

it will return Boolean value

1. **Returns:**

true if the list iterator has more elements when traversing the list in the reverse direction.

//Using List Iterator() to traverse in both the directions

System.***out***.println("===Print Using List Iterator() to traverse in both the directions ======");

System.***out***.println("BackWard Direction : ");

ListIterator<String> ltr =tvSeries.listIterator(tvSeries.size());

**while** (ltr.hasPrevious()){

String str = ltr.previous();

System.***out***.println(str);

}

System.***out***.println();

System.***out***.println("ForWard Direction: ");

**while** (ltr.hasNext()){

String str = ltr.next();

System.***out***.println(str);

}

package NaveenCollections;

import java.util.ArrayList;

import java.util.Iterator;

import java.util.ListIterator;

public class IteratingWaysArrayListDemo {

public static void main(String[] args) {

ArrayList <String> tvSeries = new ArrayList <String>();

tvSeries.add("Game of Thrones");

tvSeries.add("Breaking Bad");

tvSeries.add("The Big Bankg Theory");

tvSeries.add("The Walking Dead");

tvSeries.add("Prison Break");

// Using Java 8 with for each loop and lambda expression :

System.out.println("=======Print for each with lambda========");

tvSeries.forEach(shows ->{

System.out.println(shows);

});

System.out.println("=======Print using Iterator ==========="); // we can see the Separate output.

// Use Iterator :

Iterator <String> it =tvSeries.iterator();

while (it.hasNext()) {

String str = it.next();

System.out.println(str);

}

System.out.println("===Print using Iterator and Java 8 lambda and foreachRemaing()======");

//Print using Iterator and Java 8 foreachRemaing()

it =tvSeries.iterator();

it.forEachRemaining(str1 -> {

System.out.println(str1);

}); // }we can use any name or show [not shows = Array List object]

// Simple using for each loop

System.out.println("===Print using for each Loop======");

for(String show : tvSeries) {

System.out.println(show);

}

// Simple using for loop with order or Indexing

System.out.println("===Print using for loop with order or Indexing======");

for (int i = 0; i < tvSeries.size(); i++) {

System.out.println(tvSeries.get(i));

}

System.out.println();

//Using List Iterator() to traverse in both the directions

System.out.println("===Print Using List Iterator() to traverse in both the directions ======");

System.out.println("BackWard Direction : ");

ListIterator<String> ltr =tvSeries.listIterator(tvSeries.size());

while (ltr.hasPrevious()){

String str = ltr.previous();

System.out.println(str);

}

System.out.println();

System.out.println("ForWard Direction: ");

while (ltr.hasNext()){

String str = ltr.next();

System.out.println(str);

}}}

OUTPUT

=======Print for each with lambda========

Game of Thrones

Breaking Bad

The Big Bankg Theory

The Walking Dead

Prison Break

=======Print using Iterator ===========

Game of Thrones

Breaking Bad

The Big Bankg Theory

The Walking Dead

Prison Break

===Print using Iterator and Java 8 lambda and foreachRemaing()======

Game of Thrones

Breaking Bad

The Big Bankg Theory

The Walking Dead

Prison Break

===Print using for each Loop======

Game of Thrones

Breaking Bad

The Big Bankg Theory

The Walking Dead

Prison Break

===Print using for loop with order or Indexing======

Game of Thrones

Breaking Bad

The Big Bankg Theory

The Walking Dead

Prison Break

===Print Using List Iterator() to traverse in both the directions ======

BackWard Direction :

Prison Break

The Walking Dead

The Big Bankg Theory

Breaking Bad

Game of Thrones

ForWard Direction:

Game of Thrones

Breaking Bad

The Big Bankg Theory

The Walking Dead

Prison Break