

ArrayBuffers

In the following lesson, you will learn about ArrayBuffers and how they differ from ordinary arrays.

We'll cover the following



- Introduction
- Creating an ArrayBuffer
- Adding Elements
- Deleting Elements
 - -= Assignment Operator
 - remove method
 - clear method

Introduction

An **ArrayBuffer** in Scala is a collection that comes under the sequence class. Like Arrays, it is a mutable collection, hence, when elements in an ArrayBuffer are modified, the original ArrayBuffer is updated.

ArrayBuffers are very similar to arrays with the difference that you can add and remove elements from an ArrayBuffer while adding and removing elements is not possible in simple Arrays.

Array methods and operations are also available for ArrayBuffers.

Creating an ArrayBuffer

To be able to use an ArrayBuffer, we first need to import it using the package `scala.collection.mutable.ArrayBuffer`. Afterward, you can create and populate an ArrayBuffer the same way you create an Array.

This code requires the following environment variables to execute:



LANG

C.UTF-8

```
import scala.collection.mutable.ArrayBuffer

val myFirstArrayBuffer = ArrayBuffer(1,2,3,4,5)

// Driver Code
myFirstArrayBuffer.foreach(println)
```



To initialize an `ArrayBuffer` without populating it, we need to specify the type of element it can store. Mentioning the length is not required as the `ArrayBuffer` will adjust the allocated space automatically when you insert or delete elements.

```
val arrayBufferName = new ArrayBuffer[DataType] ()
```

```
val newArrayBuff = new ArrayBuffer[Int]()
```

Adding Elements

Let's now populate `newArrayBuffer`. To add an element to an `ArrayBuffer` you use the assignment operator `+=` which appends the specified element to the end of an `ArrayBuffer`.

This code requires the following environment variables to execute:

LANG C.UTF-8

```
import scala.collection.mutable.ArrayBuffer
val newArrayBuff = new ArrayBuffer[Int]()

newArrayBuff += 6
newArrayBuff += 15
newArrayBuff += 78
newArrayBuff += 4
newArrayBuff += 32
newArrayBuff += 11

// Driver Code
newArrayBuff.foreach(println)
```



Deleting Elements

There are multiple ways to delete an element from an `ArrayBuffer`. Let's look at the ones which are most commonly used.

For all the examples below, we will be working with `newArrayBuff`. Remember that our original `ArrayBuffer` contains the elements `[6, 15, 78, 4, 32, 11]`.

`--` Assignment Operator `#`

The same way we add an element using `+=`, we can remove a specific element using `--`.

This code requires the following environment variables to execute:

LANG C.UTF-8

```
import scala.collection.mutable.ArrayBuffer
val newArrayBuff = new ArrayBuffer[Int]()

newArrayBuff += 6
newArrayBuff += 15
newArrayBuff += 78
newArrayBuff += 4
newArrayBuff += 32
newArrayBuff += 11

newArrayBuff -- 78 //Remove the element with the value 78

// Driver Code
newArrayBuff.foreach(println)
```

Our updated `ArrayBuffer` is now `[6, 15, 4, 32, 11]`.

`remove` method `#`

If we want to remove an element at a specific index, we can use the `remove` method.

This code requires the following environment variables to execute:

LANG C.UTF-8

```
import scala.collection.mutable.ArrayBuffer
val newArrayBuff = new ArrayBuffer[Int]()

newArrayBuff += 6
newArrayBuff += 15
```

```

newArrayBuff += 15
newArrayBuff += 78
newArrayBuff += 4

newArrayBuff += 32
newArrayBuff += 11

newArrayBuff -= 78

newArrayBuff.remove(4) //Remove the element at the 4th index

// Driver Code
newArrayBuff.foreach(println)

```



Our updated ArrayBuffer is now [6, 15, 4, 32]

clear method

If we want to remove all the elements from an ArrayBuffer, we can use the **clear** method.

This code requires the following environment variables to execute:

LANG	C.UTF-8
------	---------

```

import scala.collection.mutable.ArrayBuffer
val newArrayBuff = new ArrayBuffer[Int]()

newArrayBuff += 6
newArrayBuff += 15
newArrayBuff += 78
newArrayBuff += 4
newArrayBuff += 32
newArrayBuff += 11

newArrayBuff -= 78

newArrayBuff.remove(4)

newArrayBuff.clear() //Remove all the elements from newArrayBuff

// Driver Code
newArrayBuff.foreach(println)

```



In the code widget above, no output is displayed. This is because all the elements from **newArrayBuff** have been removed.

With this, our discussion on Arrays and ArrayBuffers comes to an end. In the next

With this, our discussion on Arrays and ArrayBuilders comes to an end. In the next lesson, you will be challenged to create and populate an array of your own.