Solution Review: Array of Squares

In the following lesson, we will go over the solution of the challenge: Array of Squares.

> **We'll cover the following** ⌃
>
> - Task
> - Solution

Task

In this challenge, you were provided an array of numbers which needed to be squared.

Solution

Let's go over the solution step-by-step.

- The first thing you had to figure out is that this problem requires a `for` expression.

- Next, you had to figure out the number of iterations required for the program to return the correct output. As each element of the array needed to be squared, the number of times the `for` expression should run is equivalent to the length of the array.

```
for (i <- 0 until testArray.length)
```

> To create our range for the iterator `i` we use the `Range` keyword `until`. This is because the last index of an array is one less than its length as indexes start from **0** and length starts from **1**. As we already know from a previous lesson, `until` excludes the upper limit.

- Finally, the last step required you to assign the square of the value at an index to the same index.

```
testArray(i) = testArray(i)*testArray(i)
```

You can find the complete solution below:

> You were required to write the code from **line 3** to **line 5**.

This code requires the following environment variables to execute:

LANG   C.UTF-8

```scala
val testArray = Array(1,2,3)

for (i <- 0 until testArray.length) {
  testArray(i) = testArray(i) * testArray(i)
}

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

In the next lesson, we will learn about the `try` control structure.