Immutable Variables

In the following lesson, you will learn how to declare an immutable variable.

We'll cover the following ∧

- Declaring an Immutable Variable

Immutable is defined as *unchangeable* and is precisely what an immutable variable is; unchangeable.
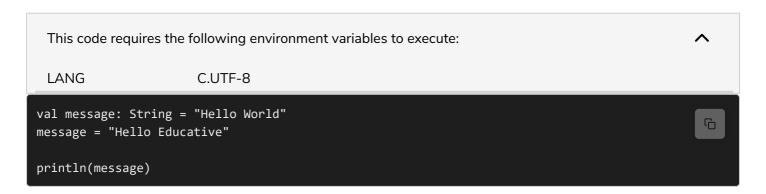
Immutable variables are basically like constants; once they are assigned a value, that value can never change.

Declaring an Immutable Variable

To declare an immutable variable, we use the `val` keyword. Let's create a variable named `message` that is assigned a `String` value.

This code requires the following environment variables to execute: ∧

LANG   C.UTF-8

```
val message: String = "Hello World"

println(message)
```

Oh no! You just realized you sent the wrong message. Let's try to *reassign* a value to the variable `message`.

This code requires the following environment variables to execute: ∧

LANG   C.UTF-8

```
val message: String = "Hello World"
message = "Hello Educative"

println(message)
```

The above code would give you a runtime error `error: reassignment to val` letting you know that `val` cannot be reassigned a value.

All Scala variables are immutable by default and it is also the preferred variable form.

Now, let's move on to mutable variables that *can* be reassigned values.