Creating a Class in Scala

In this lesson, we will start working with object-oriented programming and you will learn how to create your own class in Scala.

We'll cover the following

Introduction

In a previous lesson, you were introduced to classes and objects. We looked at some of Scala's built-in classes such as `Seq`. But you can make your own classes and, in this chapter, we will cover user-defined classes and learn how to write our very own classes and objects. We will pick up where we left off in the introductory lesson; it is recommended you go over the lesson once more before moving on.

Without further ado, let's create our own `Person` class.

Defining a Class

To define a class in Scala, the `class` keyword is used, followed by an identifier (class name) of our choosing.

```
class classIdentifier
```

Let's map the syntax to our `Person` class.

```
class Person
```

At this point, our class doesn't have any properties or methods.

Defining the Properties of a Class

Defining the Properties of a Class

Properties are variables and in Scala, they are called **fields**. They are also known as *instance variables* because every instance gets its own set of the variables.

> All the members of a class, including fields, are placed in a block.

```
class classIdentifier{
  var field1 = Initial Value
  var field2 = Initial Value
  .
  .
  var fieldn = Initial Value
}
```

Our `Person` class has three fields: **name**, **gender**, and **age**.

```
class Person{
  var name: String = "temp"
  var gender: String = "temp"
  var age: Int = 0
}
```

Make sure you define the class fields using `var` as the values will be reassigned for each instance of the class.

Defining the Methods of a Class

The methods of a class are executable functions which can only be used by instances of that class. Our methods are **walking** and **talking** which will simply print the name of the person that is walking or talking respectively.

```
class Person{
  var name: String = "temp"
  var gender: String = "temp"
  var age: Int = 0

  def walking = println(s"$name is walking")

  def talking = println(s"$name is talking")
```

```
}
```

The methods of a class have access to the fields of the same class. This is why, in our `Person` class, we can use `name` without passing it to `walking` and `talking`.

And with this, we have created our first class with three fields and two methods. In the next lesson, we will use our `Person` class to create a `Person` object.