# Type Inference

In the following lesson, you will be introduced to type inference and learn about Scala's ability to infer data types.

Type inference is Scala's ability to infer types when not specified by the user. To better understand how this works, let's look at the example we used in the first lesson of this chapter where we had to declare variables for the author, title, and number of pages in a book.

This code requires the following environment variables to execute:

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

```
val bookTitle: String = "Lord of the Rings: The Fellowship of the Ring"
val bookAuthor: String = "J. R. R. Tolkien"
val bookNoOfPages: Int = 423

// Driving Code
println(bookTitle)
println(bookAuthor)
println(bookNoOfPages)
```

In the code above, we are using the following syntax:

```
keyword variableName: DataType = Initial Value
```

However, Scala's type inference feature allows us to declare a variable without explicitly mentioning the data type. This is why Scala's variable declaration enforces the data type to be mentioned after the variable name; for easy removal. This results in the following syntax:

```
keyword variableName = Initial Value
```

Let's now map this syntax to our book example and print the values of the variables.

This code requires the following environment variables to execute:

LANG C.UTF-8

```
val bookTitle = "Lord of the Rings: The Fellowship of the Ring"
val bookAuthor = "J. R. R. Tolkien"
val bookNoOfPages = 423

// Driver Code
println(bookTitle)
println(bookAuthor)
println(bookNoOfPages)
```

When you press **RUN** the program should successfully execute. Pretty cool, right?

But how do we know if bookTitle is even being declared as a `String`?
Unfortunately, there is no way in Scala to print the variable type, however, we can try to produce an error which will let us know the data type of our variables.

Let's try to assign the value of `bookTitle` to a variable of type `Int`:

This code requires the following environment variables to execute:

LANG C.UTF-8

```
val bookTitle = "Lord of the Rings: The Fellowship of the Ring"

val testInteger: Int = bookTitle
```

When you run the code above, the console should show you:

```
error: type mismatch
```

```
found: String
```

```
required: Int
```

This is letting us know that the variable `testInteger` requires an `Int` type value, but we are providing a `String` type value. This error confirms that the variable `bookTitle` is of type `String`.

As the Scala interpreter/compiler provides type inference, it is better to let it do so rather than writing the data type and unnecessarily filling your code. Of course, if

a situation occurs where you are required to explicitly mention the data type, you should do so. That's the beauty of Scala, it gives you multiple options based on your requirements.

Moving forward, we will depend on type inference and will only mention the data type when needed.

---

With type inference, our discussion on variables and types comes to an end. Let's wrap up this chapter with a quiz to test what you have learned so far.