Printing in Scala

In the following lesson, you will be introduced to Scala's Hello World program and learn about Scala's different printing methods.

We'll cover the following

Hello World!

While learning a new programming language, the first program you usually learn to code is the famous ‘Hello World’ program.

Let’s see how the program looks in Scala:

This code requires the following environment variables to execute:

LANG	C.UTF-8

```
print("Hello world!")
```

Printing Methods in Scala

Printing in Scala is quite straight forward. However, there are three different methods you can use depending on your objective.

1. `print`
2. `println`

3. `printf`

Each print method follows the same basic syntax:

```
printMethod("text to be printed")
```

Let’s look at each method in action.

print

`print` is the simplest method for displaying output in Scala. It just prints anything you pass it, one after the other, in the same line.

This code requires the following environment variables to execute:

LANG	C.UTF-8

```
print("Hello world!")
print(3)
print("How are you?")
```

println

`println` is used if you want to display each specific output on separate lines. With each `println` statement, the output is automatically displayed on the next line.

This code requires the following environment variables to execute:

LANG	C.UTF-8

```
println("Hello world!")
println(3)
println("How are you?")
```

printf

`printf` is used for formatting text. You can use it to append different data types to your text that is to be printed. If this sounds a bit confusing, let’s have a look at an example where we append an integer to some text using both the `println` method and `printf` methods. This will help you better understand the differences between the two methods.

This code requires the following environment variables to execute:

LANG	C.UTF-8

```
println("Number = %d", 123)
printf("Number = %d", 123)
```

In the above code, when we use `println`, the compiler prints the arguments between the parenthesis as is, while `printf` inserts **123** after "Number = ". We will be looking into this with great detail in a later chapter on strings.

Printing Variables

In the last lesson, we declared a variable `myFirstScalaVariable`; let's learn how to print it.

This code requires the following environment variables to execute:

LANG	C.UTF-8

```
val myFirstScalaVariable: Int = 5
println(myFirstScalaVariable)
```

In the code snippet above, we simply passed the name of our variable to the print statement and in return, the value assigned to the variable, i.e. **5**, was displayed.

The basic syntax is as follows:

```
printMethod(variableName)
```

Now that we know how to print variables, let's get back to where we left off in the previous lesson and look at immutable variables in more detail in the next lesson.