Exercise 1 – Introduction

Objective

The objective of this exercise is to ensure that scala is setup correctly and that we can use it in three different ways.

* Use the scala REPL
* Create an eclipse or IntelliJ worksheet
* Setup an SBT projet
* Write your first Hello World statement

References:

Chapter 1: Slides 10 to 24

Overview

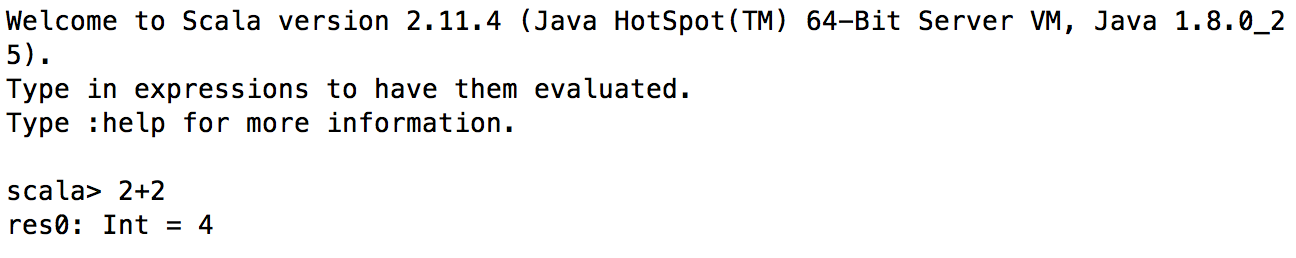
Using the command line

Open a terminal window. If you’re using a mac then this can be achieved by pressing the apple key and space to open spotlight and typing ‘terminal’ before pressing enter. In windows we do this by pressing the windows key to open the start menu, typing ‘cmd’ and then pressing enter.

In the console window open the REPL by typing ‘scala’ without the quotes.

REPL stands for ‘Read Evaluate Print Loop’. It will evaluate commands it is given and provide the output they produce and then wait for some more input. We can test scala commands using this and view the output straight away. This is different from some programming languages where we would need to compile our code before running it. The REPL evaluates every statement and retains previous statements in memory allowing us to reuse them.

Enter some expressions, such as 2+2, and see how they are evaluated. The answer should be returned to you straight away.



The result of the statement has been printed to the screen. When types are not explicitly defined they are inferred by the scala interpreter. In this case we can see that the result is 4, which has the type Int. Scala has generated an identifier for the result, 'res0' We can reuse in future commands, for example:

scala> res0 + 1

res1: Int = 5

Next, enter a function definition

scala> def Hello() = "hello"

The REPL should reply with

Hello: ()String

This shows that it infers Hello to be a function that has no parameters and returns a string. It also shows you that the return value from a function is the last value calculated.

To execute the function we use the name.

scala> **Hello**

res2: String = hello

This has evaluated the value "Hello", and the REPL tells you that it has evaluated it to the string

To print the actual string, use println()

scala> println(Hello)

hello

Note two things about this line: first, that most of the time you don't have to use semicolons unless you have more than one statement on a line. And second, you don't have to use empty parentheses if a function doesn't have any arguments.

To exit the REPL type :quit

To see the full list of available commands in the REPL type :help

Using Eclipse

We can do the same thing in Eclipse. The Scala IDE has a useful feature called a worksheet, which lets you work in the same way as the REPL, but which lets you save your interactive experiments.

Start the Eclipse IDE, and create a Scala project called "Course"

Add a package called "Ex1"

Right-click on the package name, and choose Scala Worksheet from the context menu. Call it "hello"

Enter the definition of the Hello function after the println statement

Now save the file. This will compile and run the code, and you will see the output from the REPL in comments on the right-hand side of each line

Add the println statement to execute Hello()

IntelliJ Setup

The scala plugin is not included in the community edition of IntelliJ by default. To include the plugin use the following steps. If you already have the plugin then just straight to ‘IntelliJ Worksheet’.

* To install the scala plugin:
  + Open IntelliJ
  + Click the Configure option at the bottom of the window and select “plugins”
  + Click Browse Repositories at the bottom
  + In the search bar at the top type “scala”
  + The plugins to install are the “SBT” plugin and the “Scala” plugin, select these and click “Install plugin” on the right
  + Once they have installed click close, then ok
  + Restart IntelliJ when prompted

Note: To install and run IntelliJ on a mac you need to have the legacy Java 1.6 runtime installed.

IntelliJ Worksheet

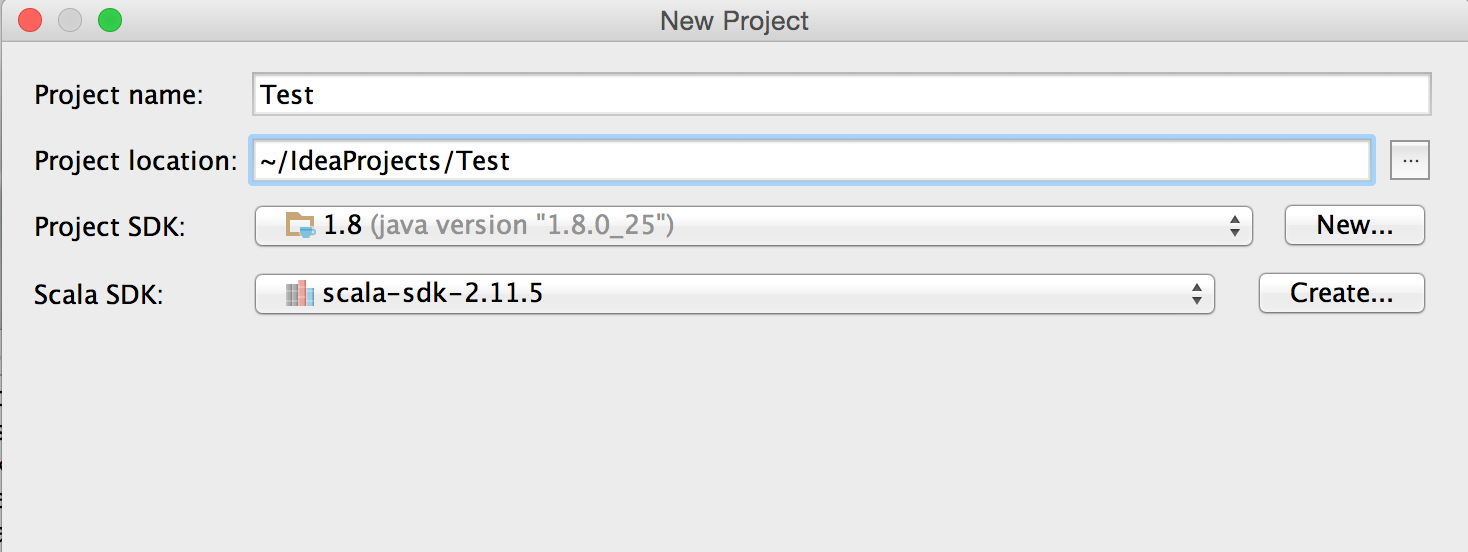
Open IntelliJ and create a new project. Down the left hand side you should see a ‘scala’ option. Select this and create a scala (not sbt) project. The worksheets do not function correctly with SBT projects due to how they are compiled and run.

A new window will open . Give your project a name. We then need to include a JDK and Scala SDK before continuing. If there is a JDK in the drop down box, select this, we want Java 1.7+. If there is not one then click the button to the right, “New…”

Select the JDK option and then navigate to where your Java installation is located. It should pick up on the standard installation directory. Click okay and ensure that a JDK is now included.

Next select a Scala SDK, if there is no option available click the button to the side, “Create …”. We want to be using at least version 2.11, so if there is not a 2.11 option then click the Download button and select the most recent version to download. Click okay to have this installed by IntelliJ.

The final setup should look something like this:



Once these steps have all been completed then you should be able to create the new project. This setup is only something you need to do once, when creating a new project the JDK and SDK will be available for use.

To create a Scala worksheet expand the project on the left hand side of the screen and right click the ‘src’ folder. Select new and Scala Worksheet. Give the file a name.

Now type your scala commands in the window. When you save the file (or whenever you change if it auto-updates are enabled) then file will be evaluated and the output displayed on the right hand side of the screen.

Create your hello world function and call it using the worksheet.

SBT – Simple Build Tool

The simple build tool is similar to dependency and build managers such as maven or gradle. It also has the ability to pick up on changes in the source code and automatically rebuild a project, allowing for continuous development and integration.

SBT can be downloaded from: <http://www.scala-sbt.org/>

To start an sbt project create a new directory on your machine by typing into a terminal window

mkdir sbttest

To change to this directory type

cd sbttest

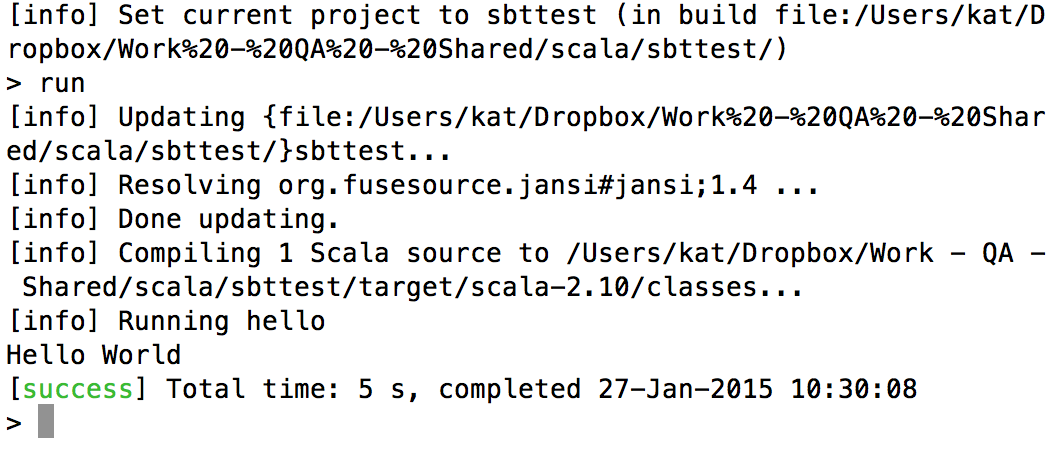
SBT projects require an object to be created rather than individual statements as we have been using in the REPL and worksheets. We will go into how these work later in the course. For now just create a new file in the sbttest directory called Hello.scala which contains the following

object hello extends App {

println("Hello World")

}

Then in the terminal window type sbt. This will start up the sbt command line. From here we have another set of commands we can use such as compile, or run. Type run and your project should build itself and run outputting “hello world” to the screen.



The first time you run sbt it will download a set of dependencies required for it, so the initial run of any sbt project may take a while, but after this point it will be faster.

Write any scala commands inside the { … } brackets in the file, save and re-run the file from sbt. You will see that it recompiles each time before running. You will need to enclose the commands in a println() statement to output them to the screen. SBT does not give you the direct feedback that the worksheets or repl does.

To exit sbt type exit

If you have time:

The hello world tutorial at: <http://www.scala-sbt.org/0.13/tutorial/Hello.html> guides you through more advanced setups using the SBT build files. If you have time take a look at this and see if you can create a full SBT project.