

# Setting up the development environment

## Instructions for the upcoming Scala course

Heiko Seeberger

Skills Matter

2011-08-22 to 2011-08-23



# Agenda

Setting up the development environment



# Scala distribution



## Exercise: Install the Scala distribution

- ▶ Download the current stable release as an archive for your platform (*.tgz* or *.zip*) from [www.scala-lang.org/downloads](http://www.scala-lang.org/downloads)
- ▶ Unpack the archive to a suitable location, e.g. *~/tools/scala*
- ▶ Add the *bin* directory to your path
- ▶ Verify the installation by opening a terminal and entering *scala -version*:

```
1 tmp$ scala -version
2 Scala code runner version 2.9.0.1 -- Copyright
   2002-2011, LAMP/EPFL
```

- ▶ Also download the Scala API documentation, unpack and browse it



## Exercise: "Hello World!" on the command line

- Create the file *Hello.scala*<sup>1</sup> using an arbitrary text editor:

```
1 object Hello {  
2   def main(args: Array[String]) {  
3     println("Hello World!")  
4   }  
5 }
```

- Compile and run it:

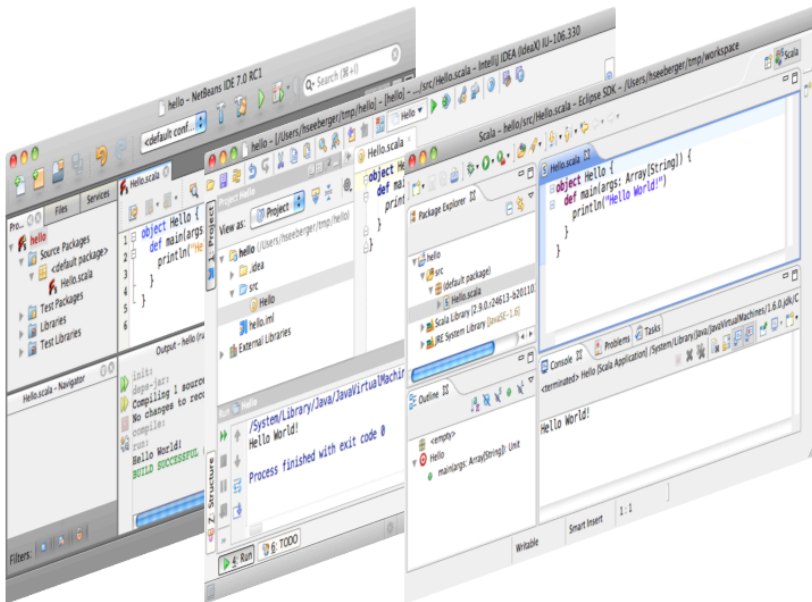
```
1 tmp$ scalac Hello.scala  
2 tmp$ scala Hello  
3 Hello World!
```

---

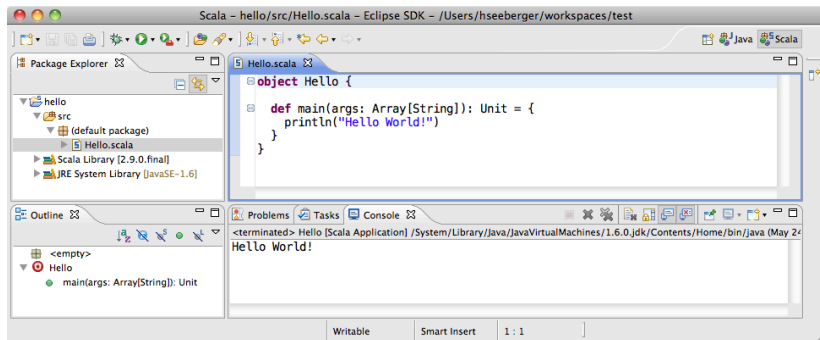
<sup>1</sup>You don't have to understand the code yet!



# There are plugins for all major IDEs



# Scala IDE for Eclipse



- ▶ We, the trainer(s), will use Eclipse for this course
- ▶ Feel free to use another IDE or none at all, but we will only be able to offer limited support



## Exercise: Install Eclipse and the Scala IDE for Eclipse

- ▶ Download and install Eclipse Indigo (3.7) Classic for your platform from [www.eclipse.org/downloads/](http://www.eclipse.org/downloads/)
- ▶ Install the Scala plugin via the menu *“Help > Install New Software ...”* using the update site [download.scala-ide.org/releases-29/2.0.0-beta](http://download.scala-ide.org/releases-29/2.0.0-beta)
- ▶ Verify the installation by opening a fresh workspace, e.g. *~/workspaces/fasttracktoscala* and switching to the Scala perspective





## Exercise: "Hello World!" in Eclipse

- ▶ Create a *"New > Scala Project"* with name *hello*
- ▶ Create a *"New > Scala Object"* with name *Hello*
- ▶ Copy the code from the previous exercise
- ▶ Select *"Run As > Scala Application"* from the context menu of the editor or package explorer
- ▶ In order to avoid conflicts with other future projects we suggest you now close or delete this project



# Simple Build Tool (sbt)

```
tmp$ cd scalatrain
scalatrain$ sbt
[info] Set current project to default (in build file:/Users/hseeberger/.sbt/plugins/)
[info] Set current project to default (in build file:/Users/hseeberger/tmp/scalatrain/)
> compile
[success] Total time: 0 s, completed May 24, 2011 1:14:42 PM
```

- ▶ THE build tool for Scala
- ▶ Written in Scala and specifically for Scala
- ▶ Used by most real-world projects



## Exercise: Install sbt

- ▶ Download the launcher:  
[repo.typesafe.com/typesafe/releases/org.scala-tools.sbt/sbt-launch/0.10.1/sbt-launch.jar](https://repo.typesafe.com/typesafe/releases/org.scala-tools.sbt/sbt-launch/0.10.1/sbt-launch.jar)
- ▶ Create the following file as a start script for sbt:
  - ▶ *sbt* on Mac/Linux: `java -Xmx512M -jar LAUNCHER "$@"`
  - ▶ *sbt.bat* on Windows: `java -Xmx512M -jar LAUNCHER %*`



## Exercise: Create a sbt project

- ▶ Create a fresh project directory, e.g. `~/projects/scalatrain` and `cd` into it
- ▶ Attention: Do not create this in your Eclipse workspace!
- ▶ Starting sbt will take you to an interactive session
- ▶ Execute the following three commands at the sbt prompt:

```
1 > set name := "scalatrain"
2 ...
3 > set scalaVersion := "2.9.0-1"
4 ...
5 > session save
6 ...
```

- ▶ Take a look at the fresh file *build.sbt*
- ▶ Keep the sbt session running!



# sbt commands - quick overview

- ▶ General commands:
  - ▶ *exit* or *quit* ends the current session
  - ▶ *help* lists available commands
- ▶ Build commands:
  - ▶ *compile* compiles main sources
  - ▶ *test:compile* compiles test sources
  - ▶ *test* runs tests
  - ▶ *console* starts the REPL
  - ▶ *run* looks for a main class and runs it
  - ▶ Triggered execution: Prefix a command with ~
- ▶ Other commands:
  - ▶ *clean* deletes all output in the *target* directory
  - ▶ *reload* reloads the build



# Exercise: Install the sbt-Eclipse integration

- ▶ The **sbteclipse** plugin let's you create Eclipse project files from an sbt project
- ▶ In the project directory create the file *project/plugins/build.sbt* with the below contents
- ▶ Attention: Copy and paste is your friend, but pay attention to the details:
  - ▶ The blank line between the blocks is important!
  - ▶ Sometimes the quotes are not copied correctly!
  - ▶ You could also copy from [github.com/typesafehub/sbteclipse](https://github.com/typesafehub/sbteclipse)

```
resolvers += {  
  val typesafeRepoUrl = new java.net.URL("http://repo.typesafe.com/typesafe/releases")  
  val pattern = Patterns(false,  
    "[organisation]/[module]/[sbtversion]/[revision]/[type]s/[module](-[classifier])-[revision].[ext]")  
  Resolver.url("Typesafe Repository", typesafeRepoUrl)(pattern)  
}  
  
libraryDependencies <=> (libraryDependencies, sbtVersion) { (deps, version) =>  
  deps :+ ("com.typesafe.sbteclipse" %% "sbteclipse" % "1.3-RC2" extra("sbtversion" -> version))  
}
```



## Exercise: Create Eclipse project files

- ▶ In the sbt session execute the commands *reload* and the now available *eclipse* with argument *create-src*

```
1 > reload
2 ...
3 > eclipse create-src
4 ...
5 [info] Successfully created Eclipse project files ...
```

- ▶ Import the new Eclipse project using “*Import...*” > “*Existing Projects into Workspace*”
- ▶ Verify the import by inspecting the project, e.g. the source folders *src/main/scala* etc.



Unless otherwise agreed, training materials may only be used for educational and reference purposes by individual named participants in a training course offered by Typesafe or a Typesafe training partner. Unauthorized reproduction, redistribution, or use of this material is prohibited.

