



**ISA**

Institut für  
SoftwareArchitektur



**THM**

TECHNISCHE HOCHSCHULE MITTELHESSEN



# Compilerbau cs1019

Th. Letschert

TH Mittelhessen Gießen

*University of Applied Sciences*

**Scala und SBT**

# Unterlagen zu Scala

## Literaturhinweise

M. C. Lewis: *Introduction to the Art of Programming using Scala*  
M. Odersky, L. Spoon, B. Venners: *Programming in Scala*, 3-te Auflage

*beide Bücher sind  
in der THM-  
Bibliothek verfügbar*

## Online-Quellen

<http://www.scala-lang.org> *Der offizielle Startpunkt*  
<https://www.scala-exercises.org> *Erklärungen und Übungen zu einzelnen Themen*  
[https://twitter.github.io/scala\\_school/](https://twitter.github.io/scala_school/) *Scala lernen bei Twitter*  
<http://scalatutorials.com> *Interaktives Tutorial*  
<http://www.scala-js-fiddle.com> *Scala lernen nur mit einem Browser*  
<http://www.scala-lang.org/community/> *hier finden sich weitere Verweise*  
<https://alvinalexander.com/scala/learning-functional-programming-in-scala-book>  
*Erfahrungsgemäß sind die Konstrukte zur funktionalen Programmierung die größte erste Hürde für viele Scala-Neulinge. Alvin Alexander erklärt hier alles sehr ausführlich. (Eine ältere Version seines Buchs ist unter dem Link zu finden und frei verfügbar)*

## Videos

<https://www.youtube.com/watch?v=kkTFx3-duc8> *Odersky: Scala with style*  
<https://www.youtube.com/watch?v=ecekSCX3B4Q&feature=youtu.be> *Odersky: Scala the simple parts*

...

# Scala Installation

## Installation SBT und Scala

### Paket-Manager und Java-8

Stellen Sie sicher, dass auf ihrem System Java-8 in einer aktuellen Version installiert ist.

Machen Sie sich kundig ob Ihr System einen Paket-Manager unterstützt und wenn ja welchen.

### Scala und SBT via Paket-Manager

SBT – **Simple Build Tool** – ist das *Standard-Build-System* für Scala.

Installieren Sie SBT auf Ihrem System, am besten verwenden Sie dazu den lokalen Paket-Manager:

- Mac: Installation mit Paket-Manager *Brew*

```
brew install sbt  
brew install scala
```

<code>brew update</code>	<i>brew aktualisieren</i>
<code>brew upgrade</code>	<i>Installierte Pakete aktualisieren</i>

- Linux: Installation mit Paket-Manager *Apt (Advanced Packaging Tool)*

```
apt-get install sbt  
apt-get install scala
```

- Windows 10: Installation mit Paket-Manager *OneGet*

Oder wählen Sie eine manuelle Installation.

Weitere Details zu SBT finden sich unter: <http://www.scala-sbt.org>

# Scala Installation

---

## Installation SBT und Scala

### Sbt und Scala-Installation prüfen

```
$ scala -version
```

```
Scala code runner version 2.12.13 -- Copyright 2002-2017, LAMP/EPFL and Lightbend, Inc.
```

```
$ sbt sbt-version
```

```
[info] 0.13.16
```

```
$ which scala
```

```
/usr/local/bin/scala
```

```
$ which sbt
```

```
/usr/local/bin/sbt
```

*Version und Installationsort anzeigen lassen.*

# Scala verwenden

## Scala bedienen: REPL

Scala bietet einen interaktiven **Interpreter** – die REPL (REPL = *Read-EvaluatePrint Loop*)  
Aufruf mit dem Kommando **scala**

```
$ scala
```

```
Welcome to Scala 2.12.3 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_45).
```

```
Type in expressions for evaluation. Or try :help.
```

```
scala> def f(x: Int): Int = if (x==0) 1 else f(x-1)*x  
f: (x: Int)Int
```

*Fakultätsfunktion definieren*

```
scala> f(5)  
res0: Int = 120
```

*... und auf 5 anwenden.*

```
scala> ^d  
$
```

# Scala verwenden

## Scala bedienen: Compiler

Die Scala-Distribution umfasst einen **Compiler** mit dem Scala-Programme in Bytecode übersetzt werden können.

Aufruf mit dem Kommando **scalac**

```
$ cat -> HalloWelt.scala
object HalloWelt extends App { println("Hallo Welt") }
^d
```

```
$ scalac HalloWelt.scala
```

```
$ scala HalloWelt
Hallo Welt
```

```
$ ls
HalloWelt$.class      HalloWelt.class
HalloWelt$delayedInit$body.class  HalloWelt.scala
```

*Hallo-Welt Programm in der Datei  
HalloWelt.scala erstellen.*

*... übersetzen*

*... und ausführen*

*Die erzeugten Dateien  
anzeigen lassen*

**cat** ist ein Unix-Kommando.

Es wird hier genutzt um auf einfache Art eine Datei zu erstellen. Das geht natürlich auch mit einem Texteditor.

Die Kommandos (ausführbaren Programme) **scala** und **scalac** entsprechen **java** und **javac** .

# Scala verwenden

## Scala und Java

Scala wird in Java-Bytecode übersetzt und auf der JVM ausgeführt.

Übersetzte Scala-Programme können darum auch mit **java** ausgeführt werden.

Dazu muss natürlich der Klassenpfad so gesetzt werden, dass die benötigten jars zur Verfügung stehen

```
$ which scala
/usr/local/bin/scala
```

*wo sind die Scala-Jars?*

```
$ ls -l /usr/local/bin/scala
lrwxr-xr-x ... /usr/local/bin/scala -> ../Cellar/scala/2.12.3/bin/scala
```

```
$ ls /usr/local/Cellar/scala/2.12.3/libexec/lib/scala-library.jar
/usr/local/Cellar/scala/2.12.3/libexec/lib/scala-library.jar
```

```
$ java -cp /usr/local/Cellar/scala/2.12.3/libexec/lib/scala-library.jar:. HalloWelt
Hallo Welt
```

*Ausführung direkt durch die JVM.*

## Scala Projekt mit SBT erzeugen

### Giter 8 Template verwenden

*Giter8 is a command line tool to generate files and directories from templates published on Github or any other git repository. It's implemented in Scala and runs through the sbt launcher, but it can produce output for any purpose. Siehe: <http://www.foundweekends.org/giter8/>*

### Beispiel:

```
$ sbt new scala/hello-world.g8
```

```
...
```

```
A template to demonstrate a minimal Scala application
```

```
name [Hello World template]: Hallo Welt
```

```
Template applied in ./hallo-welt
```

```
...
```

```
$ sbt
```

```
> run
```

```
[info] Running Main
```

```
Hello, World!
```

```
[success] Total time: 14 s, completed 01.08.2017 15:24:49
```

```
>
```

siehe auch:

<http://www.scala-lang.org/documentation/getting-started-sbt-track/getting-started-with-scala-and-sbt-in-the-command-line.html>



## Scala-Anwendungen entwickeln

### Entwicklungsumgebungen für Scala

Zur Entwicklung von Scala-Anwendungen reicht eine Scala-Installation und ein Editor.

Ob

- man damit zufrieden ist oder
- mehr gebraucht wird, und wenn ja was,

das ist Geschmacksache, und wird intensiv diskutiert.

Die üblichen Ratschläge sind:

- Editor mit Scala-Unterstützung Emacs / Atom / ...
- verwende eine IDE: IntelliJ / Eclipse / Netbeans
- ...

# Entwicklungsumgebung für Scala

## Scala-Anwendungen entwickeln

### Entwicklungsumgebungen für Scala

Wir empfehlen die Verwendung von SBT als build tool mit IntelliJ als Entwicklungsumgebung  
siehe:

- <https://www.jetbrains.com/help/idea/getting-started-with-sbt.html>
- <https://www.jetbrains.com/help/idea/creating-and-running-your-scala-application.html>

