

# BASIC WICKET AND SCALA

Thanks!

Thanks for attending the presentation about Basic Wicket and Scala!  
This handout contains all the important stuff from the presentation.

## What is Scala?

Scala is a programming language designed by Martin Odersky. First released in 2003, the language builds on the power of the Java VM and object-oriented and functional concepts.

Scala is interoperable with Java, as you can see in the demo projects that I prepared for this talk.

Although Scala compiles to Java bytecode, it is different from Java. Everything is an object in Scala, even functions. This makes it possible to pass functions around (closures). As the next version of Java also may include closures, Scala is a good learning environment. Every programmer should sometimes look outside of their realm, to learn about new stuff in the software development arena. Scala is an inviting language designed by the guy who brought you Java 5 generics.

Get started today with the sample code!

## Scala and Wicket

Scala and Wicket are a good fit. Easy, reusable, non-intrusive, safe and scalable are the design goals of Wicket, and Scala matches those, while providing more language features than Java.

## Hello, World!

```
object HelloWorld {  
  def main(args: Array[String]) {  
    println("Hello, world!")  
  }  
}
```

## Compile and run

```
$ scalac HelloWorld.scala  
$ scala HelloWorld
```

Or from Java:

```
$ java -cp scala-library.jar:.  
HelloWorld
```

## Java + Scala

It is possible to run Java and Scala code together, happily interacting with each other.

With the latest Scala compiler plugins for Maven, circular dependencies between Java and Scala are automatically compiled!

The Scala compiler parses

Java source code, so it knows about the Java classes you use from Scala.

London  
February 4  
2009

Daan van Etten



stuq.nl

## Sample code

All samples from the presentation can be downloaded at

<http://londonwicket.org>  
and <http://stuq.nl>.

More info?  
<http://stuq.nl>

# BASIC WICKET AND SCALA

## Sample code

All samples from the presentation can be downloaded at <http://londonwicket.org> and <http://stuq.nl>.

## Hello, Wicket World!

```
import org.apache.wicket._
import org.apache.wicket.protocol.http._
import org.apache.wicket.markup.html._
import org.apache.wicket.markup.html.form._
import org.apache.wicket.model.PropertyModel
import org.apache.wicket.markup.html.basic.Label

class HelloWicketWorld extends WebApplication {
  def getHomePage = classOf[HomePage]
}

class HomePage extends WebPage {
  var name = ""
  val form = new Form("form")
  add(form)
  form.add(new TextField("name", new PropertyModel(this, "name")))
  form.add(new Label("helloworld", new PropertyModel(this, "name")))
}
```

## interface vs trait

### Java interface:

```
public interface SomeInterface {
  void doSomething(String argument);
}
```

### The same, but now as trait in Scala

```
trait SomeTrait {
  def doSomething=(argument:String):Unit
}
```

Traits can have abstract methods (like Java interfaces), but also complete methods.

## unit vs void

In Scala, unit is the same as void is in Java. So a method that doesn't return anything is declared like this:

```
def doSomething(argument: String):unit
```

## val and var

Variables can be defined using val and var. Val is immutable (like a const in Java), var is mutable.

```
var mutable = "foo"
val immutable = "bar"
```

## type inferring

Scala infers which type you use:

```
var foo = 3654
```

The foo variable is now an Int. So when assigning a String, you get an error:

```
foo = "hi"
error: type mismatch;
```

## Samples

The downloadable samples zip-file contains:

- 'Hello World' Scala example
- 'Hello, Wicket World!' Scala example project with a small Wicket demo application
- 'Hello, Wicket World!' Scala example project, built by Maven with Jetty:run support and building of circular dependencies.