

Scala and its Ecosystem

Petr Hošek @petrh
CZJUG, September 2010

What is language *ecosystem*?

Tools, libraries, frameworks, community, etc.
Language is useless without its ecosystem.

Why Scala *ecosystem*?

With growing usage grows the need.
Java ecosystem is not enough.

Scala Compiler

Most important part of the Scala ecosystem.

Compiler **overview**

Version 2.8 released in July.

Contains many new features.

Second generation of Scala compiler.

Officially named *NSC* (New Scala Compiler).

Entirely developed in Scala itself.

Self-hosted compiler.

Compiler **usage**

Compiler can be used in several ways:

- Standalone compiler.
- Interactive interpreter.
Usually called *REPL* (Read-Evaluate-Print Loop).
- Part of Scala application.
Instance of compiler class, either batch or interactive.

Demo

Compiler usage

Compiler **architecture**

Compiles Scala directly into Java bytecode.

Compilation consists of several phases.

- Each phases transforms the syntax tree.

Compiler is developed in entirely modular way.

Designed to be completely re-entrant.

- Can be instantiated as any other class.

Usage of advanced Scala language constructs.

Compiler plugins

Provide way to easily extend compiler.

Without modifying its source code.

Can be inserted in all phases of compilation.

Allows code transformations, analysis, etc.

Used for some experimental language constructs.

Continuations, etc.

Demo

Compiler plugins

Scala **tools** and **frameworks**

Basis of the Scala ecosystem.

A word cloud on a black background featuring various Scala-related terms. The central and largest text is 'simple-build-tool' in white. Other prominent terms include 'ScalaCheck' (red), 'Scalatra' (white), 'Specs' (white), and 'Akka' (red). Smaller terms include 'Scalaz', 'scalala', 'Configgy', 'sbaz', 'Jiva', 'BindForge', 'Scalax', 'pinky', 'kiama', 'ScalaModules', 'sweet', 'Kestrel', 'Scalate', 'ScalaTest', and 'Lift'. The colors alternate between white and red for different words.

ScalaCheck Scalatra Specs
simple-build-tool Akka
Scalaz scalala Configgy sbaz BindForge Scalax pinky kiama ScalaModules sweet
Kestrel Scalate ScalaTest
Lift

simple-build-tool

Simple but powerful build tool for Scala.

Configuration is written in Scala.

Provides interactive and batch mode.

Dependency management support.

Based on Apache Ivy dependency manager.

Support for other Scala tools.

Many customizations and extensions exists.

Demo

simple-build-tool usage

Specs

Behaviour-driven design framework for Scala.

Simple and typed language for specifications.

Benefits from Scala expressive syntax.

Integration with testing tools and frameworks.

JUnit, ScalaCheck, Mockito, etc.

Demo

specs usage

Lift web framework

Expressive and elegant web framework.

Benefiting from Scala language features.

Embraces View-First approach to MVC.

View-Driven Development paradigm.

Importance of scalability and security.

Without loss of performance or maintainability.

Native support for Ajax a Comet.

Demo

Lift usage

Well-known Scala tools and libraries:

- sbaz
Package sharing tool.
- ScalaCheck
Tool for automatic test case generation.
- Akka framework
Concurrent, scalable applications development.
- Configgy
Library for handling configuration and logging.
- scalaz, scalax
Extensions to the standard Scala library.

Many more **tools** and **libraries** do exists
New ones are coming up to Scala ecosystem every day.

Collaborative Scaladoc

Newest addition to Scala ecosystem.

Project **overview**

Development of Scala project documentation.

Using the concepts of social collaboration.

Contribute without digging into source code.

Try to embrace new approach of source-code documentation authoring.

Scaladoc

Analogy of Javadoc for Scala.

Second generation of Scaladoc is a part of Scala 2.8.

Contains new sleek and modern interface.

Provides improved comment syntax.

Supports wiki-like syntax in the source comments.

Colladoc application

Allows to edit Scala symbols documentation.

Lift web application running the Scala compiler.

Developed as a Google SoC 2010 project.

Now being developed as open-source project.

Based heavily upon Scaladoc 2 functionality.

„Do not reinvent the wheel.“

Mergedoc application

Allows to merge changes into the source-code.

Simple command-line utility.

More features planned for the future.

Not yet officially released.

Reimplementation of *scaladoc-merge* tool.

Demo

Colladoc and Mergedoc usage

Project **status**

The idea already proved to be viable.

Comments are being propagated to standard library.

Project needs more popularisation.

There are many ideas waiting to be implemented.

Most them originated in the community.

We need *you!*

Project is looking for collaborators.

Learn **more** at
<http://petrhosek.name/>