Scala School!

From ⌀ to Distributed Service

Other Languages:

한국어
Русский
简体中文

About

Scala school started as a series of lectures at Twitter to prepare experienced engineers to be productive Scala programmers. Scala is a relatively new language, but draws on many familiar concepts. Thus, these lectures assumed the audience knew the concepts and showed how to use them in Scala. We found this an effective way of getting new engineers up to speed quickly. This is the written material that accompanied those lectures. We have found that these are useful in their own right.

Approach

We think it makes the most sense to approach teaching Scala not as if it were an improved Java but instead as a new language. Experience in Java is not expected. Focus will be on the interpreter and the object-functional style as well as the style of programming we do here. An emphasis will be placed on maintainability, clarity of expression, and leveraging the type system.

Most of the lessons require no software other than a Scala REPL. The reader is encouraged to follow along, but also to go further! Use these lessons as a starting point to explore the language.

Also

You can learn more elsewhere:

- **Effective Scala** Twitter's "best practices" for Scala. Useful for understanding idioms in Twitter's code.
- **scala-lang.org Documentation** Links to tutorials, manuals, API reference, books, ...
- **Scala API Documentation**

Lessons

Basics
Values, functions, classes, methods, inheritance, try-catch-finally. Expression-oriented programming

Basics continued
Case classes, objects, packages, apply, update, Functions are Objects (uniform access principle), pattern matching.

Collections
Lists, Maps, functional combinators (map, foreach, filter, zip, folds)

Pattern matching & functional composition
More functions! PartialFunctions, more Pattern Matching

Type & polymorphism basics
Basic Types and type polymorphism, type inference, variance, bounds, quantification

Advanced types
Advanced Types, view bounds, higher-kinded types, recursive types, structural types

Simple Build Tool
All about SBT, the standard Scala build tool

More collections
Tour of the Scala Collections library

Testing with specs

Concurrency in Scala
Runnable, Callable, threads, Futures

Java + Scala
Java interop: Using Scala from Java

An introduction to Finagle
Finagle primitives: Future, Service, Filter, Builder

Searchbird
Building a distributed search engine using Finagle

Built at @twitter by @stevej, @marius, and @lahosken with much help from @evanm, @sprsquish, @kevino, @zuercher, @timtrueman, @wickman, and @mccv; Russian translation by appigram; Chinese simple translation by jasonqu; Korean translation by enshahar;