

Training: Compendium CE  
 Introduction to programming in Scala


FORM OF TRAINING	MATERIALS	PRICE	DURATION
Traditional	Hardcopy	500 EUR	2 days
Traditional	Digital materials	500 EUR	2 days
Traditional	CTAB Tablet	650 EUR	2 days

## LOCATIONS

Krakow - 5 Tatarska Street, II floor, hours: 9:00 am - 4:00 pm

Warsaw - 17 Bielska Street, hours: 9:00 am - 4:00 pm

## TRAINING GOALS:

This training will guide programmers to a point where they can write and read Scala programs in a fairly fluent way. All core features of the language are covered, including pattern matching and how to approach problems in a functional style.

The course focuses on "Scala the language" so no more than a minimum of external frameworks are covered during this course - SBT, ScalaTest and Akka being notable mentions.

Examples and real life use-cases back each of the introduced features and principles. Participants will be asked to write automatic tests at the end of each section to get hands-on experience on the given topic.

## CONSPECT:

### Day 1

- Introduction
  - Course schedule
- A gentle introduction to the Scala language
  - What is Scala?
  - What are it's sources of inspiration?
  - Why is Scala worth learning?
- "Hello World"
  - "Java in Scala"
  - The Scala way to "Hello World"

- Types
  - Basic types
  - Primitives?
- Classes
  - Primary constructor
  - Secondary constructors
  - Inheritance
  - Type Linearization
- Methods
  - How to think of Method and Functions in Scala?
  - Why mostly everything is a method
  - Definitions, Values, Lazy Values and Variables...
  - How to act like a field?
  - Variables
  - @tailrec and why it matters
  - Using Java annotations
  - Scala's override strictness
- Objects
  - When to use object instead of class?
  - How to use objects and still be able to have testable code?
  - The apply() method
  - The unapply() method
- Case Classes
  - What is a case class?
  - Pattern matching with "match"
  - The @switch annotation
  - Limits of case classes, and how to fallback to normal classes
- Traits
  - What is a Trait?
  - When to use Traits?
  - When NOT to use Traits?
  - Self-Type annotations - a way for Dependency Injection?
  - Traits calling other Traits in the same object instance
  - Interoperating with Java interfaces
- Type Parameters
  - What are they?

- Type Parameters vs. the "type" keyword
- Implicits
  - What does "implicit" mean?
  - How to define implicit values?
  - How are implicits scoped?
  - Implicit conversions
- Parameter lists
  - Why can Scala have more than one parameter list?
  - Implicit parameters
  - Practical use case

## Day 2

- Collections
  - Types of collections
  - How to stick to the immutable nature of Scala with collections?
  - Use parallel collections for quick wins
  - What is CanBuildFrom[\_ , \_ , \_]?
- Functional programming in Scala
  - "The M word" - Monads
  - Monad examples - ( )you've seen them already!
  - Avoiding nulls with Option[\_]
- "The Frameworks"
  - Best known and recommended libraries
  - What kinds of libraries to avoid?
  - ScalaTest and Specs2
- Akka 2.0
  - What inspired Akka?
  - Why and when to use Akka? What other options are there?
  - Example of an Actor driven application
- SBT - Simple Build Tool
  - What is SBT?
  - Tips on sbt versioning and maven inter-op
  - Example of a multi-module project
- Tips for your day-to-day Scala programming
  - When should you just "let it go" and use a hack?
  - Tips for Java inter-op, when Traits don't cut it

- How not get "too smart code"
- When to avoid implicits and other "magic"?
- Scala 2.10 - a brief overview
  - What is a "SIP"?
  - Overview of key features and SIPs with examples for each feature given
  - Tips for handling multiple scala versions with sbt
- Wrap up and QA session

## REQUIREMENTS:

This course is targeted at mature **Java developers**, as a fair background on Java is assumed. Scala builds on top of Java, extending some of it's ideas and adding new ones from functional languages such as Haskell or Lisp - knowledge of those may be useful, but is not required.

## Difficulty level



## CERTIFICATE:

The participants will obtain certificates signed by Compendium Education Center.

## TRAINER:

Compendium's trainer.