

Getting Productive with Scala and Akka in 3 Days

Developer to Developer Training Series





Course Outline

The 3 day Core Scala course is aimed at developers who are encountering Scala for the first time and want to understand how their knowledge of objectoriented and functional programming idioms carries over to and is deepened in this exciting new language. It also covers how to get started with Akka which is becoming the de facto for any enterprise scale Scala development.

It covers Scala from basic syntax and language constructs, through its distinctive objectoriented characteristics and on to functional programming styles and pattern matching as applied to Scala's extensive collections framework.

You would also learn how to apply the tools and services provided by Akka to build high performance, faulttolerant systems that scale.

The course has a practical focus, mixing presentation with indepth handson labs and exercises.



Proposed Structure

Day 1	Day 2
First brush	Case classes and pattern matching
<ul style="list-style-type: none">◆ Introducing the syntax◆ Primitive and simple types◆ Control constructs◆ Classes and case classes◆ Traits◆ Objects◆ Functions	<ul style="list-style-type: none">◆ Structural recursion◆ Kinds of pattern◆ Matching with Lists and Options◆ Pattern matching vs. higherorder functions◆ Pattern matching in unexpected places◆ Pattern matching generalized
Introducing Collections	Functional programming in depth
<ul style="list-style-type: none">◆ Mutability & immutability◆ Lists & Buffers◆ Sets & Maps◆ Arrays	<ul style="list-style-type: none">◆ Byname parameters◆ Local methods◆ Recursion and tail recursion◆ Partial functions

<ul style="list-style-type: none"> ◆ For comprehensions 	<ul style="list-style-type: none"> ◆ Currying
<ul style="list-style-type: none"> ◆ Interoperability with Java 	<ul style="list-style-type: none"> ◆ Partially applied functions
	<ul style="list-style-type: none"> ◆ The power of folding
Collections and functional programming	Starting Akka
<ul style="list-style-type: none"> ◆ Streams 	<ul style="list-style-type: none"> ◆ Concurrency, scalability and fault tolerance
<ul style="list-style-type: none"> ◆ Options 	<ul style="list-style-type: none"> ◆ Understanding Actors
<ul style="list-style-type: none"> ◆ Revisiting for comprehensions 	
<ul style="list-style-type: none"> ◆ Higherorder functions 	
<ul style="list-style-type: none"> ◆ Monads 	
Objectoriented programming in Scala	Day 3
<ul style="list-style-type: none"> ◆ Objects & modules 	<ul style="list-style-type: none"> ◆ Test Driven Development with Actors
<ul style="list-style-type: none"> ◆ Traits and mixin composition 	<ul style="list-style-type: none"> ◆ Actor lifecycle
<ul style="list-style-type: none"> ◆ Self types 	<ul style="list-style-type: none"> ◆ Fault Tolerance with Actors
<ul style="list-style-type: none"> ◆ Parameterized and abstract types 	<ul style="list-style-type: none"> ◆ Messaging with Akka
<ul style="list-style-type: none"> ◆ Structural types 	<ul style="list-style-type: none"> ◆ Understanding Dispatchers
	<ul style="list-style-type: none"> ◆ Understanding Routers
	<ul style="list-style-type: none"> ◆ Distributed communication with Akka
	<ul style="list-style-type: none"> ◆ Clustering basics



Course Prerequisites

To benefit from this course you should have 12 years experience with an objectoriented (eg. Java, C#) or functional (eg. Haskell, Scheme) programming language and a good general understanding of objectoriented or functional programming language concepts. Knowledge of any distributed system is a plus.

- ◆ For more information on the course or a discussion on your custom need, send a mail to info@knoldus.com