

Getting Productive with Scala 2 Days

Developer to Developer Training Series





Course Outline

The 2 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 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.

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◆ For comprehensions◆ Interoperability with Java	<ul style="list-style-type: none">◆ Byname parameters◆ Local methods◆ Recursion and tail recursion◆ Partial functions◆ Currying◆ Partially applied functions◆ The power of folding

Collections and functional programming	
◆ Streams	
◆ Options	
◆ Revisiting for comprehensions	
◆ Higherorder functions	
◆ Monads	
Objectoriented programming in Scala	
◆ Objects & modules	
◆ Traits and mixin composition	
◆ Self types	
◆ Parameterized and abstract types	
◆ Structural types	



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.

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