

Extracted from:

Functional Programming: A PragPub Anthology

Exploring Clojure, Elixir, Haskell, Scala, and Swift

This PDF file contains pages extracted from *Functional Programming: A PragPub Anthology*, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit <http://www.pragprog.com>.

Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printed versions; the content is otherwise identical.

Copyright © 2017 The Pragmatic Programmers, LLC.

All rights reserved.

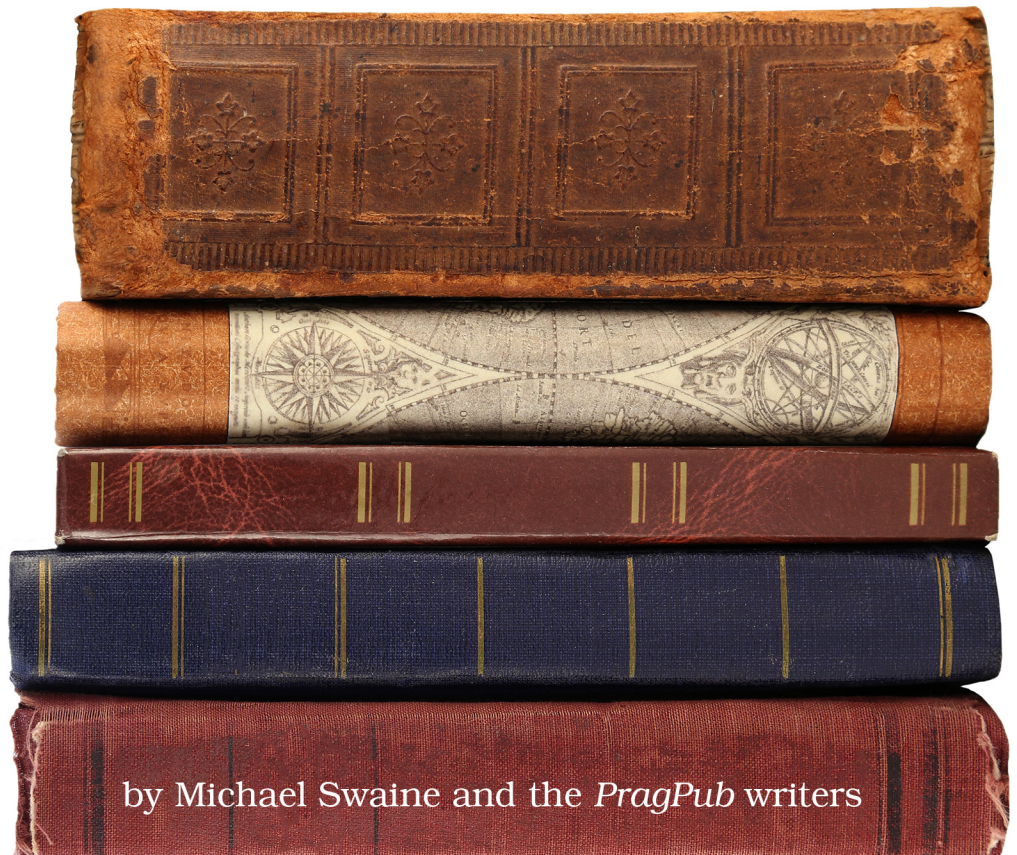
No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

The Pragmatic Bookshelf

Raleigh, North Carolina

Functional Programming: *A PragPub* Anthology

Exploring Clojure, Elixir,
Haskell, Scala, and Swift



by Michael Swaine and the *PragPub* writers

Functional Programming: A PragPub Anthology

Exploring Clojure, Elixir, Haskell, Scala, and Swift

Michael Swaine
and the PragPub writers

The Pragmatic Bookshelf

Raleigh, North Carolina



Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and The Pragmatic Programmers, LLC was aware of a trademark claim, the designations have been printed in initial capital letters or in all capitals. The Pragmatic Starter Kit, The Pragmatic Programmer, Pragmatic Programming, Pragmatic Bookshelf, PragProg and the linking *g* device are trademarks of The Pragmatic Programmers, LLC.

Every precaution was taken in the preparation of this book. However, the publisher assumes no responsibility for errors or omissions, or for damages that may result from the use of information (including program listings) contained herein.

Our Pragmatic books, screencasts, and audio books can help you and your team create better software and have more fun. Visit us at <https://pragprog.com>.

The team that produced this book includes:

Publisher: Andy Hunt

VP of Operations: Janet Furlow

Executive Editor: Susannah Davidson Pfalzer

Indexing: Potomac Indexing, LLC

Copy Editor: Nicole Abramowitz

Layout: Gilson Graphics

For sales, volume licensing, and support, please contact support@pragprog.com.

For international rights, please contact rights@pragprog.com.

Copyright © 2017 The Pragmatic Programmers, LLC.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

Printed in the United States of America.

ISBN-13: 978-1-68050-233-6

Encoded using the finest acid-free high-entropy binary digits.

Book version: P1.0—July 2017

Introduction

by Michael Swaine

This book shows how five different languages approach the paradigm of functional programming. The chapters were written by experts in the different languages and first appeared as articles in *PragPub* magazine. After publishing nearly one hundred issues of the magazine, it became clear that we were in possession of a wealth of information about functional programming, and we decided that some of the best articles would make an interesting book.

Functional programming is one of the major paradigms of programming. In functional programming, we approach computation as the evaluation of mathematical functions, and as much as possible we avoid changing state and mutable data.

Certain concepts and issues are sure to come up in any discussion of functional programming. Recursion. Lazy evaluation. Referential transparency. Eliminating side effects. Functions as first-class objects. Higher-level functions. Currying. Immutable data. Type systems. Pattern matching. The authors touch on all these concepts, looking at them from the perspective of different languages.

But functional programming is not an all-or-none thing. It is perfectly reasonable to write imperative code that uses functional techniques and practices and data structures. It is fine to mix and match styles, and some programming languages are designed as hybrids, allowing you to use the style that best fits your needs of the moment. Scala, for example, or Mathematica, or Swift. Our authors explore these different approaches in this book, and you can decide which works best for you.

We explore functional programming in five languages in this book, with articles by experts in each language. If this encourages you to look for a definitive guide to some of the languages, we can recommend Venkat Subramaniam's [Pragmatic Scala \[Sub15\]](#), Stuart Halloway and Aaron Bedra's [Programming Clojure \(2nd edition\) \[HB12\]](#), Dave Thomas's [Programming Elixir 1.3 \[Tho16\]](#),

Chris Eidhof, Florian Kugler, and Wouter Swiersta’s *Functional Swift* (<https://www.objc.io/books/functional-swift/>), and Miran Lipovaca’s *Learn You a Haskell for Great Good!: A Beginner’s Guide [Lip11]*. What you’ll find here is an exploration of an important programming paradigm in five languages, written by a team of experts. That’s what we set out to create.

I hope you will agree that the result is an interesting book.