Extracted from:

Programming Clojure

Second Edition

This PDF file contains pages extracted from *Programming Clojure*, 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 printer versions; the content is otherwise identical.

Copyright © 2010 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.

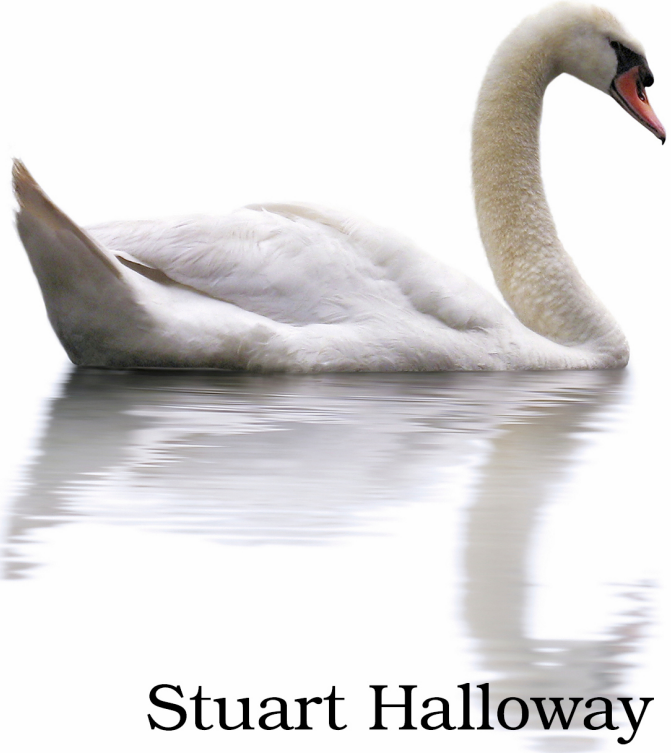
The Pragmatic Bookshelf

Dallas, Texas • Raleigh, North Carolina

The Pragmatic Programmers

Programming Clojure

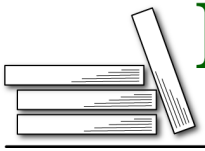
Second Edition



Stuart Halloway
Aaron Bedra

Foreword by Rich Hickey, creator of Clojure

Edited by Michael Swaine



Pragmatic Bookshelf

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 courses, workshops, and other products can help you and your team create better software and have more fun. For more information, as well as the latest Pragmatic titles, please visit us at *http://pragprog.com*.

The team that produced this book includes:

Michael Swaine (editor)
Potomac Indexing, LLC (indexer)
Kim Wimpsett (copyeditor)
David J Kelly (typesetter)
Janet Furlow (producer)
Juliet Benda (rights)
Ellie Callahan (support)

Copyright © 2012 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: 978-1-934356-86-9
Encoded using the finest acid-free high-entropy binary digits.
Book version: P1.0—April 2012

Contents

Foreword for the Second Edition . . . . . ?

Foreword for the First Edition . . . . . ?

Acknowledgments . . . . . ?

Preface . . . . . ?

1. Getting Started . . . . . ?
   - 1.1 Why Clojure? ?
   - 1.2 Clojure Coding Quick Start ?
   - 1.3 Exploring Clojure Libraries ?
   - 1.4 Wrapping Up ?

2. Exploring Clojure . . . . . ?
   - 2.1 Forms ?
   - 2.2 Reader Macros ?
   - 2.3 Functions ?
   - 2.4 Vars, Bindings, and Namespaces ?
   - 2.5 Calling Java ?
   - 2.6 Flow Control ?
   - 2.7 Where's My for Loop? ?
   - 2.8 Metadata ?
   - 2.9 Wrapping Up ?

3. Unifying Data with Sequences . . . . . ?
   - 3.1 Everything Is a Sequence ?
   - 3.2 Using the Sequence Library ?
   - 3.3 Lazy and Infinite Sequences ?
   - 3.4 Clojure Makes Java Seq-able ?
   - 3.5 Calling Structure-Specific Functions ?
   - 3.6 Wrapping Up ?

4. **Functional Programming** . . . . . ?
4.1 Functional Programming Concepts ?
4.2 How to Be Lazy ?
4.3 Lazier Than Lazy ?
4.4 Recursion Revisited ?
4.5 Wrapping Up ?

5. **State** . . . . . ?
5.1 Concurrency, Parallelism, and Locking ?
5.2 Refs and Software Transactional Memory ?
5.3 Use Atoms for Uncoordinated, Synchronous Updates ?
5.4 Use Agents for Asynchronous Updates ?
5.5 Managing Per-Thread State with Vars ?
5.6 A Clojure Snake ?
5.7 Wrapping Up ?

6. **Protocols and Datatypes** . . . . . ?
6.1 Programming to Abstractions ?
6.2 Interfaces ?
6.3 Protocols ?
6.4 Datatypes ?
6.5 Records ?
6.6 reify ?
6.7 Wrapping Up ?

7. **Macros** . . . . . ?
7.1 When to Use Macros ?
7.2 Writing a Control Flow Macro ?
7.3 Making Macros Simpler ?
7.4 Taxonomy of Macros ?
7.5 Wrapping Up ?

8. **Multimethods** . . . . . ?
8.1 Living Without Multimethods ?
8.2 Defining Multimethods ?
8.3 Moving Beyond Simple Dispatch ?
8.4 Creating Ad Hoc Taxonomies ?
8.5 When Should I Use Multimethods? ?
8.6 Wrapping Up ?

9. Java Down and Dirty . . . . . ?
9.1 Exception Handling ?
9.2 Wrestling with the Integers ?
9.3 Optimizing for Performance ?
9.4 Creating Java Classes in Clojure ?
9.5 A Real-World Example ?
9.6 Wrapping Up ?

10. Building an Application . . . . . ?
10.1 Scoring a Clojurebreaker Game ?
10.2 Testing the Scorer ?
10.3 test.generative ?
10.4 Creating an Interface ?
10.5 Deploying Your Code ?
10.6 Farewell ?

A1. Editor Support . . . . . ?

A2. Bibliography . . . . . ?

Index . . . . . ?