

Extracted from:

Programming Groovy 2

Dynamic Productivity for the Java Developer

This PDF file contains pages extracted from *Programming Groovy 2*, 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 © 2013 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.

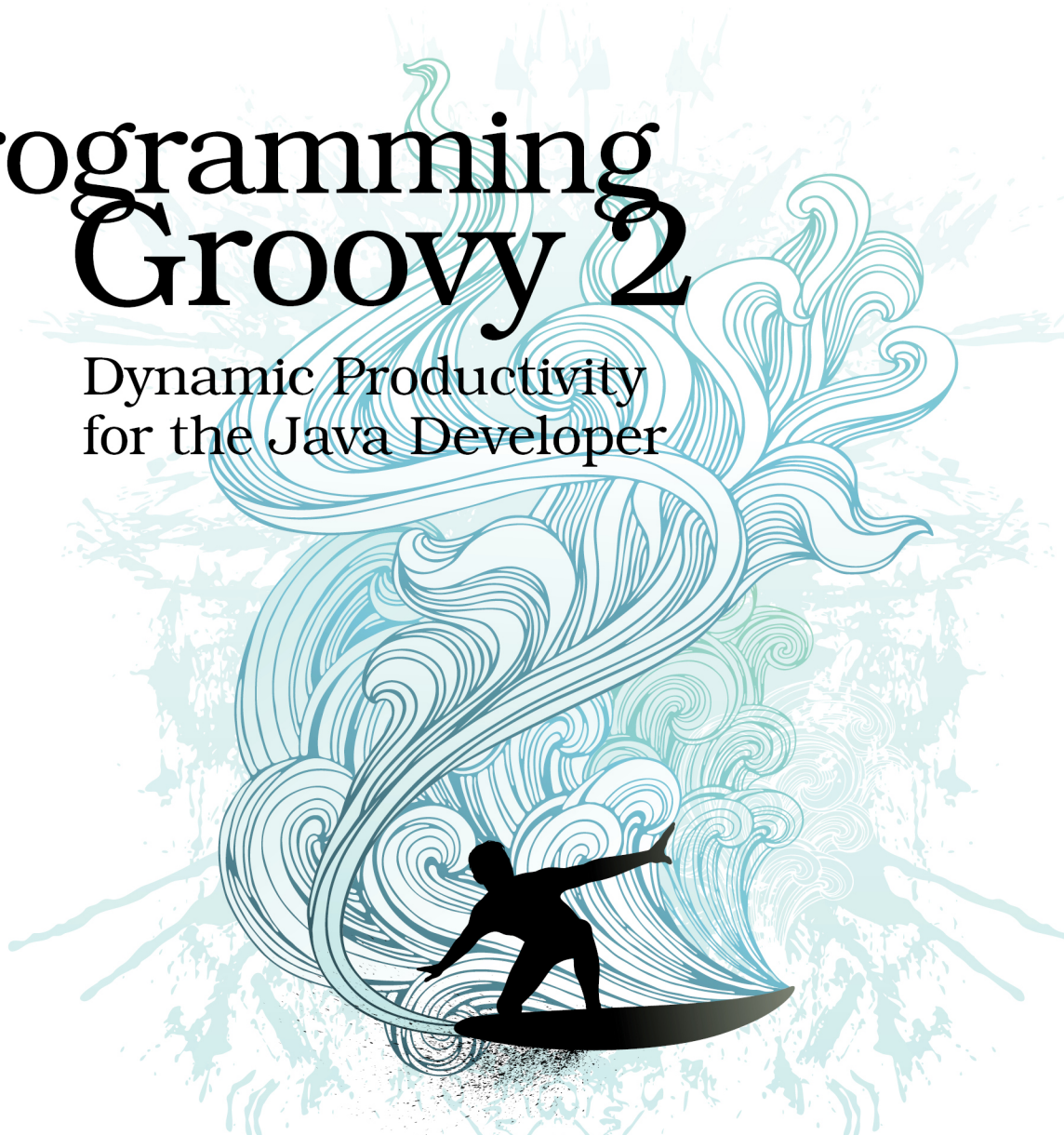
The Pragmatic Bookshelf

Dallas, Texas • Raleigh, North Carolina

The
Pragmatic
Programmers

Programming Groovy 2

Dynamic Productivity
for the Java Developer



Venkat Subramaniam

Foreword by Guillaume Laforge

Edited by Brian P. Hogan

Programming Groovy 2

Dynamic Productivity for the Java Developer

Venkat Subramaniam

The Pragmatic Bookshelf

Dallas, Texas • 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 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:

Brian P. Hogan (editor)
Potomac Indexing, LLC (indexer)
Candace Cunningham (copyeditor)
David J Kelly (typesetter)
Janet Furlow (producer)
Juliet Benda (rights)
Ellie Callahan (support)

Copyright © 2013 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: 978-1-937785-30-7
Encoded using the finest acid-free high-entropy binary digits.
Book version: P1.0—July 2013

*To Mythili and Balu—for being much more
than an aunt and an uncle—for being there
when I needed them most.*

Contents

Foreword to the Second Edition ?

Introduction ?

Part I — Beginning Groovy

1. **Getting Started** ?

1.1 **Installing Groovy** ?

1.2 **Installing and Managing Groovy Versions** ?

1.3 **Test-Drive Using groovysh** ?

1.4 **Using groovyConsole** ?

1.5 **Running Groovy on the Command Line** ?

1.6 **Using an IDE** ?

2. **Groovy for Java Eyes** ?

2.1 **From Java to Groovy** ?

2.2 **JavaBeans** ?

2.3 **Flexible Initialization and Named Arguments** ?

2.4 **Optional Parameters** ?

2.5 **Using Multiple Assignments** ?

2.6 **Implementing Interfaces** ?

2.7 **Groovy Boolean Evaluation** ?

2.8 **Operator Overloading** ?

2.9 **Support of Java 5 Language Features** ?

2.10 **Using Groovy Code-Generation Transformations** ?

2.11 **Gotchas** ?

3. **Dynamic Typing** ?

3.1 **Typing in Java** ?

3.2 **Dynamic Typing** ?

3.3 **Dynamic Typing != Weak Typing** ?

3.4	Design by Capability	?
3.5	Optional Typing	?
3.6	Multimethods	?
3.7	Dynamic: To Be or Not to Be?	?
3.8	Switching Off Dynamic Typing	?
4.	Using Closures	?
4.1	The Convenience of Closures	?
4.2	Programming with Closures	?
4.3	Ways to Use Closures	?
4.4	Passing Parameters to Closures	?
4.5	Using Closures for Resource Cleanup	?
4.6	Closures and Coroutines	?
4.7	Curried Closure	?
4.8	Dynamic Closures	?
4.9	Closure Delegation	?
4.10	Programming with Tail Recursion	?
4.11	Improving Performance Using Memoization	?
5.	Working with Strings	?
5.1	Literals and Expressions	?
5.2	GString Lazy-Evaluation Problem	?
5.3	Multiline Strings	?
5.4	String Convenience Methods	?
5.5	Regular Expressions	?
6.	Working with Collections	?
6.1	Using List	?
6.2	Iterating Over an ArrayList	?
6.3	Using Finder Methods	?
6.4	Other Convenience Methods on Lists	?
6.5	Using the Map Class	?
6.6	Iterating Over Map	?
6.7	Other Convenience Methods on Maps	?

Part II — Using Groovy

7.	Exploring the GDK	?
7.1	Using Object Extensions	?
7.2	Other Extensions	?
7.3	Custom Methods Using the Extension Modules	?

8.	Working with XML	?
8.1	Parsing XML	?
8.2	Creating XML	?
9.	Working with Databases	?
9.1	Setting Up the Database	?
9.2	Connecting to a Database	?
9.3	Database Select	?
9.4	Transforming Data to XML	?
9.5	Using DataSet	?
9.6	Inserting and Updating	?
9.7	Accessing Microsoft Excel	?
10.	Working with Scripts and Classes	?
10.1	The Melting Pot of Java and Groovy	?
10.2	Running Groovy	?
10.3	Using Groovy Classes from Groovy	?
10.4	Intermixing Groovy and Java with Joint Compilation	?
10.5	Creating and Passing Groovy Closures from Java	?
10.6	Calling Groovy Dynamic Methods from Java	?
10.7	Using Java Classes from Groovy	?
10.8	Using Groovy Scripts from Groovy	?
10.9	Using Groovy Scripts from Java	?

Part III — MOPping Groovy

11.	Exploring Metaobject Protocol (MOP)	?
11.1	Groovy Object	?
11.2	Querying Methods and Properties	?
11.3	Dynamically Accessing Objects	?
12.	Intercepting Methods Using MOP	?
12.1	Intercepting Methods Using GroovyInterceptable	?
12.2	Intercepting Methods Using MetaClass	?
13.	MOP Method Injection	?
13.1	Injecting Methods Using Categories	?
13.2	Injecting Methods Using ExpandoMetaClass	?
13.3	Injecting Methods into Specific Instances	?
13.4	Injecting Methods Using Mixins	?
13.5	Decorating Classes with Multiple Mixins	?

14.	MOP Method Synthesis	?
14.1	Method Synthesis Using <code>methodMissing</code>	?
14.2	Method Synthesis Using <code>ExpandoMetaClass</code>	?
14.3	Synthesizing Methods for Specific Instances	?
15.	MOPping Up	?
15.1	Creating Dynamic Classes with <code>Expando</code>	?
15.2	Method Delegation: Putting It All Together	?
15.3	Review of MOP Techniques	?
16.	Applying Compile-Time Metaprogramming	?
16.1	Analyzing Code at Compile Time	?
16.2	Intercepting Method Calls Using AST Transformations	?
16.3	Injecting Methods Using AST Transformations	?

Part IV — Using Metaprogramming

17.	Groovy Builders	?
17.1	Building XML	?
17.2	Building JSON	?
17.3	Building Swing	?
17.4	Custom Builder Using Metaprogramming	?
17.5	Using <code>BuilderSupport</code>	?
17.6	Using <code>FactoryBuilderSupport</code>	?
18.	Unit Testing and Mocking	?
18.1	Code in This Book and Automated Unit Tests	?
18.2	Unit Testing Java and Groovy Code	?
18.3	Testing for Exceptions	?
18.4	Mocking	?
18.5	Mocking by Overriding	?
18.6	Mocking Using Categories	?
18.7	Mocking Using <code>ExpandoMetaClass</code>	?
18.8	Mocking Using <code>Expando</code>	?
18.9	Mocking Using <code>Map</code>	?
18.10	Mocking Using the Groovy Mock Library	?
19.	Creating DSLs in Groovy	?
19.1	Context	?
19.2	Fluency	?
19.3	Types of DSLs	?

19.4	Designing Internal DSLs	?
19.5	Groovy and DSLs	?
19.6	Using Command-Chain Fluency	?
19.7	Closures and DSLs	?
19.8	Method Interception and DSLs	?
19.9	The Parentheses Limitation and a Workaround	?
19.10	Categories and DSLs	?
19.11	ExpandoMetaClass and DSLs	?
A1.	Web Resources	?
A2.	Bibliography	?
	Index	?