

THE EXPERT'S VOICE® IN OPEN SOURCE

# Beginning Ruby

## From Novice to Professional

*An instructional guide to the Ruby programming language.*

Peter Cooper

*Foreword by why the lucky stiff*

Apress®

# Beginning Ruby

## From Novice to Professional



Peter Cooper

## **Beginning Ruby: From Novice to Professional**

**Copyright © 2007 by Peter Cooper**

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN-13 (pbk): 978-1-59059-766-8

ISBN-10 (pbk): 1-59059-766-4

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Trademarked names may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, we use the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Lead Editors: Jonathan Gennick, Keir Thomas

Technical Reviewers: Tim Fletcher, Peter Marklund

Editorial Board: Steve Anglin, Ewan Buckingham, Gary Cornell, Jason Gilmore, Jonathan Gennick,  
Jonathan Hassell, James Huddleston, Chris Mills, Matthew Moodie, Jeff Pepper, Paul Sarknas,  
Dominic Shakeshaft, Jim Sumser, Matt Wade

Project Manager: Beth Christmas

Copy Edit Manager: Nicole Flores

Copy Editor: Susannah Davidson Pfalzer

Assistant Production Director: Kari Brooks-Copony

Production Editor: Lori Bring

Compositor: Gina Rexrode

Proofreader: Nancy Sixsmith

Indexer: Julie Grady

Artist: April Milne

Cover Designer: Kurt Krames

Manufacturing Director: Tom Debolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail [orders-ny@springer-sbm.com](mailto:orders-ny@springer-sbm.com), or visit <http://www.springeronline.com>.

For information on translations, please contact Apress directly at 2560 Ninth Street, Suite 219, Berkeley, CA 94710. Phone 510-549-5930, fax 510-549-5939, e-mail [info@apress.com](mailto:info@apress.com), or visit <http://www.apress.com>.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

The source code for this book is available to readers at <http://www.apress.com> in the Source Code/Download section.

*For Laura*



# Contents at a Glance

Foreword .....	xix
About the Author .....	xxiii
About the Technical Reviewers .....	xxv
Acknowledgments .....	xxvii
Introduction .....	xxix

## PART 1 ■ ■ ■ Foundations and Scaffolding

■ CHAPTER 1	Let's Get It Started: Installing Ruby .....	3
■ CHAPTER 2	Programming == Joy: A Whistle-Stop Tour of Ruby and Object Orientation .....	15
■ CHAPTER 3	Ruby's Building Blocks: Data, Expressions, and Flow Control .....	35
■ CHAPTER 4	Developing a Basic Ruby Application .....	87
■ CHAPTER 5	The Ruby Ecosystem .....	113

## PART 2 ■ ■ ■ The Core of Ruby

■ CHAPTER 6	Classes, Objects, and Modules .....	129
■ CHAPTER 7	Projects and Libraries .....	181
■ CHAPTER 8	Documentation, Error Handling, Debugging, and Testing .....	201
■ CHAPTER 9	Files and Databases .....	229
■ CHAPTER 10	Deploying Ruby Applications and Libraries .....	279
■ CHAPTER 11	Advanced Ruby Features .....	309
■ CHAPTER 12	Tying It Together: Developing a Larger Ruby Application .....	341

## PART 3 ■ ■ ■ Ruby Online

■ CHAPTER 13	Ruby on Rails: Ruby's Killer App .....	387
■ CHAPTER 14	Ruby and the Internet .....	433
■ CHAPTER 15	Networking, Sockets, and Daemons .....	467
■ CHAPTER 16	Useful Ruby Libraries and Gems .....	493

■ APPENDIX A	Ruby Primer and Review for Developers . . . . .	549
■ APPENDIX B	Ruby Reference. . . . .	581
■ APPENDIX C	Useful Resources . . . . .	601
■ INDEX	. . . . .	611

# Contents

Foreword .....	xix
About the Author .....	xxiii
About the Technical Reviewers .....	xxv
Acknowledgments .....	xxvii
Introduction .....	xxix

## PART 1 ■ ■ ■ Foundations and Scaffolding

<b>CHAPTER 1</b>	<b>Let's Get It Started: Installing Ruby</b> .....	3
	Installing Ruby .....	4
	Windows .....	4
	Apple Mac OS X .....	7
	Linux .....	10
	Other Platforms .....	12
	Summary .....	13
<b>CHAPTER 2</b>	<b>Programming == Joy: A Whistle-Stop Tour of Ruby and Object Orientation</b> .....	15
	Baby Steps .....	16
	irb: Interactive Ruby .....	16
	Ruby Is English for Computers .....	17
	Why Ruby Makes a Great Programming Language .....	17
	Trails for the Mind .....	18
	Turning Ideas into Ruby Code .....	20
	How Ruby Understands Concepts with Objects and Classes .....	20
	The Making of a Man .....	21
	Basic Variables .....	23
	From People to Pets .....	24
	Everything Is an Object .....	27
	Kernel Methods .....	29
	Passing Data to Methods .....	29
	Using the Methods of the String Class .....	31
	Using Ruby Without Object Orientation .....	32
	Summary .....	33



<b>CHAPTER 3</b>	<b>Ruby's Building Blocks: Data, Expressions, and Flow Control</b>	<b>35</b>
	Numbers and Expressions	35
	Basic Expressions	35
	Variables	36
	Comparison Operators and Expressions	38
	Looping Through Numbers with Blocks and Iterators	40
	Floating Point Numbers	42
	Constants	43
	Text and Strings	44
	String Literals	44
	String Expressions	46
	Interpolation	47
	String Methods	49
	Regular Expressions and String Manipulation	50
	Arrays and Lists	57
	Basic Arrays	58
	Splitting Strings into Arrays	60
	Array Iteration	61
	Other Array Methods	62
	Hashes	65
	Basic Hash Methods	66
	Hashes Within Hashes	67
	Flow Control	68
	if and unless	69
	?., The Ternary Operator	70
	elsif and case	71
	while and until	72
	Code Blocks	74
	Other Useful Building Blocks	76
	Dates and Times	76
	Large Numbers	80
	Ranges	81
	Symbols	83
	Converting Between Classes	84
	Summary	85
<b>CHAPTER 4</b>	<b>Developing a Basic Ruby Application</b>	<b>87</b>
	Working with Source Code Files	87
	Creating a Test File	88
	The Test Source Code File	90
	Running Your Source Code	90

Our Application: A Text Analyzer . . . . .	93
Required Basic Features . . . . .	94
Building the Basic Application . . . . .	94
Obtaining Some Dummy Text . . . . .	95
Loading Text Files and Counting Lines . . . . .	95
Counting Characters . . . . .	97
Counting Words . . . . .	98
Counting Sentences and Paragraphs . . . . .	100
Calculating Averages . . . . .	102
The Source Code So Far . . . . .	102
Adding Extra Features . . . . .	103
Percentage of “Useful” Words . . . . .	103
Summarizing by Finding “Interesting” Sentences . . . . .	105
Analyzing Files Other Than text.txt . . . . .	107
The Completed Program . . . . .	108
Summary . . . . .	111

## ■ CHAPTER 5    **The Ruby Ecosystem** . . . . . 113

Ruby’s History . . . . .	113
The Land of the Rising Sun . . . . .	114
Ruby’s Influences . . . . .	115
Go West . . . . .	115
Ruby on Rails . . . . .	117
Why Rails Came into Existence . . . . .	118
How the Web (2.0) Was Won . . . . .	119
The Open Source Culture . . . . .	119
What Is Open Source? . . . . .	120
Where and How to Get Help . . . . .	121
Mailing Lists . . . . .	121
Usenet Newsgroups . . . . .	121
Internet Relay Chat (IRC) . . . . .	122
Documentation . . . . .	123
Forums . . . . .	124
Joining the Community . . . . .	124
Give Help to Others . . . . .	124
Contribute Code . . . . .	125
Weblogs . . . . .	125
Summary . . . . .	126

## PART 2 ■ ■ ■ The Core of Ruby

<b>CHAPTER 6</b>	<b>Classes, Objects, and Modules</b>	129
	Why Use Object Orientation?	129
	Object Orientation Basics	133
	Local, Global, Object, and Class Variables	133
	Class Methods vs. Object Methods	138
	Inheritance	140
	Overriding Existing Methods	143
	Reflection and Discovering an Object's Methods	145
	Encapsulation	146
	Polymorphism	151
	Nested Classes	153
	The Scope of Constants	154
	Modules, Namespaces, and Mix-Ins	156
	Namespaces	156
	Mix-Ins	159
	Building a Dungeon Text Adventure with Objects	168
	Dungeon Concepts	168
	Creating the Initial Classes	168
	Structs: Quick and Easy Data Classes	170
	Creating Rooms	173
	Making the Dungeon Work	173
	Summary	178
<b>CHAPTER 7</b>	<b>Projects and Libraries</b>	181
	Projects and Using Code from Other Files	181
	Basic File Inclusion	181
	Inclusions from Other Directories	184
	Logic and Including Code	185
	Nested Inclusions	185
	Libraries	186
	The Standard Libraries	187
	RubyGems	189
	Summary	198
<b>CHAPTER 8</b>	<b>Documentation, Error Handling, Debugging, and Testing</b>	201
	Documentation	201
	Generating Documentation with RDoc	202
	RDoc Techniques	203

Debugging and Errors . . . . .	208
Exceptions and Error Handling . . . . .	208
Catch and Throw . . . . .	212
The Ruby Debugger . . . . .	213
Testing . . . . .	216
The Philosophy of Test-Driven Development . . . . .	217
Unit Testing . . . . .	219
More Test::Unit Assertions . . . . .	221
Benchmarking and Profiling . . . . .	222
Simple Benchmarking . . . . .	223
Profiling . . . . .	225
Summary . . . . .	227

## CHAPTER 9   **Files and Databases** . . . . . 229

Input and Output . . . . .	229
Keyboard Input . . . . .	230
File I/O . . . . .	231
Basic Databases . . . . .	248
Text File Databases . . . . .	248
Storing Objects and Data Structures . . . . .	251
Relational Databases and SQL . . . . .	255
Relational Database Concepts . . . . .	256
The Big Four: MySQL, PostgreSQL, Oracle, and SQLite . . . . .	257
Installing SQLite . . . . .	258
A Crash Course in Basic Database Actions and SQL . . . . .	259
Using SQLite with Ruby . . . . .	264
Connecting to Other Database Systems . . . . .	269
ActiveRecord: A Sneak Peek . . . . .	274
Summary . . . . .	275

## CHAPTER 10   **Deploying Ruby Applications and Libraries** . . . . . 279

Distributing Basic Ruby Programs . . . . .	279
The Shebang Line . . . . .	280
Associated File Types in Windows . . . . .	282
“Compiling” Ruby . . . . .	282
Detecting Ruby’s Runtime Environment . . . . .	284
Easy OS Detection with RUBY_PLATFORM . . . . .	285
Environment Variables . . . . .	285
Accessing Command Line Arguments . . . . .	287

Distributing and Releasing Ruby Libraries As Gems .....	288
Creating a Gem .....	289
Distributing a Gem .....	293
RubyForge .....	294
Deploying Ruby Applications As Remote Services .....	295
CGI Scripts .....	295
Generic HTTP Servers .....	298
Remote Procedure Calls .....	303
Summary .....	308

## ■ CHAPTER 11    **Advanced Ruby Features** ..... 309

Dynamic Code Execution .....	309
Bindings .....	310
Other Forms of eval .....	311
Creating Your Own Version of attr_accessor .....	314
Running Other Programs from Ruby .....	315
Getting Results from Other Programs .....	315
Transferring Execution to Another Program .....	316
Running Two Programs at the Same Time .....	316
Interacting with Another Program .....	317
Safely Handling Data and Dangerous Methods .....	318
Tainted Data and Objects .....	319
Safe Levels .....	321
Working with Microsoft Windows .....	322
Using the Windows API .....	323
Controlling Windows Programs .....	325
Threads .....	327
Basic Ruby Threads in Action .....	328
Advanced Thread Operations .....	329
RubyInline .....	331
Why Use C As an Inline Language? .....	332
Creating a Basic Method or Function .....	332
Benchmarking C vs. Ruby .....	334
Unicode and UTF-8 Support .....	336
Summary .....	339

## ■ CHAPTER 12    **Tying It Together: Developing a Larger Ruby Application** .. 341

Let's Build a Bot .....	341
What Is a Bot? .....	341
Why a Bot? .....	343
How? .....	343

Creating a Text Processing Tools Library . . . . .	344
Building the WordPlay Library . . . . .	345
Testing the Library . . . . .	351
WordPlay's Source Code . . . . .	354
Building the Bot's Core . . . . .	357
The Program's Life Cycle and Parts . . . . .	358
Bot Data . . . . .	360
Constructing the Bot Class and Data Loader . . . . .	363
The response_to Method . . . . .	365
Playing with the Bot . . . . .	371
Main Bot Code Listings . . . . .	374
bot.rb . . . . .	375
basic_client.rb . . . . .	378
Extending the Bot . . . . .	378
Using Text Files As a Source of Conversation . . . . .	379
Connecting the Bot to the Web . . . . .	379
Bot-to-Bot Conversations . . . . .	382
Summary . . . . .	384

## PART 3 ■ ■ ■ Ruby Online

■ CHAPTER 13 <b>Ruby on Rails: Ruby's Killer App</b> . . . . .	387
First Steps . . . . .	387
What Is Rails and Why Use It? . . . . .	387
Installing Rails . . . . .	389
Database Considerations . . . . .	391
Building a Basic Rails Application . . . . .	391
Creating a Blank Rails Application . . . . .	391
Database Initialization . . . . .	396
Creating a Model and Migrations . . . . .	398
Scaffolding . . . . .	404
Controllers and Views . . . . .	408
Routing . . . . .	418
Model Relationships . . . . .	420
Sessions and Filters . . . . .	422
Other Features . . . . .	424
Layouts . . . . .	424
Testing . . . . .	426
Plugins . . . . .	428
References and Demo Applications . . . . .	429
Reference Sites and Tutorials . . . . .	429
Example Rails Applications . . . . .	430
Summary . . . . .	430

<b>CHAPTER 14</b>	<b>Ruby and the Internet</b>	433
	HTTP and the Web	433
	Downloading Web Pages	433
	Generating Web Pages and HTML	443
	Processing Web Content	448
	E-Mail	454
	Receiving Mail with POP3	454
	Sending Mail with SMTP	456
	Sending Mail with ActionMailer	457
	File Transfers with FTP	458
	Connection and Basic FTP Actions	459
	Downloading Files	461
	Uploading Files	462
	Summary	464
 <b>CHAPTER 15</b>	 <b>Networking, Sockets, and Daemons</b>	 467
	Networking Concepts	467
	TCP and UDP	468
	IP Addresses and DNS	468
	Basic Network Operations	469
	Checking Machine and Service Availability	469
	Performing DNS Queries	471
	Connecting to a TCP Server Directly	474
	Servers and Clients	475
	UDP Client and Server	475
	Building a Simple TCP Server	478
	Multi-Client TCP Servers	480
	GServer	482
	A GServer-Based Chat Server	485
	Web/HTTP Servers	488
	Daemon Processes	488
	Summary	490
 <b>CHAPTER 16</b>	 <b>Useful Ruby Libraries and Gems</b>	 493
	abbrev	494
	Installation	494
	Examples	494
	Further Information	495
	base64	496
	Installation	496
	Examples	496
	Further Information	499

BlueCloth	500
Installation	500
Examples	500
Further Information	501
cgi	502
Installation	502
Examples	502
Further Information	507
chronic	508
Installation	508
Examples	508
Further Information	509
Digest	510
Installation	510
Examples	510
Further Information	512
English	513
Installation	513
Examples	513
Further Information	514
ERB	515
Installation	515
Examples	515
Further Information	518
FasterCSV	519
Installation	519
Examples	519
Further Information	525
iconv	526
Installation	526
Examples	526
Further Information	527
logger	528
Installation	528
Examples	528
Further Information	530
pp	531
Installation	531
Examples	531
Further Information	532
RedCloth	533
Installation	533
Examples	533
Further Information	534



StringScanner . . . . .	535
Installation . . . . .	535
Examples . . . . .	535
Further Information . . . . .	538
tempfile . . . . .	539
Installation . . . . .	539
Examples . . . . .	539
Further Information . . . . .	541
uri . . . . .	542
Installation . . . . .	542
Examples . . . . .	542
Further Information . . . . .	546
zlib . . . . .	547
Installation . . . . .	547
Examples . . . . .	547
Further Information . . . . .	548

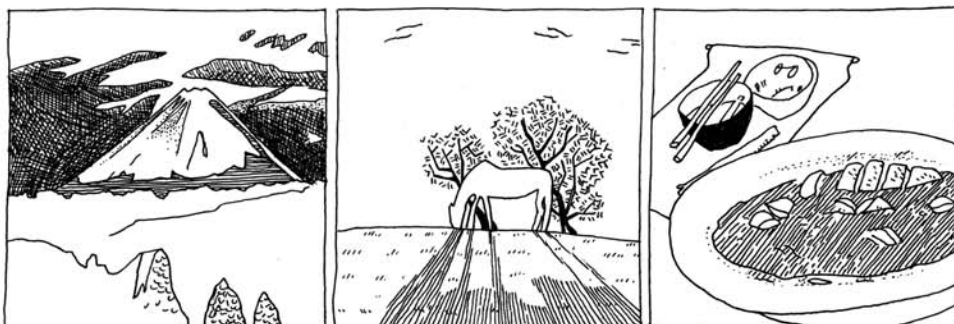
## ■ APPENDIX A    **Ruby Primer and Review for Developers** . . . . . 549

The Basics . . . . .	549
Definition and Concepts . . . . .	549
The Ruby Interpreter and Running Ruby Code . . . . .	551
Interactive Ruby . . . . .	553
Expressions, Logic, and Flow Control . . . . .	553
Basic Expressions . . . . .	553
Class Mismatches . . . . .	554
Comparison Expressions . . . . .	555
Flow . . . . .	557
Object Orientation . . . . .	561
Objects . . . . .	562
Classes and Methods . . . . .	563
Reflection . . . . .	565
Reopening Classes . . . . .	567
Method Visibility . . . . .	568
Data . . . . .	569
Strings . . . . .	569
Regular Expressions . . . . .	569
Numbers . . . . .	572
Arrays . . . . .	574
Hashes (Associative Arrays) . . . . .	574
Complex Structures . . . . .	576
Input/Output . . . . .	576
Files . . . . .	577
Databases . . . . .	577
Web Access . . . . .	578

Libraries .....	579
File Organization .....	579
Packaging .....	580
<b>■ APPENDIX B Ruby Reference .....</b>	<b>581</b>
Useful Classes and Methods .....	581
Array .....	581
Bignum and Fixnum .....	583
Enumerable .....	584
Float .....	585
Hash .....	585
Integer .....	586
Numeric .....	586
Object .....	587
String .....	588
Regular Expression Syntax .....	590
Regular Expression Options .....	591
Special Characters and Formations .....	591
Character and Sub-Expression Suffixes .....	592
Exception Classes .....	592
Special Variables .....	596
Ruby License .....	597
<b>■ APPENDIX C Useful Resources .....</b>	<b>601</b>
References .....	601
Ruby .....	601
Ruby on Rails .....	602
Blogs .....	603
Aggregators and Community Blogs .....	603
Personal Blogs .....	604
Forums and Newsgroups .....	604
Mailing Lists .....	605
Real-Time Chat .....	606
Tutorials and Guides .....	607
Installation .....	607
Ruby and Techniques .....	608
Ruby on Rails .....	609
Other .....	610
<b>■ INDEX .....</b>	<b>611</b>



# Foreword

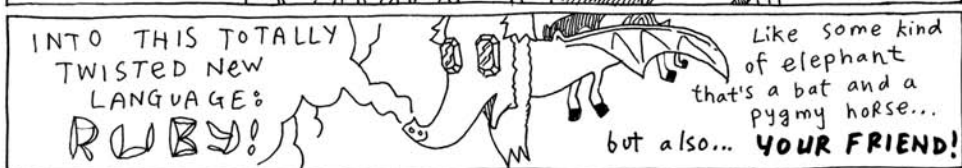
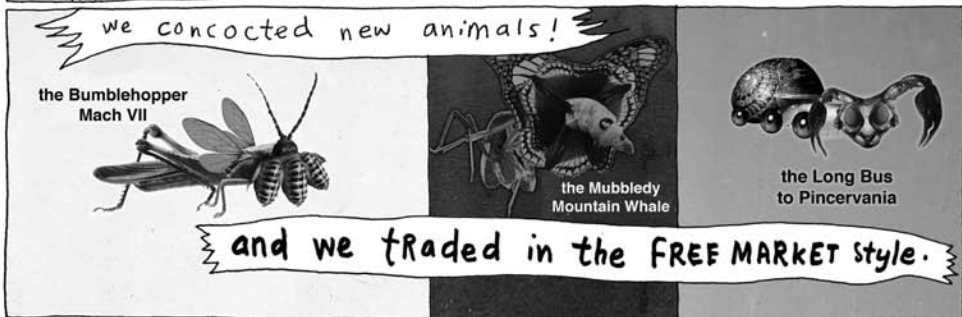
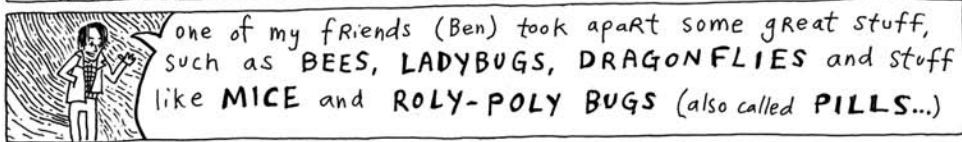


here is a nice mountain in japan.

this one's just a meadow with a horse taking it easy.

oh, nice! a bowl of CURRY!  
⚠️ so delicious...







*why the lucky stiff*

<http://whytheluckystiff.net/>



# About the Author



■ **PETER COOPER** is an experienced Ruby developer and trainer, and editor of Ruby Inside (<http://www.rubyinside.com/>), the most popular Ruby news blog. Until 2007 he was primarily a Ruby trainer and developer, but is now the full-time owner and developer of Feed Digest (<http://www.feeditdigest.com/>), a Ruby- and Rails-powered RSS feed processing and redistribution service that serves more than 200 million requests per month and was recently profiled by *Business 2.0* magazine.

Since 2004 Peter has developed many commercial Web sites using Ruby on Rails, the Ruby-based Web framework. In addition, he created Code Snippets (<http://www.bigbold.com/snippets/>), one of the Web's largest public code repositories, and Congress, an online chat client using Ajax and Ruby on Rails technologies.

In addition to development work, Peter has written professionally about various development techniques and tools, with more than 100 bylines since 1998. He was co-editor of WebDeveloper.com, and worked on iBoost.com and Webpedia.com during the dot-com boom.

He lives in Lincolnshire, England, with his girlfriend. In his limited spare time he enjoys hiking, camping, and exploring.





# About the Technical Reviewers

■ **TIM FLETCHER** is 22 years old and lives in Winchester, England, on a student placement with IBM. He likes Ruby because it's fun. He has no children or pets, but an admirable younger sister called Sophie. When not writing code, he loves to read, eat, sleep, and ski as much as possible.



■ **PETER MARKLUND** has extensive experience with and expertise in object orientation, Web development, relational databases, and testing, and has been doing Web development with Java and Tcl since 2000. He was one of the core developers of the OpenACS open source Web framework. In late 2004, he was introduced to Ruby on Rails and has since helped develop an online community and a CRM system with Rails. Peter is working as a Ruby on Rails freelancer and is also helping organize events for the Ruby on Rails developer community in Stockholm. Peter has a personal blog at <http://marklunds.com>, where he shares Rails tips with other developers.



# Acknowledgments

It is often said that writing is a lonely task, but it's not until you write a book that you realize the process has to be anything *but* lonely. Without the help and reassurance of the large team of people backing this book, and backing me personally, this book could not have been written.

My first thanks go to Keir Thomas, who approached me with the idea of writing a Ruby book. He gave me great freedom over the scope and specification of the book and was the most essential piece of the puzzle in getting the book approved and everything sorted out in the early stages.

Beth Christmas of Apress deserves a special thanks for her superb project management and constant reassurance during the writing of this book. Without her schedules and assurance that everything was on track, I would have been a nervous wreck.

Jonathan Gennick, Tim Fletcher, and Peter Marklund deserve much praise for their seemingly unending reading and rereading of this book's chapters throughout the various stages of their development. As a newcomer to Ruby, Jonathan provided some especially interesting insights that have served to make the book even better for Ruby newcomers to read.

I'd also like to praise Susannah Davidson Pfalzer for her diligent approach to copy editing this book by fixing my pronouns, removing my overuse of words like "however" and "therefore," and generally making it possible to read the book without going insane. As this is my first book for Apress, I have depended on Susannah's deep knowledge of Apress customs a great deal.

Naturally, thanks go to all of those I directly worked with on the book, whether they're from Apress or independent. In no particular order: Jonathan Gennick, Keir Thomas, Beth Christmas, Tim Fletcher, Peter Marklund, Susannah Davidson Pfalzer, Jason Gilmore, Lori Bring, Nancy Sixsmith, and why the lucky stiff.

Separately from the book itself, I have to give thanks to many in the Ruby community for either working alongside me, producing tools I've used, or just making the Ruby language more appealing in general. In no particular order: why the lucky stiff (for an unforgettable foreword), Yukihiro "Matz" Matsumoto, Jamie van Dyke, Amy Hoy, Evan Weaver, Geoffrey Grosenbach, Obie Fernandez, Damien Tanner, Chris Roos, Martin Sadler, Zach Dennis, Pat Toner, Pat Eyler, Hendy Irawan, Ian Ozsvald, Nic Williams, Shane Vitarana, Josh Catone, Alan Bradburne, Jonathan Conway, Alex MacCaw, Benjamin Curtis, and David Heinemeier Hansson. I am anxious I've missed some names, so if you're missing from this list, I humbly apologize.

Those in my personal life have also supported me a great deal by putting up with my weird work hours and annoying habits, and by asking questions about the book, feeding

me, or just being there to talk to. In this regard I'd like to thank—again in no particular order—Laura Craggs, Clive Cooper, Ann Cooper, David Sculley, Ed Farrow, Michael Wong, Bob Pardoe, Dave Hunt, Chris Ueland, Kelly Smith, Graham Craggs, Lorraine Craggs, and Robert Smith. Laura Craggs deserves a special mention for having had to put up with me nearly 24 hours a day during the writing of this book; she is amazing.

Last, it's necessary to thank *you*, the reader, for choosing to buy this book, for if no one bought this book, these acknowledgments and the efforts of many people during the writing of this book would have been wasted. Thank you!

# Introduction

*I wanted to minimize my frustration during programming, so I want to minimize my effort in programming. That was my primary goal in designing Ruby. I want to have fun in programming myself.*

—Yukihiro Matsumoto (Matz), creator of Ruby

*Ruby is a “best of breed” language that has been assembled from the best and most powerful programming features found in its predecessors.*

—Jim White

*Ruby makes me smile.*

—Amy Hoy (slash7.com)

**R**uby is a fun toy. It's also a serious programming language. Ruby is the jolly uncle who keeps the kids entertained, but who puts in solid 12-hour days at the construction site during the week. To hundreds of thousands of programmers, Ruby has become a good friend, a trusted servant, and has revealed a new way of thinking about programming and software development.

Like the guitar, it's often claimed that Ruby is an easy language to learn and a hard one to master. I'd agree, with some provisions. If you don't know any programming languages already, Ruby will be surprisingly easy to learn. If you already know some languages such as PHP, Perl, BASIC, C, or Pascal, some of the concepts in Ruby will already be familiar to you, but the different perspective Ruby takes with problem solving will probably throw you at first. Like the differences between spoken languages, Ruby differs from most other programming languages not only by syntax, but by culture, grammar, and customs. In fact, Ruby has more in common with more esoteric languages such as LISP and Smalltalk than with better-known languages such as PHP and C++.

While Ruby's roots might be different from other languages, it's heavily used and respected in many industries. Companies that use or support Ruby in one way or another include such prestigious names as Sun Microsystems, Intel, Microsoft, Apple, and Amazon.com. The Ruby on Rails Web framework is a system for developing Web applications that uses Ruby as its base language, and it powers hundreds of large Web sites. Ruby is also used as a generic language from the command prompt, much like Perl.

Grammarians, biochemists, database administrators, and thousands of other professionals and hobbyists use Ruby to make their work easier. Ruby is a truly international language with almost unlimited application.

This book is designed to cater both to people new to programming and those with programming experience in other languages. Ruby's culture is different enough from other languages that most of this book will be of use to both groups. Any large sections that can be skipped by already proficient programmers are noted in the text. In any case, I'd suggest that all programmers at least speed-read the sections that might seem obvious to them, as there are some surprising ways in which Ruby is different from what you've done before.

When reading this book be prepared for a little informality, some quirky examples, and a heavy dose of pragmatism. Ruby is an extremely pragmatic language, less concerned with formalities and more concerned with ease of development and valid results. From time to time I'll show you how you can do things the "wrong" way in Ruby, merely for illustrative purposes, but mostly you'll be working with code that does things "the Ruby way." When I started to learn Ruby I learned primarily by example, and with a language as original and idiomatic as Ruby, it's the easiest way to pick up good habits for the future. However, there's always "more than one way to do it," so if you think some code in this book could be rewritten in a different way that fits in more with your way of thinking, try it out!

As you start this book, be prepared to think in new ways, and to feel motivated to start coding for both fun and profit. Ruby has helped a lot of jaded developers become productive once again, and whether you're a beginner to programming or one of those jaded programmers, it's almost inevitable that you'll see how Ruby can be both fun and productive for you.

Last, if you're coming from other modern scripting languages such as Perl, PHP, or Python, you might want to jump to Appendix A before reading Chapter 1. It covers the key differences between Ruby and other scripting languages, which might help you move through the initial chapters of this book more easily.

Good luck, and I hope you enjoy this book. I'll see you in Chapter 1.