

Addison-Wesley Professional Ruby Series



"Sticking to its tried and tested formula of cutting right to the techniques the modern day Rubyist needs to know, the latest edition of The Ruby Way keeps its strong reputation going for the latest generation of the Ruby language."

—PETER COOPER, Editor of Ruby Weekly

THE RUBY WAY

THIRD EDITION

HAL FULTON
with **ANDRÉ ARKO**

Praise for *The Ruby Way*, Third Edition

“Sticking to its tried and tested formula of cutting right to the techniques the modern day Rubyist needs to know, the latest edition of *The Ruby Way* keeps its strong reputation going for the latest generation of the Ruby language.”

Peter Cooper
Editor of *Ruby Weekly*

“The authors’ excellent work and meticulous attention to detail continues in this latest update; this book remains an outstanding reference for the beginning Ruby programmer—as well as the seasoned developer who needs a quick refresh on Ruby. Highly recommended for anyone interested in Ruby programming.”

Kelvin Meeks
Enterprise Architect

Praise for Previous Editions of *The Ruby Way*

“Among other things, this book excels at explaining metaprogramming, one of the most interesting aspects of Ruby. Many of the early ideas for Rails were inspired by the first edition, especially what is now Chapter 11. It puts you on a rollercoaster ride between ‘How could I use this?’ and ‘This is so cool!’ Once you get on that rollercoaster, there’s no turning back.”

David Heinemeier Hansson
Creator of Ruby on Rails,
Founder at Basecamp

“The appearance of the second edition of this classic book is an exciting event for Rubyists—and for lovers of superb technical writing in general. Hal Fulton brings a lively erudition and an engaging, lucid style to bear on a thorough and meticulously exact exposition of Ruby. You palpably feel the presence of a teacher who knows a tremendous amount and really wants to help you know it too.”

David Alan Black
Author of *The Well-Grounded Rubyist*

Ruby Way, The: Solutions and Techniques in Ruby Programming, Portable Documents

Table of Contents

Contents

Foreword

Acknowledgments

About the Authors

Introduction

1 Ruby in Review

1.1 An Introduction to Object Orientation

1.1.1 What Is an Object?

1.1.2 Inheritance

1.1.3 Polymorphism

1.1.4 A Few More Terms

1.2 Basic Ruby Syntax and Semantics

1.2.1 Keywords and Identifiers

1.2.2 Comments and Embedded Documentation

1.2.3 Constants, Variables, and Types

1.2.4 Operators and Precedence

1.2.5 A Sample Program

1.2.6 Looping and Branching

1.2.7 Exceptions

1.3 OOP in Ruby

Table of Contents

1.3.1 Objects

1.3.2 Built-in Classes

1.3.3 Modules and Mixins

1.3.4 Creating Classes

1.3.5 Methods and Attributes

1.4 Dynamic Aspects of Ruby

1.4.1 Coding at Runtime

1.4.2 Reflection

1.4.3 Missing Methods

1.4.4 Garbage Collection

1.5 Training Your Intuition: Things to Remember

1.5.1 Syntax Issues

1.5.2 Perspectives in Programming

1.5.3 Rubys case Statement

1.5.4 Rubyisms and Idioms

1.5.5 Expression Orientation and Other Miscellaneous Issues

1.6 Ruby Jargon and Slang

1.7 Conclusion

2 Working with Strings

2.1 Representing Ordinary Strings

2.2 Representing Strings with Alternate Notations

2.3 Using Here-Documents

2.4 Finding the Length of a String

2.5 Processing a Line at a Time

2.6 Processing a Character or Byte at a Time

2.7 Performing Specialized String Comparisons

2.8 Tokenizing a String

Table of Contents

- 2.9 Formatting a String
- 2.10 Using Strings as IO Objects
- 2.11 Controlling Uppercase and Lowercase
- 2.12 Accessing and Assigning Substrings
- 2.13 Substituting in Strings
- 2.14 Searching a String
- 2.15 Converting Between Characters and ASCII Codes
- 2.16 Implicit and Explicit Conversion
- 2.17 Appending an Item onto a String
- 2.18 Removing Trailing Newlines and Other Characters
- 2.19 Trimming Whitespace from a String
- 2.20 Repeating Strings
- 2.21 Embedding Expressions within Strings
- 2.22 Delayed Interpolation of Strings
- 2.23 Parsing Comma-Separated Data
- 2.24 Converting Strings to Numbers (Decimal and
Otherwise)
- 2.25 Encoding and Decoding rot13 Text
- 2.26 Encrypting Strings
- 2.27 Compressing Strings
- 2.28 Counting Characters in Strings
- 2.29 Reversing a String
- 2.30 Removing Duplicate Characters
- 2.31 Removing Specific Characters
- 2.32 Printing Special Characters
- 2.33 Generating Successive Strings

Table of Contents

- 2.34 Calculating a 32-Bit CRC
- 2.35 Calculating the SHA-256 Hash of a String
- 2.36 Calculating the Levenshtein Distance Between Two Strings
- 2.37 Encoding and Decoding Base64 Strings
- 2.38 Expanding and Compressing Tab Characters
- 2.39 Wrapping Lines of Text
- 2.40 Conclusion

3 Working with Regular Expressions

- 3.1 Regular Expression Syntax
- 3.2 Compiling Regular Expressions
- 3.3 Escaping Special Characters
- 3.4 Using Anchors
- 3.5 Using Quantifiers
- 3.6 Positive and Negative Lookahead
- 3.7 Positive and Negative Lookbehind
- 3.8 Accessing Backreferences
- 3.9 Named Matches
- 3.10 Using Character Classes
- 3.11 Extended Regular Expressions
- 3.12 Matching a Newline with a Dot
- 3.13 Using Embedded Options
- 3.14 Using Embedded Subexpressions
 - 3.14.1 Recursion in Regular Expressions
- 3.15 A Few Sample Regular Expressions
 - 3.15.1 Matching an IP Address

Table of Contents

- 3.15.2 Matching a Keyword-Value Pair
- 3.15.3 Matching Roman Numerals
- 3.15.4 Matching Numeric Constants
- 3.15.5 Matching a Date/Time String
- 3.15.6 Detecting Doubled Words in Text
- 3.15.7 Matching All-Caps Words
- 3.15.8 Matching Version Numbers
- 3.15.9 A Few Other Patterns

3.16 Conclusion

4 Internationalization in Ruby

- 4.1 Background and Terminology
- 4.2 Working with Character Encodings
 - 4.2.1 Normalization
 - 4.2.2 Encoding Conversions
 - 4.2.3 Transliteration
 - 4.2.4 Collation
- 4.3 Translations
 - 4.3.1 Defaults
 - 4.3.2 Namespaces
 - 4.3.3 Interpolation
 - 4.3.4 Pluralization
- 4.4 Localized Formatting
 - 4.4.1 Dates and Times
 - 4.4.2 Numbers
 - 4.4.3 Currencies
- 4.5 Conclusion

5 Performing Numerical Calculations

Table of Contents

- 5.1 Representing Numbers in Ruby
- 5.2 Basic Operations on Numbers
- 5.3 Rounding Floating Point Values
- 5.4 Comparing Floating Point Numbers
- 5.5 Formatting Numbers for Output
- 5.6 Formatting Numbers with Commas
- 5.7 Working with Very Large Integers
- 5.8 Using BigDecimal
- 5.9 Working with Rational Values
- 5.10 Matrix Manipulation
- 5.11 Working with Complex Numbers
- 5.12 Using mathn
- 5.13 Finding Prime Factorization, GCD, and LCM
- 5.14 Working with Prime Numbers
- 5.15 Implicit and Explicit Numeric Conversion
- 5.16 Coercing Numeric Values
- 5.17 Performing Bit-Level Operations on Numbers
- 5.18 Performing Base Conversions
- 5.19 Finding Cube Roots, Fourth Roots, and So On
- 5.20 Determining the Architectures Byte Order
- 5.21 Numerical Computation of a Definite Integral
- 5.22 Trigonometry in Degrees, Radians, and Grads
- 5.23 Finding Logarithms with Arbitrary Bases
- 5.24 Finding the Mean, Median, and Mode of a Data Set
- 5.25 Variance and Standard Deviation
- 5.26 Finding a Correlation Coefficient

Table of Contents

5.27 Generating Random Numbers

5.28 Caching Functions with Memoization

5.29 Conclusion

6 Symbols and Ranges

6.1 Symbols

6.1.1 Symbols as Enumerations

6.1.2 Symbols as Metavalues

6.1.3 Symbols, Variables, and Methods

6.1.4 Converting to/from Symbols

6.2 Ranges

6.2.1 Open and Closed Ranges

6.2.2 Finding Endpoints

6.2.3 Iterating Over Ranges

6.2.4 Testing Range Membership

6.2.5 Converting to Arrays

6.2.6 Backward Ranges

6.2.7 The Flip-Flop Operator

6.2.8 Custom Ranges

6.3 Conclusion

7 Working with Times and Dates

7.1 Determining the Current Time

7.2 Working with Specific Times (Post-Epoch)

7.3 Determining the Day of the Week

7.4 Determining the Date of Easter

7.5 Finding the Nth Weekday in a Month

7.6 Converting Between Seconds and Larger Units

Table of Contents

- 7.7 Converting to and from the Epoch
- 7.8 Working with Leap Seconds: Dont!
- 7.9 Finding the Day of the Year
- 7.10 Validating a Date or Time
- 7.11 Finding the Week of the Year
- 7.12 Detecting Leap Years
- 7.13 Obtaining the Time Zone
- 7.14 Working with Hours and Minutes Only
- 7.15 Comparing Time Values
- 7.16 Adding Intervals to Time Values
- 7.17 Computing the Difference in Two Time Values
- 7.18 Working with Specific Dates (Pre-Epoch)
- 7.19 Time, Date, and DateTime
- 7.20 Parsing a Date or Time String
- 7.21 Formatting and Printing Time Values
- 7.22 Time Zone Conversions
- 7.23 Determining the Number of Days in a Month
- 7.24 Dividing a Month into Weeks
- 7.25 Conclusion

8 Arrays, Hashes, and Other Enumerables

- 8.1 Working with Arrays
 - 8.1.1 Creating and Initializing an Array
 - 8.1.2 Accessing and Assigning Array Elements
 - 8.1.3 Finding an Arrays Size
 - 8.1.4 Comparing Arrays
 - 8.1.5 Sorting an Array

Table of Contents

- 8.1.6 Selecting from an Array by Criteria
- 8.1.7 Using Specialized Indexing Functions
- 8.1.8 Implementing a Sparse Matrix
- 8.1.9 Using Arrays as Mathematical Sets
- 8.1.10 Randomizing an Array
- 8.1.11 Using Multidimensional Arrays
- 8.1.12 Finding Elements in One Array But Not Another
- 8.1.13 Transforming or Mapping Arrays
- 8.1.14 Removing nil Values from an Array
- 8.1.15 Removing Specific Array Elements
- 8.1.16 Concatenating and Appending onto Arrays
- 8.1.17 Using an Array as a Stack or Queue
- 8.1.18 Iterating over an Array
- 8.1.19 Interposing Delimiters to Form a String
- 8.1.20 Reversing an Array
- 8.1.21 Removing Duplicate Elements from an Array
- 8.1.22 Interleaving Arrays
- 8.1.23 Counting Frequency of Values in an Array
- 8.1.24 Inverting an Array to Form a Hash
- 8.1.25 Synchronized Sorting of Multiple Arrays
- 8.1.26 Establishing a Default Value for New Array Elements

8.2 Working with Hashes

- 8.2.1 Creating a New Hash
- 8.2.2 Specifying a Default Value for a Hash
- 8.2.3 Accessing and Adding Key-Value Pairs
- 8.2.4 Deleting Key-Value Pairs
- 8.2.5 Iterating Over a Hash
- 8.2.6 Inverting a Hash

Table of Contents

- 8.2.7 Detecting Keys and Values in a Hash
- 8.2.8 Extracting Hashes into Arrays
- 8.2.9 Selecting Key-Value Pairs by Criteria
- 8.2.10 Sorting a Hash
- 8.2.11 Merging Two Hashes
- 8.2.12 Creating a Hash from an Array
- 8.2.13 Finding Difference or Intersection of Hash Keys
- 8.2.14 Using a Hash as a Sparse Matrix
- 8.2.15 Implementing a Hash with Duplicate Keys
- 8.2.16 Other Hash Operations

8.3 Enumerables in General

- 8.3.1 The inject Method
- 8.3.2 Using Quantifiers
- 8.3.3 The partition Method
- 8.3.4 Iterating by Groups
- 8.3.5 Converting to Arrays or Sets
- 8.3.6 Using Enumerator Objects

8.4 More on Enumerables

- 8.4.1 Searching and Selecting
- 8.4.2 Counting and Comparing
- 8.4.3 Iterating
- 8.4.4 Extracting and Converting
- 8.4.5 Lazy Enumerators

8.5 Conclusion

9 More Advanced Data Structures

9.1 Working with Sets

- 9.1.1 Simple Set Operations
- 9.1.2 More Advanced Set Operations

Table of Contents

9.2 Working with Stacks and Queues

- 9.2.1 Implementing a Stricter Stack
- 9.2.2 Detecting Unbalanced Punctuation in Expressions
- 9.2.3 Understanding Stacks and Recursion
- 9.2.4 Implementing a Stricter Queue

9.3 Working with Trees

- 9.3.1 Implementing a Binary Tree
- 9.3.2 Sorting Using a Binary Tree
- 9.3.3 Using a Binary Tree as a Lookup Table
- 9.3.4 Converting a Tree to a String or Array

9.4 Working with Graphs

- 9.4.1 Implementing a Graph as an Adjacency Matrix
- 9.4.2 Determining Whether a Graph Is Fully Connected
- 9.4.3 Determining Whether a Graph Has an Euler Circuit
- 9.4.4 Determining Whether a Graph Has an Euler Path
- 9.4.5 Graph Tools in Ruby

9.5 Conclusion

10 I/O and Data Storage

10.1 Working with Files and Directories

- 10.1.1 Opening and Closing Files
- 10.1.2 Updating a File
- 10.1.3 Appending to a File
- 10.1.4 Random Access to Files
- 10.1.5 Working with Binary Files
- 10.1.6 Locking Files
- 10.1.7 Performing Simple I/O
- 10.1.8 Performing Buffered and Unbuffered I/O
- 10.1.9 Manipulating File Ownership and Permissions

Table of Contents

- 10.1.10 Retrieving and Setting Timestamp Information
- 10.1.11 Checking File Existence and Size
- 10.1.12 Checking Special File Characteristics
- 10.1.13 Working with Pipes
- 10.1.14 Performing Special I/O Operations
- 10.1.15 Using Nonblocking I/O
- 10.1.16 Using readpartial
- 10.1.17 Manipulating Pathnames
- 10.1.18 Using the Pathname Class
- 10.1.19 Command-Level File Manipulation
- 10.1.20 Grabbing Characters from the Keyboard
- 10.1.21 Reading an Entire File into Memory
- 10.1.22 Iterating Over a File by Lines
- 10.1.23 Iterating Over a File by Byte or Character
- 10.1.24 Treating a String As a File
- 10.1.25 Copying a Stream
- 10.1.26 Working with Character Encodings
- 10.1.27 Reading Data Embedded in a Program
- 10.1.28 Reading Program Source
- 10.1.29 Working with Temporary Files
- 10.1.30 Changing and Setting the Current Directory
- 10.1.31 Changing the Current Root
- 10.1.32 Iterating Over Directory Entries
- 10.1.33 Getting a List of Directory Entries
- 10.1.34 Creating a Chain of Directories
- 10.1.35 Deleting a Directory Recursively
- 10.1.36 Finding Files and Directories

10.2 Higher-Level Data Access

Table of Contents

- 10.2.1 Simple Marshaling
- 10.2.2 Deep Copying with Marshal
- 10.2.3 More Complex Marshaling
- 10.2.4 Marshaling with YAML
- 10.2.5 Persisting Data with JSON
- 10.2.6 Working with CSV Data
- 10.2.7 SQLite3 for SQL Data Storage

10.3 Connecting to External Data Stores

- 10.3.1 Connecting to MySQL Databases
- 10.3.2 Connecting to PostgreSQL Databases
- 10.3.3 Object-Relational Mappers (ORMs)
- 10.3.4 Connecting to Redis Data Stores

10.4 Conclusion

11 OOP and Dynamic Features in Ruby

11.1 Everyday OOP Tasks

- 11.1.1 Using Multiple Constructors
- 11.1.2 Creating Instance Attributes
- 11.1.3 Using More Elaborate Constructors
- 11.1.4 Creating Class-Level Attributes and Methods
- 11.1.5 Inheriting from a Superclass
- 11.1.6 Testing Classes of Objects
- 11.1.7 Testing Equality of Objects
- 11.1.8 Controlling Access to Methods
- 11.1.9 Copying an Object
- 11.1.10 Using `initialize_copy`
- 11.1.11 Understanding `allocate`
- 11.1.12 Working with Modules
- 11.1.13 Transforming or Converting Objects

Table of Contents

11.1.14 Creating Data-Only Classes (Structs)

11.1.15 Freezing Objects

11.1.16 Using tap in Method Chaining

11.2 More Advanced Techniques

11.2.1 Sending an Explicit Message to an Object

11.2.2 Specializing an Individual Object

11.2.3 Nesting Classes and Modules

11.2.4 Creating Parametric Classes

11.2.5 Storing Code as Proc Objects

11.2.6 Storing Code as Method Objects

11.2.7 Using Symbols as Blocks

11.2.8 How Module Inclusion Works

11.2.9 Detecting Default Parameters

11.2.10 Delegating or Forwarding

11.2.11 Defining Class-Level Readers and Writers

11.2.12 Working in Advanced Programming Disciplines

11.3 Working with Dynamic Features

11.3.1 Evaluating Code Dynamically

11.3.2 Retrieving a Constant by Name

11.3.3 Retrieving a Class by Name

11.3.4 Using `define_method`

11.3.5 Obtaining Lists of Defined Entities

11.3.6 Removing Definitions

11.3.7 Handling References to Nonexistent Constants

11.3.8 Handling Calls to Nonexistent Methods

11.3.9 Improved Security with taint

11.3.10 Defining Finalizers for Objects

11.4 Program Introspection

Table of Contents

- 11.4.1 Traversing the Object Space
- 11.4.2 Examining the Call Stack
- 11.4.3 Tracking Changes to a Class or Object Definition
- 11.4.4 Monitoring Program Execution

11.5 Conclusion

12 Graphical Interfaces for Ruby

12.1 Shoes 4

- 12.1.1 Starting Out with Shoes
- 12.1.2 An Interactive Button
- 12.1.3 Text and Input
- 12.1.4 Layout
- 12.1.5 Images and Shapes
- 12.1.6 Events
- 12.1.7 Other Notes

12.2 Ruby/Tk

- 12.2.1 Overview
- 12.2.2 A Simple Windowed Application
- 12.2.3 Working with Buttons
- 12.2.4 Working with Text Fields
- 12.2.5 Working with Other Widgets
- 12.2.6 Other Notes

12.3 Ruby/GTK3

- 12.3.1 Overview
- 12.3.2 A Simple Windowed Application
- 12.3.3 Working with Buttons
- 12.3.4 Working with Text Fields
- 12.3.5 Working with Other Widgets
- 12.3.6 Other Notes

Table of Contents

12.4 QtRuby

- 12.4.1 Overview
- 12.4.2 A Simple Windowed Application
- 12.4.3 Working with Buttons
- 12.4.4 Working with Text Fields
- 12.4.5 Working with Other Widgets
- 12.4.6 Other Notes

12.5 Swing

12.6 Other GUI Toolkits

- 12.6.1 UNIX and X11
- 12.6.2 FXRuby (FOX)
- 12.6.3 RubyMotion for iOS and Mac OS X
- 12.6.4 The Windows Win32API

12.7 Conclusion

13 Threads and Concurrency

13.1 Creating and Manipulating Threads

- 13.1.1 Creating Threads
- 13.1.2 Accessing Thread-Local Variables
- 13.1.3 Querying and Changing Thread Status
- 13.1.4 Achieving a Rendezvous (and Capturing a Return Value)
- 13.1.5 Dealing with Exceptions
- 13.1.6 Using a Thread Group

13.2 Synchronizing Threads

- 13.2.1 Performing Simple Synchronization
- 13.2.2 Synchronizing Access with a Mutex
- 13.2.3 Using the Built-in Queue Classes
- 13.2.4 Using Condition Variables

Table of Contents

13.2.5 Other Synchronization Techniques 518

13.2.6 Setting a Timeout for an Operation

13.2.7 Waiting for an Event

13.2.8 Collection Searching in Parallel

13.2.9 Recursive Deletion in Parallel

13.3 Fibers and Cooperative Multitasking

13.4 Conclusion

14 Scripting and System Administration

14.1 Running External Programs

14.1.1 Using system and exec

14.1.2 Capturing Command Output

14.1.3 Manipulating Processes

14.1.4 Manipulating Standard Input and Output

14.2 Command-Line Options and Arguments

14.2.1 Working with ARGV

14.2.2 Working with ARGF

14.2.3 Parsing Command-Line Options

14.3 The Shell Library

14.3.1 Using Shell for I/O Redirection

14.3.2 Other Notes on Shell

14.4 Accessing Environment Variables

14.4.1 Getting and Setting Environment Variables

14.4.2 Storing Environment Variables as an Array or Hash

14.5 Working with Files, Directories, and Trees

14.5.1 A Few Words on Text Filters

14.5.2 Copying a Directory Tree

14.5.3 Deleting Files by Age or Other Criteria

Table of Contents

14.5.4 Determining Free Space on a Disk

14.6 Other Scripting Tasks

14.6.1 Distributing Ruby Programs

14.6.2 Piping into the Ruby Interpreter

14.6.3 Testing Whether a Program Is Running Interactively

14.6.4 Determining the Current Platform or Operating System

14.6.5 Using the Etc Module

14.7 Conclusion

15 Ruby and Data Formats

15.1 Parsing JSON

15.1.1 Navigating JSON Data

15.1.2 Handling Non-JSON Data Types

15.1.3 Other JSON Libraries

15.2 Parsing XML (and HTML)

15.2.1 Document Parsing

15.2.2 Stream Parsing

15.3 Working with RSS and Atom

15.3.1 Parsing Feeds

15.3.2 Generating Feeds

15.4 Manipulating Image Data with RMagick

15.4.1 Common Graphics Tasks

15.4.2 Special Effects and Transformations

15.4.3 The Drawing API

15.5 Creating PDF Documents with Prawn

15.5.1 Basic Concepts and Techniques

15.5.2 An Example Document

15.6 Conclusion

Table of Contents

16 Testing and Debugging

- 16.1 Testing with RSpec
- 16.2 Testing with Minitest
- 16.3 Testing with Cucumber
- 16.4 Using the byebug Debugger
- 16.5 Using pry for Debugging
- 16.6 Measuring Performance
- 16.7 Pretty-Printing Objects
- 16.8 Not Covered Here
- 16.9 Conclusion

17 Packaging and Distributing Code

- 17.1 Libraries and Rubygems
 - 17.1.1 Using Rubygems
 - 17.1.2 Creating Gems
- 17.2 Managing Dependencies with Bundler
 - 17.2.1 Semantic Versioning
 - 17.2.2 Dependencies from Git
 - 17.2.3 Creating Gems with Bundler
 - 17.2.4 Private Gems
- 17.3 Using RDoc
 - 17.3.1 Simple Markup
 - 17.3.2 Advanced Documentation with Yard
- 17.4 Conclusion

18 Network Programming

- 18.1 Network Servers
 - 18.1.1 A Simple Server: Time of Day

Table of Contents

18.1.2 Implementing a Threaded Server

18.1.3 Case Study: A Peer-to-Peer Chess Server

18.2 Network Clients

18.2.1 Retrieving Truly Random Numbers from the Web

18.2.2 Contacting an Official Timeserver

18.2.3 Interacting with a POP Server

18.2.4 Sending Mail with SMTP

18.2.5 Interacting with an IMAP Server

18.2.6 Encoding/Decoding Attachments

18.2.7 Case Study: A Mail-News Gateway

18.2.8 Retrieving a Web Page from a URL

18.2.9 Using the Open-URI Library

18.3 Conclusion

19 Ruby and Web Applications

19.1 HTTP Servers

19.1.1 A Simple HTTP Server

19.1.2 Rack and Web Servers

19.2 Application Frameworks

19.2.1 Routing in Sinatra

19.2.2 Routing in Rails

19.2.3 Parameters in Sinatra

19.2.4 Parameters in Rails

19.3 Storing Data

19.3.1 Databases

19.3.2 Data Stores

19.4 Generating HTML

19.4.1 ERB

Table of Contents

19.4.2 Haml

19.4.3 Other Templating Systems

19.5 The Asset Pipeline

19.5.1 CSS and Sass

19.5.2 JavaScript and CoffeeScript

19.6 Web Services via HTTP

19.6.1 JSON for APIs

19.6.2 REST (and REST-ish) APIs

19.7 Generating Static Sites

19.7.1 Middleman

19.7.2 Other Static Site Generators

19.8 Conclusion

20 Distributed Ruby

20.1 An Overview: Using drb

20.2 Case Study: A Stock Ticker Simulation

20.3 Rinda: A Ruby Tuplespace

20.4 Service Discovery with Distributed Ruby

20.5 Conclusion

21 Ruby Development Tools

21.1 Using Rake

21.2 Using irb

21.3 The Basics of pry

21.4 The ri Utility

21.5 Editor Support

21.5.1 Vim

21.5.2 Emacs

Table of Contents

21.6 Ruby Version Managers

21.6.1 Using rvm

21.6.2 Using rbenv

21.6.3 Using chruby

21.7 Conclusion

22 The Ruby Community

22.1 Web Resources

22.2 Mailing Lists, Podcasts, and Forums

22.3 Ruby Bug Reports and Feature Requests

22.4 IRC Channels

22.5 Ruby Conferences

22.6 Local Ruby Groups

22.7 Conclusion

Index