

Jeremy McAnally  
Assaf Arkin

with YEHUDA KATZ, DAVID BLACK, GREGORY BROWN,  
PETER COOPER, AND LUKE MELIA

# RUBY

IN PRACTICE



MANNING



*preface* xv  
*acknowledgments* xvii  
*about this book* xix

---

## PART 1 RUBY TECHNIQUES ..... 1

---

### **I** *Ruby under the microscope* 3

- 1.1 Why Ruby now? 4
  - Optimizing developer cycles* 4 ▪ *Language features* 6
- 1.2 Ruby by example 6
- 1.3 Facets of Ruby 10
  - Duck typing* 12 ▪ *Simplicity* 14 ▪ *DRY efficiency* 14 ▪ *Functional programming* 15
- 1.4 Metaprogramming 16
  - Getting started with metaprogramming* 17 ▪ *Domain-specific languages* 21 ▪ *Refining your metaprogramming* 22
- 1.5 Summary 24

## 2 *Testing Ruby* 25

- 2.1 Testing principles 26
  - Why bother with testing?* 26
  - Types of testing* 27
  - Testing workflow* 28
- 2.2 Test-driven development with Ruby 29
- 2.3 Behavior-driven development with RSpec 34
  - What is behavior-driven development?* 35
  - Testing with RSpec* 36
- 2.4 A testing environment 41
  - Setting up a baseline with fixture data* 41
  - Faking components with stubs* 44
  - Setting behavior expectations with mock objects* 46
- 2.5 Testing your tests 49
  - Testing code coverage* 49
  - Testing quality with Heckle* 51
- 2.6 Summary 53

## 3 *Scripting with Ruby* 54

- 3.1 Scripting with Ruby 55
- 3.2 Automating with OLE and OSA 59
  - Automating Outlook with Ruby* 59
  - Automating iCal with Ruby* 61
- 3.3 Using Rake 64
  - Using tasks* 65
  - File tasks* 69
- 3.4 Summary 73

---

## PART 2 INTEGRATION AND COMMUNICATION..... 75

## 4 *Ruby on Rails techniques* 77

- 4.1 Extending Rails 77
  - Using helpers to expose Ruby libraries* 78
  - Metaprogramming away duplication* 82
  - Turning your code into reusable components* 84
- 4.2 Rails performance 86
  - Benchmarking a Rails application* 86
  - Profiling a Rails application* 88
- 4.3 Summary 89

## 5 **Web services** 90

- 5.1 Using HTTP 91
  - HTTP GET 91* ▪ *HTTP POST 93* ▪ *Serving HTTP requests 97*
- 5.2 REST with Rails 101
  - RESTful resources 101* ▪ *Serving XML, JSON, and Atom 109*
  - Using ActiveResource 114*
- 5.3 SOAP services 117
  - Implementing the service 117* ▪ *Invoking the service 123*
- 5.4 Summary 125

## 6 **Automating communication** 126

- 6.1 Automating email 127
  - Automating sending email 127* ▪ *Receiving email 130* ▪ *Processing email 132*
- 6.2 Automating instant communication 134
  - Sending messages with AIM 134* ▪ *Automating Jabber 136*
- 6.3 Summary 139

## 7 **Asynchronous messaging** 140

- 7.1 Open source messaging servers 140
  - Using ActiveMQ 141* ▪ *Using reliable-msg 145*
- 7.2 WebSphere MQ 149
  - Queuing messages 149* ▪ *Processing messages 155*
- 7.3 Summary 159

## 8 **Deployment** 160

- 8.1 Creating deployable packages with RubyGems 161
  - Using RubyGems in your organization 161* ▪ *Setting up a RubyGems repository 164*
- 8.2 Deploying web applications 167
  - Simplifying deployment with Capistrano 167* ▪ *Tailing remote logs with Capistrano 170* ▪ *Deploying with Vlad the Deployer 171*
- 8.3 Monitoring with God.rb 173
  - A typical God setup 173* ▪ *Notifications 176*
- 8.4 Summary 178

## 9 Database facilities and techniques 183

- 9.1 Using plain-text files for data persistence 184
- 9.2 Using the (g)dbm API 189
- 9.3 The MySQL driver 197
- 9.4 Using DBI 204
- 9.5 Summary 207

## 10 Structured documents 208

- 10.1 XML in practice 209
  - Using XML to read configuration files 209* • *Writing configuration data to disk 212*
- 10.2 Parsing HTML and XHTML with Hpricot 214
  - Post-processing HTML output 215* • *Reading broken HTML 216*
- 10.3 Writing configuration data: revisited 220
- 10.4 Reading RSS feeds 221
- 10.5 Creating your own feed 225
- 10.6 Using YAML for data storage 230
- 10.7 Summary 231

## 11 Identity and authentication 233

- 11.1 Securely storing a password 234
  - Authenticating against LDAP 237*
- 11.2 Authenticating against Active Directory 238
- 11.3 Adding authentication to your Rails application 240
- 11.4 Semi-private, personalized feeds 243
- 11.5 HTTP Basic Authentication 246
- 11.6 Integrating OpenID into your application 247
- 11.7 Summary 252

## 12 Searching and indexing 253

- 12.1 The principles of searching 254
- 12.2 Standalone and high-performance searching 254
  - Standalone indexing and search with Ferret 255* • *Integrating with the Solr search engine 259* • *Ultrafast indexing and searching with FTSearch 263* • *Indexing and searching Rails data with Ferret and Solr 267* • *Searching in Rails with Utrasphinx 270*

- 12.3 Integrating search with other technologies 273
  - Web search using a basic API (Yahoo!) 273* ▪ *Web search using a scraping technique (Google) 275*
- 12.4 Summary 277

## **13 Document processing and reporting 278**

- 13.1 Processing CSV data 279
- 13.2 Generating and emailing daily reports 283
- 13.3 Comparing text reports to well-formed data 289
- 13.4 Creating customized documents for printing 295
- 13.5 Reporting against large datasets 301
- 13.6 Summary 306

- appendix A Installing Ruby 307*
- appendix B JRuby 312*
- appendix C Deploying web apps 316*
- index 325*