

Confident Ruby Ebook Avdi Grimm

Build Awesome Command-Line Applications in Ruby 2
Object-oriented Software Construction
Everyday Scripting with Ruby
Design Patterns in Ruby (Adobe Reader)
Ruby Under a Microscope
Practical Object-Oriented Design
Rails Recipes
Effective Ruby
Python in Practice
Moral Panics and the Copyright Wars
The RSpec Book
Object Design
Practical Object-oriented Design in Ruby
Agile Web Development with Rails
Take My Money: Accepting Payments on the Web
Crafting Rails 4 Applications
Refactoring Workbook
Leading Out Loud
Ruby on Rails Tutorial
Metaprogramming Ruby 2
Refactoring Databases
Learning the Vi Editor
Beginning Ruby
Refactoring The Pragmatic Programmer
Wicked Cool Ruby Scripts
Clean Ruby
From Bash to Z Shell
The Leprechauns of Software Engineering
The Rails 5 Way
The Art of Unit Testing
Refactoring SQL Applications
The Craft of Software Testing
Agile Web Development with Rails 5
The Well-Grounded Rubyist
Metaprogramming Ruby
Text Processing with Ruby
Eloquent Ruby
The Stoic Art of Living
Learning Ruby

Build Awesome Command-Line Applications in Ruby 2

This is the eBook of the printed book and may not include any media, website access codes, or print supplements that may come packaged with the bound book. Used by sites as varied as Twitter,

GitHub, Disney, and Airbnb, Ruby on Rails is one of the most popular frameworks for developing web applications, but it can be challenging to learn and use. Whether you're new to web development or new only to Rails, *Ruby on Rails™ Tutorial, Fourth Edition*, is the solution. Best-selling author and leading Rails developer Michael Hartl teaches Rails by guiding you through the development of three example applications of increasing sophistication. The tutorial's examples focus on the general principles of web development needed for virtually any kind of website. The updates to this edition include full compatibility with Rails 5, a division of the largest chapters into more manageable units, and a huge number of new exercises interspersed in each chapter for maximum reinforcement of the material. This indispensable guide provides integrated tutorials not only for Rails, but also for the essential Ruby, HTML, CSS, and SQL skills you need when developing web applications. Hartl explains how each new technique solves a real-world problem, and then he demonstrates it with bite-sized code that's simple enough to understand, yet novel enough to be useful. Whatever your previous web development experience, this book will guide you to true Rails mastery. This book will help you install and set up your Rails development environment, including pre-installed integrated development environment (IDE) in the cloud. Go beyond generated code to truly understand how to build Rails applications from scratch. Learn testing and test-driven development (TDD). Effectively use the Model-View-Controller (MVC) pattern. Structure applications using the REST architecture. Build static pages and transform them into dynamic ones. Master the Ruby

programming skills all Rails developers need Create high-quality site layouts and data models Implement registration and authentication systems, including validation and secure passwords Update, display, and delete users Upload images in production using a cloud storage service Implement account activation and password reset, including sending email with Rails Add social features and microblogging, including an introduction to Ajax Record version changes with Git and create a secure remote repository at Bitbucket Deploy your applications early and often with Heroku

Object-oriented Software Construction

Beginning Ruby is a thoroughly contemporary guide to this powerful object-oriented language. It's one of the only guides aimed at both the novice programmer as well as experienced developers who are new to Ruby. The book starts by explaining the principles behind object oriented programming and within a few chapters builds towards creating a genuine Ruby application. The book then explains key Ruby principles, such as classes and objects; projects, modules and libraries; and other aspects of Ruby such as database access. In addition, Ruby on Rails is covered in some depth and the book's appendixes provide essential and long-lasting reference information.

Everyday Scripting with Ruby

The Complete Guide to Writing More Maintainable,

Manageable, Pleasing, and Powerful Ruby Applications

Ruby's widely admired ease of use has a downside: Too many Ruby and Rails applications have been created without concern for their long-term maintenance or evolution. The Web is awash in Ruby code that is now virtually impossible to change or extend. This text helps you solve that problem by using powerful real-world object-oriented design techniques, which it thoroughly explains using simple and practical Ruby examples. Sandi Metz has distilled a lifetime of conversations and presentations about object-oriented design into a set of Ruby-focused practices for crafting manageable, extensible, and pleasing code. She shows you how to build new applications that can survive success and repair existing applications that have become impossible to change. Each technique is illustrated with extended examples, all downloadable from the companion Web site, poodr.info. The first title to focus squarely on object-oriented Ruby application design, *Practical Object-Oriented Design in Ruby* will guide you to superior outcomes, whatever your previous Ruby experience. Novice Ruby programmers will find specific rules to live by; intermediate Ruby programmers will find valuable principles they can flexibly interpret and apply; and advanced Ruby programmers will find a common language they can use to lead development and guide their colleagues. This guide will help you

- Understand how object-oriented programming can help you craft Ruby code that is easier to maintain and upgrade
- Decide what belongs in a single Ruby class
- Avoid entangling objects that should be kept separate
- Define flexible interfaces among objects
- Reduce programming

overhead costs with duck typing Successfully apply inheritance Build objects via composition Design cost-effective tests Solve common problems associated with poorly designed Ruby code

Design Patterns in Ruby (Adobe Reader)

Summary The Well-Grounded Rubyist, Third Edition is a beautifully written tutorial that begins with your first Ruby program and takes you all the way to sophisticated topics like reflection, threading, and recursion. Ruby masters David A. Black and Joe Leo distill their years of knowledge for you, concentrating on the language and its uses so you can use Ruby in any way you choose. Updated for Ruby 2.5. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology Designed for developer productivity, Ruby is an easy-to-learn dynamic language perfect for creating virtually any kind of software. Its famously friendly development community, countless libraries, and amazing tools, like the Rails framework, have established it as the language of choice for high-profile companies, including GitHub, SlideShare, and Shopify. The future is bright for the well-grounded Rubyist! About the Book In The Well-Grounded Rubyist, Third Edition, expert authors David A. Black and Joseph Leo deliver Ruby mastery in an easy-to-read, casual style. You'll lock in core principles as you write your first Ruby programs. Then, you'll progressively build up to topics like reflection, threading, and recursion, cementing your knowledge with high-value exercises to practice your skills along

the way. What's Inside Basic Ruby syntax Running Ruby extensions FP concepts like currying, side-effect-free code, and recursion Ruby 2.5 updates About the Reader For readers with beginner-level programming skills. About the Authors David A. Black is an internationally known Ruby developer and author, and a cofounder of Ruby Central. Ruby teacher and advocate Joseph Leo III is the founder of Def Method and lead organizer of the Gotham Ruby Conference.

Table of Contents

PART 1 RUBY FOUNDATIONS

Bootstrapping your Ruby literacy Objects, methods, and local variables Organizing objects with classes Modules and program organization The default object (self), scope, and visibility Control-flow techniques

PART 2 BUILT-IN CLASSES AND MODULES Built-in essentials Strings, symbols, and other scalar objects Collection and container objects Collections central: Enumerable and Enumerator Regular expressions and regexp-based string operations File and I/O operations

PART 3 RUBY DYNAMICS Object individuation Callable and runnable objects Callbacks, hooks, and runtime introspection Ruby and functional programming

Ruby Under a Microscope

If you're an experienced Ruby programmer, *Effective Ruby* will help you harness Ruby's full power to write more robust, efficient, maintainable, and well-performing code. Drawing on nearly a decade of Ruby experience, Peter J. Jones brings together 48 Ruby best practices, expert tips, and shortcuts—all supported by realistic code examples. Jones offers practical advice for each major area of Ruby

development, from modules to memory to metaprogramming. Throughout, he uncovers little-known idioms, quirks, pitfalls, and intricacies that powerfully impact code behavior and performance. Each item contains specific, actionable, clearly organized guidelines; careful advice; detailed technical arguments; and illuminating code examples. When multiple options exist, Jones shows you how to choose the one that will work best in your situation. Effective Ruby will help you systematically improve your code—not by blindly following rules, but by thoroughly understanding Ruby programming techniques. Key features of this concise guide include

- How to avoid pitfalls associated with Ruby's sometimes surprising idiosyncrasies
- What you should know about inheritance hierarchies to successfully use Rails (and other large frameworks)
- How to use misunderstood methods to do amazingly useful things with collections
- Better ways to use exceptions to improve code reliability
- Powerful metaprogramming approaches (and techniques to avoid)
- Practical, efficient testing solutions, including MiniTest Unit and Spec Testing
- How to reliably manage RubyGem dependencies
- How to make the most of Ruby's memory management and profiling tools
- How to improve code efficiency by understanding the Ruby interpreter's internals

Practical Object-Oriented Design

Annotation Everyone in the Ruby world seems to be talking about metaprogramming--how you can use it to remove duplication in your code and write elegant,

beautiful programs. Now you can get in on the action as well. This book describes metaprogramming as an essential component of Ruby. Once you understand the principles of Ruby, including the object model, scopes, and eigenclasses, you're on your way to applying metaprogramming both in your daily work and in your fun, after-hours projects. Learning metaprogramming doesn't have to be difficult or boring. By taking you on a Monday-through-Friday workweek adventure with a pair of programmers, Paolo Perrotta helps make mastering the art of metaprogramming both straightforward and entertaining. The book is packed with: Pragmatic examples of metaprogramming in action, many of which come straight from popular libraries or frameworks, such as Rails. Programming challenges that let you experiment and play with some of the most fun, "out-there" metaprogramming concepts. Metaprogramming "spells"--34 practical recipes and idioms that you can study and apply right now, to write code that is sure to impress. Whether you're a Ruby apprentice on the path to mastering the language or a Ruby wiz in search of new tips, this book is for you.

Rails Recipes

2nd edition of the step-by-step guide that helps developers to write test sets that are maintainable, readable and trustworthy.

Effective Ruby

In *Moral Panics and the Copyright Wars*, William Patry offers a lively, unflinching examination of the pitched battles over new technology, business models, and most of all, consumers. He lays bare how we got to where we are: a bloated, punitive legal regime that has strayed far from its modest, but important roots. A centrist and believer in appropriately balanced copyright laws, Patry concludes that the only laws we need are effective laws, laws that further the purpose of encouraging the creation of new works and learning.

Python in Practice

Provides information on metaprogramming concepts to help write productive Ruby code.

Moral Panics and the Copyright Wars

It's easy to write correct Ruby code, but to gain the fluency needed to write great Ruby code, you must go beyond syntax and absorb the "Ruby way" of thinking and problem solving. In *Eloquent Ruby*, Russ Olsen helps you write Ruby like true Rubyists do—so you can leverage its immense, surprising power. Olsen draws on years of experience internalizing the Ruby culture and teaching Ruby to other programmers. He guides you to the "Ah Ha!" moments when it suddenly becomes clear why Ruby works the way it does, and how you can take advantage of this language's elegance and expressiveness. *Eloquent Ruby* starts small, answering tactical questions focused on a single statement, method, test, or bug. You'll learn

how to write code that actually looks like Ruby (not Java or C#); why Ruby has so many control structures; how to use strings, expressions, and symbols; and what dynamic typing is really good for. Next, the book addresses bigger questions related to building methods and classes. You'll discover why Ruby classes contain so many tiny methods, when to use operator overloading, and when to avoid it. Olsen explains how to write Ruby code that writes its own code—and why you'll want to. He concludes with powerful project-level features and techniques ranging from gems to Domain Specific Languages. A part of the renowned Addison-Wesley Professional Ruby Series, *Eloquent Ruby* will help you “put on your Ruby-colored glasses” and get results that make you a true believer.

The RSpec Book

Rails 5 and Ruby 2.2 bring many improvements, including new APIs and substantial performance enhancements, and the fifth edition of this award-winning classic is now updated! If you're new to Rails, you'll get step-by-step guidance. If you're an experienced developer, this book will give you the comprehensive, insider information you need for the latest version of Ruby on Rails. *Ruby on Rails* helps you produce high-quality, beautiful-looking web applications quickly. You concentrate on creating the application, and Rails takes care of the details. Learn Rails the way the Rails core team recommends it, along with the tens of thousands of developers who have used this broad, far-reaching tutorial and

reference. We start with a step-by-step walkthrough of building a real application, and in-depth chapters look at the built-in Rails features. Follow along with an extended tutorial as you write a web-based store application. Eliminate tedious configuration and housekeeping; internationalize your applications; incorporate Ajax, REST, web services, and e-mail handling into your applications; test your applications as you write them using the built-in testing frameworks; and deploy your applications easily and securely. New in this edition is coverage of Action Cable, and completely updated code for Rails 5. Rails 1.0 was released in December 2005, more than 10 years ago. This book was there from the start, and didn't just evolve alongside Rails, it evolved with Rails. It has been developed in consultation with the Rails core team. In fact, Rails itself is tested against the code in this book. What You Need: All you need is a Windows, Mac OS X, or Linux machine to do development on. This book will take you through the steps to install Rails and its dependencies. If you aren't familiar with the Ruby programming language, this book contains a chapter that covers the basics necessary to understand the material in the book.

Object Design

Learn how to make better decisions and write cleaner Ruby code. This book shows you how to avoid messy code that is hard to test and which cripples productivity. Author Carleton DiLeo shares hard-learned lessons gained from years of experience across numerous codebases both large and small.

Each chapter covers the topics you need to know to make better decisions and optimize your productivity. Many books will tell you how to do something; this book will tell you why you should do it. Start writing code you love. What You Will Learn Build better classes to help promote code reuse Improve your decision making and make better, smarter choices Identify bad code and fixed it Create quality names for all of your variables, classes, and modules Write better, concise classes Improve the quality of your methods Properly use modules Clarify your Boolean logic See when and how you refactor Improve your understanding of TDD and write better tests Who This Book Is For This book is written for Ruby developers. There is no need to learn a new language or translate concepts to Ruby.

Practical Object-oriented Design in Ruby

The Definitive Refactoring Guide, Fully Revamped for Ruby With refactoring, programmers can transform even the most chaotic software into well-designed systems that are far easier to evolve and maintain. What's more, they can do it one step at a time, through a series of simple, proven steps. Now, there's an authoritative and extensively updated version of Martin Fowler's classic refactoring book that utilizes Ruby examples and idioms throughout-not code adapted from Java or any other environment. The authors introduce a detailed catalog of more than 70 proven Ruby refactorings, with specific guidance on when to apply each of them, step-by-step instructions for using them, and example code illustrating how

they work. Many of the authors' refactorings use powerful Ruby-specific features, and all code samples are available for download. Leveraging Fowler's original concepts, the authors show how to perform refactoring in a controlled, efficient, incremental manner, so you methodically improve your code's structure without introducing new bugs. Whatever your role in writing or maintaining Ruby code, this book will be an indispensable resource. This book will help you Understand the core principles of refactoring and the reasons for doing it Recognize "bad smells" in your Ruby code Rework bad designs into well-designed code, one step at a time Build tests to make sure your refactorings work properly Understand the challenges of refactoring and how they can be overcome Compose methods to package code properly Move features between objects to place responsibilities where they fit best Organize data to make it easier to work with Simplify conditional expressions and make more effective use of polymorphism Create interfaces that are easier to understand and use Generalize more effectively Perform larger refactorings that transform entire software systems and may take months or years Successfully refactor Ruby on Rails code

Agile Web Development with Rails

The "Bible" for Rails Development: Fully Updated for Rails 5 "When I read The Rails Way for the first time, I felt like I truly understood Rails for the first time."
—Steve Klabnik, Rails contributor and mentor The Rails™ 5 Way is the comprehensive, authoritative

reference guide for professionals delivering production-quality code using modern Ruby on Rails. Obie Fernandez illuminates the entire Rails 5 API, its most powerful idioms, design approaches, and libraries. He presents new and updated content on Action Cable, RSpec 3.4, Turbolinks 5.0, the Attributes API, and many other enhancements, both major and subtle. Through detailed code examples, you'll dive deep into Ruby on Rails, discover why it's designed as it is, and learn to make it do exactly what you want. Proven in thousands of production systems, the knowledge in this book will maximize your productivity and help you build more successful solutions. Build powerful, scalable, REST-compliant back-end services Program complex program flows using Action Controller Represent models, relationships, and operations in Active Record, and apply advanced Active Record techniques Smoothly evolve database schema via Migrations Craft front-ends with ActionView and the Asset Pipeline Optimize performance and scalability with caching and Turbolinks 5.0 Improve your productivity using Haml HTML templating Secure your systems against attacks like SQL Injection, XSS, and XSRF Integrate email using Action Mailer Enable real-time, websockets-based browser behavior with Action Cable Improve responsiveness with background processing Build "API-only" back-end projects that speak JSON Leverage enhancements to Active Job, serialization, and Ajax support

Take My Money: Accepting Payments on the Web

In this updated edition of the Jolt Award-winning book, users are shown a new approach to Web development using Rails 2, making this the most up-to-date and authoritative Rails book out there.

Crafting Rails 4 Applications

The newest edition of the bestselling guide to authentic leadership communication Much has changed in the world since the original publication of Leading Out Loud, Terry Pearce's book on authentic leadership communication. Now, more than ever, the development of a leader's message is as crucial to success as the delivery of that message. In the third edition of his classic book, Terry Pearce shows leaders in all sectors how to communicate their values and vision to inspire commitment. In this important resource, Pearce continues to broaden the application of core principles, putting the spotlight on every day, spontaneous communication. New examples, covering the range of today's multi-faceted communication, show the application of the sage advice Pearce offers. Readers will see how to develop a Personal Leadership Communication Guide that supports any venue, through any media and in multiple cultures. This completely revised and updated version of the bestselling classic is designed to meet the communication needs of today's leaders. Pearce expands his exploration of the internal work necessary to create an honest and compelling vision. He emphasizes the deepening of emotional awareness necessary to inspire others This edition demonstrates how readers can find their authentic

voices and articulate their messages with increasing confidence and empathy Some examples carry through across chapters, clarifying how one develops and strengthens the Personal Leadership Communication Guide over time The work presents new models that are applicable to the multi-cultural world in which we live. Readers, leaders of any organization, and teachers at any level will find practical illustrations of how differences can be bridged with universal principles Foreword by Randy Komisar, General Partner of Kleiner Perkins Caufield & Byers and author of The Monk and The Riddle This new edition offers information, stories and experiences that demonstrate success in authentic leadership communication, in any technology, whatever the field or venue, local or global.

Refactoring Workbook

Winner of the 2014 Jolt Award for "Best Book"
"Whether you are an experienced programmer or are starting your career, Python in Practice is full of valuable advice and example to help you improve your craft by thinking about problems from different perspectives, introducing tools, and detailing techniques to create more effective solutions." --Doug Hellmann, Senior Developer, DreamHost If you're an experienced Python programmer, Python in Practice will help you improve the quality, reliability, speed, maintainability, and usability of all your Python programs. Mark Summerfield focuses on four key themes: design patterns for coding elegance, faster processing through concurrency and compiled Python

(Cython), high-level networking, and graphics. He identifies well-proven design patterns that are useful in Python, illuminates them with expert-quality code, and explains why some object-oriented design patterns are irrelevant to Python. He also explodes several counterproductive myths about Python programming--showing, for example, how Python can take full advantage of multicore hardware. All examples, including three complete case studies, have been tested with Python 3.3 (and, where possible, Python 3.2 and 3.1) and crafted to maintain compatibility with future Python 3.x versions. All code has been tested on Linux, and most code has also been tested on OS X and Windows. All code may be downloaded at www.qtrac.eu/pipbook.html. Coverage includes Leveraging Python's most effective

- creational, structural, and behavioral design patterns
- Supporting concurrency with Python's multiprocessing, threading, and concurrent.futures modules
- Avoiding concurrency problems using thread-safe queues and futures rather than fragile locks
- Simplifying networking with high-level modules, including xmlrpclib and RPyC
- Accelerating Python code with Cython, C-based Python modules, profiling, and other techniques
- Creating modern-looking GUI applications with Tkinter
- Leveraging today's powerful graphics hardware via the OpenGL API using pyglet and PyOpenGL

Leading Out Loud

You don't have to know everything about a car to drive one, and you don't need to know everything

about Ruby to start programming with it. Written for both experienced and new programmers alike, Learning Ruby is a just-get-in-and-drive book -- a hands-on tutorial that offers lots of Ruby programs and lets you know how and why they work, just enough to get you rolling down the road. Interest in Ruby stems from the popularity of Rails, the web development framework that's attracting new devotees and refugees from Java and PHP. But there are plenty of other uses for this versatile language. The best way to learn is to just try the code! You'll find examples on nearly every page of this book that you can imitate and hack. Briefly, this book: Outlines many of the most important features of Ruby Demonstrates how to use conditionals, and how to manipulate strings in Ruby. Includes a section on regular expressions Describes how to use operators, basic math, functions from the Math module, rational numbers, etc. Talks you through Ruby arrays, and demonstrates hashes in detail Explains how to process files with Ruby Discusses Ruby classes and modules (mixins) in detail, including a brief introduction to object-oriented programming (OOP) Introduces processing XML, the Tk toolkit, RubyGems, reflection, RDoc, embedded Ruby, metaprogramming, exception handling, and other topics Acquaints you with some of the essentials of Rails, and includes a short Rails tutorial. Each chapter concludes with a set of review questions, and appendices provide you with a glossary of terms related to Ruby programming, plus reference material from the book in one convenient location. If you want to take Ruby out for a drive, Learning Ruby holds the keys.

Ruby on Rails Tutorial

Text is everywhere. Web pages, databases, the contents of files--for almost any programming task you perform, you need to process text. Cut even the most complex text-based tasks down to size and learn how to master regular expressions, scrape information from Web pages, develop reusable utilities to process text in pipelines, and more. Most information in the world is in text format, and programmers often find themselves needing to make sense of the data hiding within. It might be to convert it from one format to another, or to find out information about the text as a whole, or to extract information from it. But how do you do this efficiently, avoiding labor-intensive, manual work? Text Processing with Ruby takes a practical approach. You'll learn how to get text into your Ruby programs from the file system and from user input. You'll process delimited files such as CSVs, and write utilities that interact with other programs in text-processing pipelines. Decipher character encoding mysteries, and avoid the pain of jumbled characters and malformed output. You'll learn to use regular expressions to match, extract, and replace patterns in text. You'll write a parser and learn how to process Web pages to pull out information from even the messiest of HTML. Before long you'll be able to tackle even the most enormous and entangled text with ease, scything through gigabytes of data and effortlessly extracting the bits that matter. What You Need: This book requires a passing familiarity with the Ruby programming language, and assumes that you

already have Ruby installed on your computer.

Metaprogramming Ruby 2

Behaviour Driven Development is about writing software that matters. It is an approach to agile software development that takes cues from Test Driven Development, Domain Driven Design, and Acceptance Test Driven Planning. RSpec and Cucumber are the leading Behaviour Driven Development tools in Ruby. RSpec supports Test Driven Development in Ruby through the BDD lens, keeping your focus on design and documentation while also supporting thorough testing and quick fault isolation. Cucumber, RSpec's steadfast companion, supports Acceptance Test Driven Planning with business-facing, executable requirements documentation that helps to ensure that you are writing relevant software targeted at real business needs. The RSpec Book will introduce you to RSpec, Cucumber, and a number of other tools that make up the Ruby BDD family. Replete with tutorials and practical examples, the RSpec Book will help you get your BDD on, taking you from executable requirements to working software that is clean, well tested, well documented, flexible and highly maintainable.

Refactoring Databases

Getting paid using Stripe or PayPal is only the beginning of creating a fully functional e-commerce application. You also need to handle failure cases,

inventory management, administration, security, reporting, and be compliant with legal issues. Manage one-time transactions and recurring subscriptions, handle inventory management, issue discounts and refunds, mitigate administration and compliance issues, and test your code to ensure your customers have a smooth, hassle-free experience. An e-commerce payment application is literally rewarding to build--you can see the return on investment as genuine money is added to your account. But it can be stressful to manage, with security and compliance concerns and administration issues. And your entire business may depend on these features working smoothly. Let Noel Rappin guide you through the setup and complications of dealing with online financial transactions. Go beyond just the interaction with the gateway service and build an application that will be robust and useful over time. Set up a Stripe and PayPal payment gateway and accept credit card payments. Use the Stripe API to improve security by validating credit card data without sending it through your own server. Design your application for maximum flexibility against the inevitable complexities of business logic, including handling discounts. Manage the multiple failure points of dealing with payment gateways and test for failure cases. Use background jobs to simplify third party interactions. Handle administrative tasks such as issuing refunds and discounts while maintaining data integrity and security. Create subscription plans and manage recurring payments, and stay on top of legal issues regarding taxes, reporting, and compliance. Pay affiliates or contributors from your application. By the end, you will know how to create a fully-functional

web payment-taking machine. What You Need: The code in this book works with Ruby 2.3.1 and Rails 5, though nearly all of the code will run with earlier versions of Ruby and Rails.

Learning the Vi Editor

The Complete Guide to Writing Maintainable, Manageable, Pleasing, and Powerful Object-Oriented Applications Object-oriented programming languages exist to help you create beautiful, straightforward applications that are easy to change and simple to extend. Unfortunately, the world is awash with object-oriented (OO) applications that are difficult to understand and expensive to change. Practical Object-Oriented Design, Second Edition, immerses you in an OO mindset and teaches you powerful, real-world, object-oriented design techniques with simple and practical examples. Sandi Metz demonstrates how to build new applications that can “survive success” and repair existing applications that have become impossible to change. Each technique is illustrated with extended examples in the easy-to-understand Ruby programming language, all downloadable from the companion website, poodr.com. Fully updated for Ruby 2.5, this guide shows how to Decide what belongs in a single class Avoid entangling objects that should be kept separate Define flexible interfaces among objects Reduce programming overhead costs with duck typing Successfully apply inheritance Build objects via composition Whatever your previous object-oriented experience, this concise guide will help you achieve the superior outcomes you’re

looking for. Register your book for convenient access to downloads, updates, and/or corrections as they become available. See inside book for details.

Beginning Ruby

Refactoring has proven its value in a wide range of development projects—helping software professionals improve system designs, maintainability, extensibility, and performance. Now, for the first time, leading agile methodologist Scott Ambler and renowned consultant Pramodkumar Sadalage introduce powerful refactoring techniques specifically designed for database systems. Ambler and Sadalage demonstrate how small changes to table structures, data, stored procedures, and triggers can significantly enhance virtually any database design—without changing semantics. You’ll learn how to evolve database schemas in step with source code—and become far more effective in projects relying on iterative, agile methodologies. This comprehensive guide and reference helps you overcome the practical obstacles to refactoring real-world databases by covering every fundamental concept underlying database refactoring. Using start-to-finish examples, the authors walk you through refactoring simple standalone database applications as well as sophisticated multi-application scenarios. You’ll master every task involved in refactoring database schemas, and discover best practices for deploying refactorings in even the most complex production environments. The second half of this book systematically covers five major categories of

database refactorings. You'll learn how to use refactoring to enhance database structure, data quality, and referential integrity; and how to refactor both architectures and methods. This book provides an extensive set of examples built with Oracle and Java and easily adaptable for other languages, such as C#, C++, or VB.NET, and other databases, such as DB2, SQL Server, MySQL, and Sybase. Using this book's techniques and examples, you can reduce waste, rework, risk, and cost—and build database systems capable of evolving smoothly, far into the future.

Refactoring

For many users, working in the Unix environment means using vi, a full-screen text editor available on most Unix systems. Even those who know vi often make use of only a small number of its features. Learning the vi Editor is a complete guide to text editing with vi. Topics new to the sixth edition include multiscreen editing and coverage of four clones: vim, elvis, nvi, and vile and their enhancements to vi, such as multi-window editing, GUI interfaces, extended regular expressions, and enhancements for programmers. A new appendix describes vi's place in the Unix and Internet cultures. Quickly learn the basics of editing, cursor movement, and global search and replacement. Then take advantage of the more subtle power of vi. Extend your editing skills by learning to use ex, a powerful line editor, from within vi. For easy reference, the sixth edition also includes a command

summary at the end of each appropriate chapter. Topics covered include: Basic editing Moving around in a hurry Beyond the basics Greater power with ex Global search and replacement Customizing vi and ex Command shortcuts Introduction to the vi clones' extensions The nvi, elvis, vim, and vile editors Quick reference to vi and ex commands vi and the Internet

The Pragmatic Programmer

Get ready to see Rails as you've never seen it before. Learn how to extend the framework, change its behavior, and replace whole components to bend it to your will. Eight different test-driven tutorials will help you understand Rails' inner workings and prepare you to tackle complicated projects with solutions that are well-tested, modular, and easy to maintain. This second edition of the bestselling *Crafting Rails Applications* has been updated to Rails 4 and discusses new topics such as streaming, mountable engines, and thread safety. Rails is one of the most extensible frameworks out there. This pioneering book deep-dives into the Rails plugin APIs and shows you, the intermediate Rails developer, how to use them to write better web applications and make your day-to-day work with Rails more productive. Rails Core developer Jose Valim guides you through eight different tutorials, each using test-driven development to build a new Rails plugin or application that solves common problems with these APIs. You'll learn how the Rails rendering stack works and customize it to read templates from the database

while you discover how to mimic Active Record behavior, such as validations, in any other object. You'll find out how Rails integrates with Rack, the different ways to stream data from your web application, and how to mix Rails engines and Sinatra applications into your Rails apps, so you can choose the most appropriate tool for the job. In addition, you'll improve your productivity by customizing generators and responders. This book will help you understand Rails' inner workings, including generators, template handlers, internationalization, routing, and responders. With the knowledge you'll gain, you'll create well-tested, modular, and robust solutions for your next project.

Wicked Cool Ruby Scripts

& Most software practitioners deal with inherited code; this book teaches them how to optimize it & & Workbook approach facilitates the learning process & & Helps you identify where problems in a software application exist or are likely to exist

Clean Ruby

Speak directly to your system. With its simple commands, flags, and parameters, a well-formed command-line application is the quickest way to automate a backup, a build, or a deployment and simplify your life. With this book, you'll learn specific ways to write command-line applications that are easy to use, deploy, and maintain, using a set of clear best practices and the Ruby programming language.

This book is designed to make any programmer or system administrator more productive in their job. Now updated for Ruby 2. Writing a command-line application that's self-documenting, robust, adaptable and forever useful is easier than you might think. Ruby is particularly suited to this task, because it combines high-level abstractions with "close to the metal" system interaction wrapped up in a concise, readable syntax. Plus, Ruby has the support of a rich ecosystem of open source tools and libraries. Ten insightful chapters each explain and demonstrate a command-line best practice. You'll see how to use these tools to elevate the lowliest automation script to a maintainable, polished application. You'll learn how to use free, open source parsers to create user-friendly command-line interfaces as well as command suites. You'll see how to use defaults to keep options simple for everyday users, while giving advanced users options for more complex tasks. There's no reason why a command-line application should lack documentation, whether it's part of a help command or a man page; you'll find out when and how to use both. Your journey from command-line novice to pro ends with a look at valuable approaches to testing your apps, and includes some fun techniques for outside-the-box, colorful interfaces that will delight your users. With Ruby, the command line is not dead. Long live the command line.

From Bash to Z Shell

This reference presents, in detail, an effective, step-by-step, cost-effective approach to software testing

that is based on common practice—with improvements inspired by academic testing research and practical experience. The approach is designed to be gradually adoptable, so that it does not disrupt current work, and it scales down gracefully under schedule pressure. Outlines a systematic process/strategy of software testing that incorporates test design, test implementation, and measurements of test quality. Explains solid testing techniques in detail and shows how to apply them to testing tasks. Answers such questions as: How do I design tests? What are common tester errors, and how do I avoid them? How do I implement tests? How do I know how good my tests are? How do I know when I've tested enough? Features unique coverage of testing bug fixes and other changes. For software developers testing their own code or designs; independent testers testing someone else's code; and testers or developers testing bug fixes and other maintenance changes.

The Leprechauns of Software Engineering

What can you do when database performance doesn't meet expectations? Before you turn to expensive hardware upgrades to solve the problem, reach for this book. Refactoring SQL Applications provides a set of tested options for making code modifications to dramatically improve the way your database applications function. Backed by real-world examples, you'll find quick fixes for simple problems, in-depth answers for more complex situations, and complete

solutions for applications with extensive problems. Learn to: Determine if and where you can expect performance gains Apply quick fixes, such as limiting calls to the database in stored functions and procedures Refactor tasks, such as replacing application code by a stored procedure, or replacing iterative, procedural statements with sweeping SQL statements Refactor flow by increasing parallelism and switching business-induced processing from synchronous to asynchronous Refactor design using schema extensions, regular views, materialized views, partitioning, and more Compare before and after versions of a program to ensure you get the same results once you make modifications Refactoring SQL Applications teaches you to recognize and assess code that needs refactoring, and to understand the crucial link between refactoring and performance. If and when your application bogs down, this book will help you get it back up to speed.

The Rails 5 Way

Praise for Design Patterns in Ruby " Design Patterns in Ruby documents smart ways to resolve many problems that Ruby developers commonly encounter. Russ Olsen has done a great job of selecting classic patterns and augmenting these with newer patterns that have special relevance for Ruby. He clearly explains each idea, making a wealth of experience available to Ruby developers for their own daily work." —Steve Metsker, Managing Consultant with Dominion Digital, Inc. "This book provides a great demonstration of the key 'Gang of Four' design

patterns without resorting to overly technical explanations. Written in a precise, yet almost informal style, this book covers enough ground that even those without prior exposure to design patterns will soon feel confident applying them using Ruby. Olsen has done a great job to make a book about a classically 'dry' subject into such an engaging and even occasionally humorous read." —Peter Cooper

"This book renewed my interest in understanding patterns after a decade of good intentions. Russ picked the most useful patterns for Ruby and introduced them in a straightforward and logical manner, going beyond the GoF's patterns. This book has improved my use of Ruby, and encouraged me to blow off the dust covering the GoF book." —Mike Stok

" Design Patterns in Ruby is a great way for programmers from statically typed objectoriented languages to learn how design patterns appear in a more dynamic, flexible language like Ruby." —Rob Sanheim, Ruby Ninja, Relevance

Most design pattern books are based on C++ and Java. But Ruby is different—and the language's unique qualities make design patterns easier to implement and use. In this book, Russ Olsen demonstrates how to combine Ruby's power and elegance with patterns, and write more sophisticated, effective software with far fewer lines of code. After reviewing the history, concepts, and goals of design patterns, Olsen offers a quick tour of the Ruby language—enough to allow any experienced software developer to immediately utilize patterns with Ruby. The book especially calls attention to Ruby features that simplify the use of patterns, including dynamic typing, code closures, and "mixins" for easier code reuse. Fourteen of the

classic "Gang of Four" patterns are considered from the Ruby point of view, explaining what problems each pattern solves, discussing whether traditional implementations make sense in the Ruby environment, and introducing Ruby-specific improvements. You'll discover opportunities to implement patterns in just one or two lines of code, instead of the endlessly repeated boilerplate that conventional languages often require. Design Patterns in Ruby also identifies innovative new patterns that have emerged from the Ruby community. These include ways to create custom objects with metaprogramming, as well as the ambitious Rails-based "Convention Over Configuration" pattern, designed to help integrate entire applications and frameworks. Engaging, practical, and accessible, Design Patterns in Ruby will help you build better software while making your Ruby programming experience more rewarding.

The Art of Unit Testing

Object technology pioneer Wirfs-Brock teams with expert McKean to present a thoroughly updated, modern, and proven method for the design of software. The book is packed with practical design techniques that enable the practitioner to get the job done.

Refactoring SQL Applications

What others in the trenches say about The Pragmatic Programmer "The cool thing about this book is that

it's great for keeping the programming process fresh. The book helps you to continue to grow and clearly comes from people who have been there." —Kent Beck, author of *Extreme Programming Explained: Embrace Change* "I found this book to be a great mix of solid advice and wonderful analogies!" —Martin Fowler, author of *Refactoring* and *UML Distilled* "I would buy a copy, read it twice, then tell all my colleagues to run out and grab a copy. This is a book I would never loan because I would worry about it being lost." —Kevin Ruland, Management Science, MSG-Logistics "The wisdom and practical experience of the authors is obvious. The topics presented are relevant and useful. By far its greatest strength for me has been the outstanding analogies—tracer bullets, broken windows, and the fabulous helicopter-based explanation of the need for orthogonality, especially in a crisis situation. I have little doubt that this book will eventually become an excellent source of useful information for journeymen programmers and expert mentors alike." —John Lakos, author of *Large-Scale C++ Software Design* "This is the sort of book I will buy a dozen copies of when it comes out so I can give it to my clients." —Eric Vought, Software Engineer "Most modern books on software development fail to cover the basics of what makes a great software developer, instead spending their time on syntax or technology where in reality the greatest leverage possible for any software team is in having talented developers who really know their craft well. An excellent book." —Pete McBreen, Independent Consultant "Since reading this book, I have implemented many of the practical suggestions and tips it contains. Across the board, they have saved my

company time and money while helping me get my job done quicker! This should be a desktop reference for everyone who works with code for a living.”

—Jared Richardson, Senior Software Developer, iRenaissance, Inc. “I would like to see this issued to every new employee at my company.” —Chris

Cleeland, Senior Software Engineer, Object Computing, Inc. “If I’m putting together a project, it’s the authors of this book that I want. . . . And failing that I’d settle for people who’ve read their book.”

—Ward Cunningham Straight from the programming trenches, *The Pragmatic Programmer* cuts through the increasing specialization and technicalities of modern software development to examine the core process—taking a requirement and producing working, maintainable code that delights its users. It covers topics ranging from personal responsibility and career development to architectural techniques for keeping your code flexible and easy to adapt and reuse. Read this book, and you'll learn how to Fight software rot; Avoid the trap of duplicating knowledge; Write flexible, dynamic, and adaptable code; Avoid programming by coincidence; Bullet-proof your code with contracts, assertions, and exceptions; Capture real requirements; Test ruthlessly and effectively; Delight your users; Build teams of pragmatic programmers; and Make your developments more precise with automation. Written as a series of self-contained sections and filled with entertaining anecdotes, thoughtful examples, and interesting analogies, *The Pragmatic Programmer* illustrates the best practices and major pitfalls of many different aspects of software development. Whether you're a new coder, an experienced programmer, or a

manager responsible for software projects, use these lessons daily, and you'll quickly see improvements in personal productivity, accuracy, and job satisfaction. You'll learn skills and develop habits and attitudes that form the foundation for long-term success in your career. You'll become a Pragmatic Programmer.

The Craft of Software Testing

* In-depth, unique coverage of ZSH, one of most modern and powerful of all shells. Also covers Bash, the preferred shell for most serious Linux and Unix users. * Very strong author and tech review team: Co-author Peter Stephenson has been involved in the development of Zsh since the 1990s when he started to write the FAQ. For the last few years, he has served as coordinator of the shell's development. Tech Reviewers: Ed Schaefer is the "Shell Corner" columnist for SysAdmin Magazine and Bart Schaefer is one of the lead developers of Zsh development. * Book is immediately useful, packed with short example and suggestions that the reader can put to use in their shell environment. * Extensive coverage of interactive and advanced shell features, including shell extensions, completion functions, and shortcuts. * Great book for users of all expertise; perennial seller.

Agile Web Development with Rails 5

Software -- Software Engineering.

The Well-Grounded Rubyist

Provides information on creating Web-based applications using Ruby.

Metaprogramming Ruby

Tom Morris is the former Notre Dame philosophy professor whose classes became a campus legend and whose nationwide speaking engagements have electrified the audiences of corporate America. Continuing in his mission to bring philosophical wisdom into the trenches of everyday life, he shows how ideas of Stoic Philosophy ? which emphasizes goals like gaining command of one's passions and achieving indifference to pain and distress ? are completely up to date in their relevance to the practical issues people confront in the 21st century. Divided into three accessible sections, the book focuses on three leading Stoics: the slave Epictetus, the lawyer Seneca, and the Roman emperor Marcus Aurelius. From the bottom of society to the upwardly mobile middle to society's very top, the book highlights how these Stoics' insights relate to modern experience. Philosophy buffs and fans of Morris's other works will appreciate this latest application of ancient wisdom to new concerns.

Text Processing with Ruby

The software profession has a problem, widely recognized but which nobody seems willing to do anything about; a variant of the well known ""telephone game,"" where some trivial rumor is repeated from one person to the next until it has

become distorted beyond recognition and blown up out of all proportion. Unfortunately, the objects of this telephone game are generally considered cornerstone truths of the discipline, to the point that their acceptance now seems to hinder further progress. This book takes a look at some of those "ground truths" the claimed 10x variation in productivity between developers; the "software crisis"; the cost-of-change curve; the "cone of uncertainty"; and more. It assesses the real weight of the evidence behind these ideas - and confronts the scary prospect of moving the state of the art forward in a discipline that has had the ground kicked from under it.

Eloquent Ruby

Contains fifty-eight Ruby scripts to solve a variety of problems for system administration, image manipulation, and management of a Website.

The Stoic Art of Living

"An under-the-hood look at how the Ruby programming language runs code. Extensively illustrated with complete explanations and hands-on experiments. Covers Ruby 2.x"--

Learning Ruby

Provides information on the basics of the Ruby scripting language and how to create scripts using test-driven design.

[ROMANCE](#) [ACTION & ADVENTURE](#) [MYSTERY & THRILLER](#) [BIOGRAPHIES & HISTORY](#) [CHILDREN'S](#) [YOUNG ADULT](#) [FANTASY](#) [HISTORICAL FICTION](#) [HORROR](#) [LITERARY FICTION](#) [NON-FICTION](#) [SCIENCE FICTION](#)