
Ruby Under A Microscope An Illustrated Guide To Ruby Internals

Book of F#
Head First Ruby
Mineralogy and Geochemistry of Ruby
Second Harmonic Generation Imaging
Ruby and Sapphire
Ruby Wizardry
Good Luck with That
What Species of Creatures
If Hemingway Wrote JavaScript
Programming Crystal
Beyond the Ruby Veil
The Rust Programming Language (Covers Rust 2018)
A Memorial Discourse
Design Patterns in Ruby
Ruby by Example
Test Driven Development in Ruby
Eloquent Ruby
The Big Book of Small Python Projects
Super Scratch Programming Adventure! (Scratch 3)
Practical Object-oriented Design in Ruby
Ruby Under a Microscope
Racket Programming the Fun Way
Ruby Performance Optimization
Clean Ruby
Learning Ruby
The Heat Treatment of Ruby & Sapphire
The Recursive Book of Recursion
Build Awesome Command-line Applications in Ruby 2
Photoatlas of Inclusions in Gemstones
Programming in CoffeeScript
Russian Alexandrites
The 12-Minute Athlete
Why?
The Well-Grounded Rubyist
Crafting Rails 4 Applications
Perl One-Liners
Practical Object-Oriented Design
The Art of the Snowflake
Into the Midnight Void
Ruby Pocket Reference

*Ruby Under A
Microscope An
Illustrated Guide To
Ruby Internals*

Downloaded from
db.mwpai.edu by guest

SEMAJ HERRERA

Book of F# No Starch Press

Learn the basics of test driven development (TDD) using Ruby. You will carry out problem domain analysis, solution domain analysis, designing test cases, and writing tests first. These fundamental concepts will give you a solid TDD foundation to build upon. Test Driven Development in Ruby is written by a developer for developers. The concepts are first explained, then a coding demo illustrates how to apply the theory in practice. At the end of each chapter an exercise is given to reinforce the material. Complete with working files and code samples, you'll be able to work alongside the author, a trainer, by following the material in this book. What You Will Learn Carry out problem domain analysis, solution domain analysis, designing test cases, and writing tests first Use assertions Discover the structure of a test and the TDD cycle Gain an understanding of minimal implementation, starter test, story test, and next test Handle refactoring using Ruby Hide implementation details Test precisely and concretely Make your code robust Who This Book Is For Experienced Ruby programmers or web developers with some prior experience with Ruby.

Head First Ruby Pearson Education
Unlock your athletic potential and get into the best shape of your life with Krista Stryker's HIIT and bodyweight workouts—all of which can be done in just minutes a day! If you've ever thought you couldn't get results without spending hours in the gym, that you'd never be able to do a pull-up, or that it's too late to get in your best shape ever,

The 12-Minute Athlete will change your mind, your body, and your life. Get serious results with high-intensity interval training (HIIT) workouts that can be done in just minutes a day. Give up the excuses and learn to use your own bodyweight and a few basic pieces of portable equipment for short, incredibly effective workouts. Reset your mindset, bust through mental blocks, and set meaningful goals you'll actually accomplish. You can finally ditch the dieting and enjoy food as fuel with simple eating guidelines to the 80/20 rule. In The 12-Minute Athlete you'll also find: -A guide to basic calisthenics and bodyweight exercises for any fitness level -Progressive exercises to achieve seemingly "impossible" feats like pistol squats, one-arm push-ups, pull-ups, and handstands -More than a dozen simple and healthy recipes that will fuel your workouts -Two 8-week workout plans for getting fitter, faster, and stronger -Bonus Tabata workouts -And so much more! The 12-Minute Athlete is for men and women, ex-athletes and new athletes, experienced athletes and "non-athletes"—for anyone who has a body and wants to get stronger and start living their healthiest life.

Mineralogy and Geochemistry of Ruby
MDPI

There may be no better way to learn how to program than by dissecting real, representative examples written in your language of choice. Ruby by Example analyzes a series of Ruby scripts, examining how the code works, explaining the concepts it illustrates, and showing how to modify it to suit your needs. Baird's examples demonstrate key features of the language (such as inheritance, encapsulation, higher-order functions, and recursion), while simultaneously solving difficult problems

(such as validating XML, creating a bilingual program, and creating command-line interfaces). Each chapter builds upon the previous, and each key concept is highlighted in the margin to make it easier for you to navigate the book. You'll learn how to: -Use the interactive Ruby shell (irb) to learn key features of the language -Extend Ruby using RubyGems, the Ruby package manager -Create numerical utilities, as well as utilities that process and analyze HTML/XML -Implement purely functional and metaprogramming techniques to save time and effort -Optimize, profile, and test your code to make sure that it not only does its job, but does it well -Create web applications using Rails Ruby is the fastest growing programming language today, and for good reason: its elegant syntax and readable code make for prolific and happy programmers. But it can be difficult to understand and implement without a little help. Ruby by Example shows you how to take advantage of Ruby as you explore Ruby's fundamental concepts in action.

Second Harmonic Generation

Imaging Pragmatic Bookshelf

F# brings the power of functional-first programming to the .NET Framework, a platform for developing software in the Microsoft Windows ecosystem. If you're a traditional .NET developer used to C# and Visual Basic, discovering F# will be a revelation that will change how you code, and how you think about coding. In *The Book of F#*, Microsoft MVP Dave Fancher shares his expertise and teaches you how to wield the power of F# to write succinct, reliable, and predictable code. As you learn to take advantage of features like default immutability, pipelining, type inference, and pattern matching, you'll be amazed

at how efficient and elegant your code can be. You'll also learn how to: * Exploit F#'s functional nature using currying, partial application, and delegation * Streamline type creation and safety with record types and discriminated unions * Use collection types and modules to handle data sets more effectively * Use pattern matching to decompose complex types and branch your code within a single expression * Make your software more responsive with parallel programming and asynchronous workflows * Harness object orientation to develop rich frameworks and interact with code written in other .NET languages * Use query expressions and type providers to access and manipulate data sets from disparate sources Break free of that old school of programming. *The Book of F#* will show you how to unleash the expressiveness of F# to create smarter, leaner code.

Ruby and Sapphire Simon and Schuster From the dawn of time, ruby and sapphire have both attracted and fascinated humans in ways that few other items could. While objects of desire are found throughout the natural world, physical beauty is too often ephemeral. From the allure of a man, woman, flower or butterfly, through the fleeting moments of a sunset, there is little that lasts and practically nothing that can be passed down to our descendants. The exception is precious stones. Not only are they the most durable creations of mother nature, but their visual splendor is truly eternal. In this companion to his 2013 book, *Ruby & Sapphire--A Collector's Guide*, Richard Hughes examines these gems from the gemological standpoint, delving into these gems not just from the aesthetic, but also from the scientific point of view. The product of nearly 40 years of

firsthand experience, it covers every aspect of the subject from A-Z. History, sources, prices, quality analysis, synthetics and treatments, everything is here. *Ruby & Sapphire--A Gemologist's Guide* represents the most comprehensive book ever written on a single precious stone. With over 1000 photos, maps and illustrations and 3500 references, it is nothing less than a tour-de-force of gemological scholarship.

Ruby Wizardry Beyond the Ruby Veil, 1 A dark, queer YA fantasy that's perfect for fans of the Three Dark Crowns series and Wicked Saints. After Emanuela Ragno kills the one person in Occhia who can create water, she must find a way to save her city from dying of thirst. Emanuela Ragno always gets what she wants. With her daring mind and socialite schemes, she refuses to be the demure young lady everyone wants her to be. In her most ambitious move yet, she's about to marry Alessandro Morandi, her childhood best friend and the heir to the wealthiest house in Occhia. Emanuela doesn't care that she and her groom are both gay, because she doesn't want a love match. She wants power, and through Ale, she'll have it all. But Emanuela has a secret that could shatter her plans. In the city of Occhia, the only source of water is the watercrea, a mysterious being who uses magic to make water from blood. When their first bruise-like omen appears on their skin, all Occhians must surrender themselves to the watercrea to be drained of life. Everyone throughout history has given themselves up for the greater good. Everyone except Emanuela. She's kept the tiny omen on her hip out of sight for years. When the watercrea exposes Emanuela during her wedding ceremony and takes her to be sacrificed, Emanuela fights back . . . and

kills her. Now Occhia has no one to make their water and no idea how to get more. In a race against time, Emanuela and Ale must travel through the mysterious, blood-red veil that surrounds their city to uncover the secrets of the watercrea's magic and find a way to save their people -- no matter what it takes.

Good Luck with That "O'Reilly Media, Inc."

You don't have to accept slow Ruby or Rails performance. In this comprehensive guide to Ruby optimization, you'll learn how to write faster Ruby code--but that's just the beginning. See exactly what makes Ruby and Rails code slow, and how to fix it. Alex Dymo will guide you through perils of memory and CPU optimization, profiling, measuring, performance testing, garbage collection, and tuning. You'll find that all those "hard" things aren't so difficult after all, and your code will run orders of magnitude faster. This is the first book ever that consolidates all the Ruby performance optimization advice in one place. It's your comprehensive guide to memory optimization, CPU optimization, garbage collector tuning, profiling, measurements, performance testing, and more. You'll go from performance rookie to expert. First, you'll learn the best practices for writing Ruby code that's easy not only on the CPU, but also on memory, and that doesn't trigger the dreaded garbage collector. You'll find out that garbage collection accounts for 80% of slowdowns, and often takes more than 50% of your program's execution time. And you'll discover the bottlenecks in Rails code and learn how selective attribute loading and preloading can mitigate the performance costs of ActiveRecord. As you advance to Ruby performance expert, you'll learn how

profile your code, how to make sense out of profiler reports, and how to make optimization decisions based on them. You'll make sure slow code doesn't creep back into your Ruby application by writing performance tests, and you'll learn the right way to benchmark Ruby. And finally, you'll dive into the Ruby interpreter internals to really understand why garbage collection makes Ruby so slow, and how you can tune it up. What You Need: Some version of Ruby. The advice from this book applies to all modern Ruby versions from 1.9 to 2.2. 80% of the material will also be useful for legacy Ruby 1.8 users, and there is 1.8-specific advice as well.

What Species of Creatures Simon and Schuster

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.

If Hemingway Wrote JavaScript No Starch Press

Astrophysicist and author Mario Livio investigates perhaps the most human of all our characteristics—curiosity—in this “lively, expert, and definitely not dumbed-down account” (*Kirkus Reviews*) as he explores our innate desire to know why. Experiments demonstrate that people are more distracted when they overhear a phone conversation—where they can know only one side of the dialogue—than when they overhear two people talking and know both sides. Why does half a conversation make us more curious than a whole conversation? “Have you ever wondered why we wonder why? Mario Livio has, and he takes you on a fascinating quest to understand the origin and mechanisms of our curiosity. I thoroughly recommend it.” (*Adam Riess, Nobel Prize Winner in Physics, 2011*). Curiosity is not only at the heart of mystery and suspense novels, it is also essential to other creative endeavors, from painting to sculpture to music. It is the principal

driver of basic scientific research. Even so, there is still no definitive scientific consensus about why we humans are so curious, or about the mechanisms in our brain that are responsible for curiosity. In the ever-fascinating *Why?* Livio interviewed scientists in several fields to explore the nature of curiosity. He examined the lives of two of history's most curious geniuses, Leonardo da Vinci and Richard Feynman. He also talked to people with boundless curiosity: a superstar rock guitarist who is also an astrophysicist; an astronaut with degrees in computer science, biology, literature, and medicine. What drives these people to be curious about so many subjects? An astrophysicist who has written about mathematics, biology, and now psychology and neuroscience, Livio has firsthand knowledge of his subject which he explores in a lucid, entertaining way that will captivate anyone who is curious about curiosity.

[Programming Crystal](#) No Starch Press
 What if William Shakespeare were asked to generate the Fibonacci series or Jane Austen had to write a factorial program? In *If Hemingway Wrote JavaScript*, author Angus Croll imagines short JavaScript programs as written by famous wordsmiths. The result is a peculiar and charming combination of prose, poetry, and programming. The best authors are those who obsess about language—and the same goes for JavaScript developers. To master either craft, you must experiment with language to develop your own style, your own idioms, and your own expressions. To that end, *If Hemingway Wrote JavaScript* playfully bridges the worlds of programming and literature for the literary geek in all of us. Featuring original artwork by Miran Lipovača.

[Beyond the Ruby Veil](#) No Starch Press

Fans of Holly Black and Kendare Blake will obsess over the conclusion to this deliciously dark YA fantasy duology! Emanuela has finally gotten what she's always wanted. Since escaping her catacomb prison, she's become the supreme ruler of everything under the veils. Finally, she has the power to throw aside senseless, old traditions and run things exactly the way they should be. But when cracks in her magic start to show, Emanuela begrudgingly allies herself with her enemies, including her frustratingly alluring archnemesis, Verene. Together, they discover deeper truths about the mysterious blood magic Emanuela and Verene both wield. There is a higher, otherworldly authority outside the veils, and in order to save Occhia and the other realms, Emanuela may just have to rip another crown off someone's head.

The Rust Programming Language (Covers Rust 2018) Hachette UK
 Comics! Games! Programming! Now updated to cover Scratch 3. Scratch is the wildly popular educational programming language used by millions of first-time learners in classrooms and homes worldwide. By dragging together colorful blocks of code, kids can learn computer programming concepts and make cool games and animations. The latest version, Scratch 3, features an updated interface, new sprites and programming blocks, and extensions that let you program things like the micro:bit. In *Super Scratch Programming Adventure!*, kids learn programming fundamentals as they make their very own playable video games. They'll create projects inspired by classic arcade games that can be programmed (and played!) in an afternoon. Patient, step-by-step explanations of the code and fun programming challenges will

have kids creating their own games in no time. This full-color comic book makes programming concepts like variables, flow control, and subroutines effortless to absorb. Packed with ideas for games that kids will be proud to show off, *Super Scratch Programming Adventure!* is the perfect first step for the budding programmer. Covers Scratch 3

[A Memorial Discourse](#) Apress
Presents a series of detailed photographs of snowflakes from different locations around the world, demonstrating how complex crystalline patterns form and emerge.

[Design Patterns in Ruby](#) Tiller Press
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.

[Ruby by Example](#) No Starch Press
"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"--

[Test Driven Development in Ruby](#) Voyageur Press (MN)
Best-selling author Al Sweigart shows you how to easily build over 80 fun programs with minimal code and maximum creativity. If you’ve mastered basic Python syntax and you’re ready to start writing programs, you’ll find *The Big Book of Small Python Projects* both enlightening and fun. This collection of 81 Python projects will have you making digital art, games, animations, counting programs, and more right away. Once you see how the code works, you’ll practice re-creating the programs and experiment by adding your own custom touches. These simple, text-based programs are 256 lines of code or less. And whether it’s a vintage screensaver, a snail-racing game, a clickbait headline generator, or animated strands of DNA, each project is designed to be self-contained so you can easily share it online. You’ll create:

- Hangman, Blackjack, and other games to play against your friends or the computer
- Simulations of a forest fire, a million dice rolls, and a Japanese abacus
- Animations like a virtual fish tank, a

rotating cube, and a bouncing DVD logo screensaver • A first-person 3D maze game • Encryption programs that use ciphers like ROT13 and Vigenère to conceal text If you're tired of standard step-by-step tutorials, you'll love the learn-by-doing approach of *The Big Book of Small Python Projects*. It's proof that good things come in small programs!

Eloquent Ruby "O'Reilly Media, Inc." An accessible yet rigorous crash course on recursive programming using Python and JavaScript examples. Recursion has an intimidating reputation: it's considered to be an advanced computer science topic frequently brought up in coding interviews. But there's nothing magical about recursion. *The Recursive Book of Recursion* uses Python and JavaScript examples to teach the basics of recursion, exposing the ways that it's often poorly taught and clarifying the fundamental principles of all recursive algorithms. You'll learn when to use recursive functions (and, most importantly, when not to use them), how to implement the classic recursive algorithms often brought up in job interviews, and how recursive techniques can help solve countless problems involving tree traversal, combinatorics, and other tricky topics. This project-based guide contains complete, runnable programs to help you learn: How recursive functions make use of the call stack, a critical data structure almost never discussed in lessons on recursion How the head-tail and "leap of faith" techniques can simplify writing recursive functions How to use recursion to write custom search scripts for your filesystem, draw fractal art, create mazes, and more How optimization and memoization make recursive algorithms more efficient Al Sweigart has built a career explaining

programming concepts in a fun, approachable manner. If you've shied away from learning recursion but want to add this technique to your programming toolkit, or if you're racing to prepare for your next job interview, this book is for you.

The Big Book of Small Python Projects
Pragmatic Bookshelf

An introduction to the Racket functional programming language and DrRacket development environment to explore topics in mathematics (mostly recreational) and computer science. At last, a lively guided tour through all the features, functions, and applications of the Racket programming language.

You'll learn a variety of coding paradigms, including iterative, object oriented, and logic programming; create interactive graphics, draw diagrams, and solve puzzles as you explore Racket through fun computer science topics-- from statistical analysis to search algorithms, the Turing machine, and more. Early chapters cover basic Racket concepts like data types, syntax, variables, strings, and formatted output. You'll learn how to perform math in Racket's rich numerical environment, and use programming constructs in different problem domains (like coding solutions to the Tower of Hanoi puzzle). Later, you'll play with plotting, grapple with graphics, and visualize data. Then, you'll escape the confines of the command line to produce animations, interactive games, and a card trick program that'll dazzle your friends. You'll learn how to: Use DrRacket, an interactive development environment (IDE) for writing programs Compute classical math problems, like the Fibonacci sequence Generate two-dimensional function plots and create drawings using graphics primitives

Import and export data to and from Racket using ports, then visually analyze it Build simple computing devices (pushdown automaton, Turing machine, and so on) that perform tasks Leverage Racket's built-in libraries to develop a command line algebraic calculator Racket Programming the Fun Way is just like the language itself--an embodiment of everything that makes programming interesting and worthwhile, and that makes you a better programmer.

Super Scratch Programming Adventure! (Scratch 3) Penguin

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.

Practical Object-oriented Design in Ruby No Starch Press

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 fix 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.

Best Sellers - Books :

- [American Prometheus: The Triumph And Tragedy Of J. Robert Oppenheimer](#)
- [I'm Glad My Mom Died By Jennette McCurdy](#)

- [Oh, The Places You'll Go! By Dr. Seuss](#)
- [A Court Of Thorns And Roses Paperback Box Set \(5 Books\)](#)
- [Baking Yesteryear: The Best Recipes From The 1900s To The 1980s By B. Dylan Hollis](#)
- [The Collector: A Novel By Daniel Silva](#)
- [Fast Like A Girl: A Woman's Guide To Using The Healing Power Of Fasting To Burn Fat, Boost Energy, And Balance Hormones](#)
- [Rich Dad Poor Dad: What The Rich Teach Their Kids About Money That The Poor And Middle Class Do Not! By Robert T. Kiyosaki](#)
- [If He Had Been With Me](#)
- [Playground](#)