

---

# Programming Phoenix

---

Exploring Clojure, Elixir, Haskell, Scala, and Swift  
 Phoenix Web Development  
 Reclaim Your Creative Life in a Too-Busy World  
 The Phoenix Project  
 The Developer's Code  
 From Journeyman to Master  
 Control Your Computer, Simplify Your Life  
 Infinite Jest  
 Real-Time Phoenix  
 Build Awesome Command-Line Applications in Ruby 2  
 Programming Phoenix  
 Programming Elixir  $\geq$  1.6  
 Write Less Code, Get More Done (and Have Fun!)  
 New Foundations for a New World  
 The Transformative Practice of Taking Nothing for Granted  
 Find Your Unicorn Space  
 Learning Perl  
 Visualize Your Sensor Data with Phoenix and Grafana  
 Functional Programming: A PragPub Anthology  
 Designing Elixir Systems with Otp: Write Highly Scalable, Self-Healing Software with Layers  
 Learn Functional Programming with Elixir  
 Native, Cross-Platform Apps the Easy Way  
 Programming Phoenix 1.4  
 Stuff You Should Know  
 Functional |> Concurrent |> Pragmatic |> Fun  
 Phoenix in Action  
 Build Database Apps in Elixir for Scalability and Performance  
 Interactive Elixir Web Programming Without Writing Any JavaScript  
 Maximizing the Impact of Comics in Your Library: Graphic Novels, Manga, and More  
 Programming Ecto  
 Accelerate  
 From Concept to Production  
 Build Highly Scalable Systems with Channels  
 An Introduction to Computer Science Using Python 3.6  
 Modern C++ Programming with Test-Driven Development  
 Productive |> Reliable |> Fast  
 Metaprogramming Elixir  
 A Novel  
 The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations

*Programming Phoenix*

Downloaded from [ftp.wtvq.com](http://ftp.wtvq.com) by guest

---

## ZOE ELLEN

---

**Exploring Clojure, Elixir, Haskell, Scala, and Swift** ABC-CLIO  
 Summary Revised and updated for Elixir 1.7, *Elixir in Action*, Second Edition teaches you how to apply Elixir to practical problems associated with scalability, fault tolerance, and high availability. Along the way, you'll develop an appreciation for, and considerable skill in, a functional and concurrent style of programming. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology When you're building mission-critical software, fault tolerance matters. The Elixir programming language delivers fast, reliable applications, whether you're building a large-scale distributed system, a set of backend services, or a simple web app. And Elixir's elegant syntax and functional programming mindset make your software easy to write, read, and maintain. About the Book *Elixir in Action*, Second Edition teaches you how to build production-quality distributed applications using the Elixir programming language. Author Saša Jurić introduces this powerful language using examples that highlight the benefits of Elixir's functional and concurrent programming. You'll discover

how the OTP framework can radically reduce tedious low-level coding tasks. You'll also explore practical approaches to concurrency as you learn to distribute a production system over multiple machines. What's inside Updated for Elixir 1.7 Functional and concurrent programming Introduction to distributed system design Creating deployable releases About the Reader You'll need intermediate skills with client/server applications and a language like Java, C#, or Ruby. No previous experience with Elixir required. About the Author Saša Jurić is a developer with extensive experience using Elixir and Erlang in complex server-side systems. Table of Contents First steps Building blocks Control flow Data abstractions Concurrency primitives Generic server processes Building a concurrent system Fault-tolerance basics Isolating error effects Beyond GenServer Working with components Building a distributed system Running the system **Phoenix Web Development** Storey Publishing  
 Explore functional programming and discover new ways of thinking about code. You know you need to master functional programming, but learning one functional language is only the start. In this book, through articles drawn from PragPub magazine and articles written specifically for this book, you'll explore functional thinking and functional style and idioms across

languages. Led by expert guides, you'll discover the distinct strengths and approaches of Clojure, Elixir, Haskell, Scala, and Swift and learn which best suits your needs. Contributing authors: Rich Hickey, Stuart Halloway, Aaron Bedra, Michael Bevilacqua-Linn, Venkat Subramaniam, Paul Callaghan, Jose Valim, Dave Thomas, Natasha Murashev, Tony Hillerson, Josh Chisholm, and Bruce Tate. Functional programming is on the rise because it lets you write simpler, cleaner code, and its emphasis on immutability makes it ideal for maximizing the benefits of multiple cores and distributed solutions. So far nobody's invented the perfect functional language - each has its unique strengths. In *Functional Programming: A PragPub Anthology*, you'll investigate the philosophies, tools, and idioms of five different functional programming languages. See how Swift, the development language for iOS, encourages you to build highly scalable apps using functional techniques like map and reduce. Discover how Scala allows you to transition gently but deeply into functional programming without losing the benefits of the JVM, while with Lisp-based Clojure, you can plunge fully into the functional style. Learn about advanced functional concepts in Haskell, a pure functional language making powerful use of the type system with type inference and type classes. And see how functional programming is becoming more elegant and friendly with Elixir, a new functional language built on the powerful Erlang base. The industry has been embracing functional programming more and more, driven by the need for concurrency and parallelism. This collection of articles will lead you to mastering the functional approach to problem solving. So put on your explorer's hat and prepare to be surprised. The goal of exploration is always discovery. What You Need: Familiarity with one or more programming languages.

#### Reclaim Your Creative Life in a Too-Busy World Pragmatic Bookshelf

Don't accept the compromise between fast and beautiful: you can have it all. Phoenix creator Chris McCord, Elixir creator Jose Valim, and award-winning author Bruce Tate walk you through building an application that's fast and reliable. At every step, you'll learn from the Phoenix creators not just what to do, but why. Packed with insider insights and completely updated for Phoenix 1.4, this definitive guide will be your constant companion in your journey from Phoenix novice to expert, as you build the next generation of web applications. Phoenix is the long-awaited web framework based on Elixir, the highly concurrent language that combines a beautiful syntax with rich metaprogramming. The best way to learn Phoenix is to code, and you'll get to attack some interesting problems. Start working with controllers, views, and templates within the first few pages. Build an in-memory context, and then back it with an Ecto database layer, complete with changesets and constraints that keep readers informed and your database integrity intact. Craft your own interactive application based on the channels API for the real-time applications that this ecosystem made famous. Write your own authentication plugs, and use the OTP layer for supervised services. Organize code with modular umbrella projects. This edition is fully updated for Phoenix 1.4, with a new chapter on using Channel Presence to find out who's connected, even on a distributed application. Use the new generators and the new ExUnit features to organize tests and make Ecto tests concurrent. This is a book by developers and for developers, and we know how to help you ramp up quickly. Any book can tell you what to do. When you've finished this one, you'll also know why to do it. What You Need: To work through this book, you will need a computer capable of running Erlang 18 or higher, Elixir 1.5 or higher, and Phoenix 1.4 or higher. A rudimentary knowledge of Elixir is also highly recommended.

#### The Phoenix Project Pragmatic Bookshelf

Did you hear about PLC Programming, RSLogix 500, or Ladder Logic Programming? This is one of the five-book series of PLC programming using RSLogix 500, It will open your mind about ladder logic programming. It will open your mind to ladder logic programming. If you are starting to write the first program, this book is for you. It covers the basics of programming with ladder logic, and how ladder logic correlates to the PLC inputs and outputs, these principles are then put to work inside RSLogix 500, by explaining the basic commands that are required to control a machine. Besides, It references practical scenarios where the various aspects and discussed are applied in the real world.

#### The Developer's Code Manning Publications

This unique guide offers fresh insights on how graphic novels and comics differ from traditional books and require different treatment in the library—from purchasing, shelving, and cataloging to readers' advisory services, programs, and curriculum. Challenging librarians to rethink some of their traditional practices, *Maximizing the Impact of Comics in Your Library* provides creative and proven solutions for libraries of all types that want to get comics into the hands of fans and promote readership. The author describes how libraries would benefit from an in-house classification system and organization that accounts for both publishers and series. In addition, acquiring comics can often be tricky due to renumbering of series, reboots, shifting creative teams, and more—this book shows you how to work around those obstacles. Shelving and displays that reflect comic readers' browsing habits, creative programs that boost circulation of comics and graphic novels, and how comics can play a vital role in educational institutions are also covered. • Addresses common challenges librarians face with comics and graphic novels collections, and shows how to surmount them • Offers a solutions-focused approach • Describes how comics can be used to better engage your community and to educate youth • Fills a gap in the professional literature, covering topics not touched upon in the existing literature • Serves as a vital resource for public, academic, and school libraries

#### From Journeyman to Master Pragmatic Bookshelf

Shows how to write, debug, and run a Perl program, describes CGI scripting and data manipulation, and describes scalar values, basic operators, and associative arrays.

#### Control Your Computer, Simplify Your Life Pragmatic Bookshelf

"Seven Languages in Seven Weeks" presents a meaningful exploration of seven languages within a single book. Rather than serve as a complete reference or installation guide, the book hits what's essential and unique about each language.

#### Infinite Jest Flatiron Books

From the New York Times bestselling author of *Fair Play* comes an inspirational guide for setting new personal goals, rediscovering your interests, cultivating creativity, and reclaiming your Unicorn Space. With her acclaimed New York Times bestseller (and Reese's Book Club pick) *Fair Play*, Eve Rodsky began a national conversation about greater equality on the home front. But she soon realized that even when the domestic workload becomes more balanced, people still report something missing in their lives—that is, unless they create and prioritize time for activities that not only fill their calendars but also unleash their creativity. Rodsky calls this vital time Unicorn Space—the active and open pursuit of creative self-expression in any form that makes you uniquely YOU. To help readers embrace all the unlikely, surprising, and delightful places where their own Unicorn Space may be found, she speaks with trail blazers, thought leaders, academics, and countless real people who have discovered theirs everywhere—from activism to artistic endeavors to second careers. Rodsky reveals what researchers

already know: Creativity is not optional. It's essential. Though most of us do need to remind ourselves how (and where) to find it. With her trademark mix of research based, how-to advice and big-picture inspirational thinking, Rodsky shows you a clear path to reclaim your permission to have fun, manifest your own Unicorn Space in an already too-busy life, and unleash your special gifts and undiscovered talents into the world.

[Real-Time Phoenix](#) Addison-Wesley Professional

Dragons. Art. Revolution. Gyen Jebi isn't a fighter or a subversive. They just want to paint. One day they're jobless and desperate; the next, Jebi finds himself recruited by the Ministry of Armor to paint the mystical sigils that animate the occupying government's automaton soldiers. But when Jebi discovers the depths of the Razanei government's horrifying crimes—and the awful source of the magical pigments they use—they find they can no longer stay out of politics. What they can do is steal Arazi, the ministry's mighty dragon automaton, and find a way to fight...

[Build Awesome Command-Line Applications in Ruby 2](#) Penguin

If you program in C++ you've been neglected. Test-driven development (TDD) is a modern software development practice that can dramatically reduce the number of defects in systems, produce more maintainable code, and give you the confidence to change your software to meet changing needs. But C++ programmers have been ignored by those promoting TDD—until now. In this book, Jeff Langr gives you hands-on lessons in the challenges and rewards of doing TDD in C++. *Modern C++ Programming With Test-Driven Development*, the only comprehensive treatment on TDD in C++ provides you with everything you need to know about TDD, and the challenges and benefits of implementing it in your C++ systems. Its many detailed code examples take you step-by-step from TDD basics to advanced concepts. As a veteran C++ programmer, you're already writing high-quality code, and you work hard to maintain code quality. It doesn't have to be that hard. In this book, you'll learn: how to use TDD to improve legacy C++ systems how to identify and deal with troublesome system dependencies how to do dependency injection, which is particularly tricky in C++ how to use testing tools for C++ that aid TDD new C++11 features that facilitate TDD As you grow in TDD mastery, you'll discover how to keep a massive C++ system from becoming a design mess over time, as well as particular C++ trouble spots to avoid. You'll find out how to prevent your tests from being a maintenance burden and how to think in TDD without giving up your hard-won C++ skills. Finally, you'll see how to grow and sustain TDD in your team. Whether you're a complete unit-testing novice or an experienced tester, this book will lead you to mastery of test-driven development in C++. What You Need A C++ compiler running under Windows or Linux, preferably one that supports C++11. Examples presented in the book were built under gcc 4.7.2. Google Mock 1.6 (downloadable for free; it contains Google Test as well) or an alternate C++ unit testing tool. Most examples in the book are written for Google Mock, but it isn't difficult to translate them to your tool of choice. A good programmer's editor or IDE. cmake, preferably. Of course, you can use your own preferred make too. CMakeLists.txt files are provided for each project. Examples provided were built using cmake version 2.8.9. Various freely-available third-party libraries are used as the basis for examples in the book. These include: cURL JsonCpp Boost (filesystem, date\_time/gregorian, algorithm, assign) Several examples use the boost headers/libraries. Only one example uses cURL and JsonCpp.

*Programming Phoenix* Simon and Schuster

This book is the introduction to Elixir for experienced programmers, completely updated for Elixir 1.6 and beyond.

Explore functional programming without the academic overtones (tell me about monads just one more time). Create concurrent applications, but get them right without all the locking and consistency headaches. Meet Elixir, a modern, functional, concurrent language built on the rock-solid Erlang VM. Elixir's pragmatic syntax and built-in support for metaprogramming will make you productive and keep you interested for the long haul. Maybe the time is right for the Next Big Thing. Maybe it's Elixir. Functional programming techniques help you manage the complexities of today's real-world, concurrent systems; maximize uptime; and manage security. Enter Elixir, with its modern, Ruby-like, extendable syntax, compile and runtime evaluation, hygienic macro system, and more. But, just as importantly, Elixir brings a sense of enjoyment to parallel, functional programming. Your applications become fun to work with, and the language encourages you to experiment. Part 1 covers the basics of writing sequential Elixir programs. We'll look at the language, the tools, and the conventions. Part 2 uses these skills to start writing concurrent code-applications that use all the cores on your machine, or all the machines on your network! And we do it both with and without OTP. Part 3 looks at the more advanced features of the language, from DSLs and code generation to extending the syntax. This edition is fully updated with all the new features of Elixir 1.6, with a new chapter on structuring OTP applications, and new sections on the debugger, code formatter, Distillery, and protocols. What You Need: You'll need a computer, a little experience with another high-level language, and a sense of adventure. No functional programming experience is needed.

**Programming Elixir ≥ 1.6** Pragmatic Bookshelf

Programming Phoenix Productive |> Reliable |> Fast Pragmatic Bookshelf

[Write Less Code, Get More Done \(and Have Fun!\)](#) "O'Reilly Media, Inc."

Your domain is rich and interconnected, and your API should be too. Upgrade your web API to GraphQL, leveraging its flexible queries to empower your users, and its declarative structure to simplify your code. Absinthe is the GraphQL toolkit for Elixir, a functional programming language designed to enable massive concurrency atop robust application architectures. Written by the creators of Absinthe, this book will help you take full advantage of these two groundbreaking technologies. Build your own flexible, high-performance APIs using step-by-step guidance and expert advice you won't find anywhere else. GraphQL is a new way of structuring and building web services, and the result is transformational. Find out how to offer a more tailored, cohesive experience to your users, easily aggregate data from different data sources, and improve your back end's maintainability with Absinthe's declarative approach to defining how your API works. Build a GraphQL-based API from scratch using Absinthe, starting from core principles. Learn the type system and how to expand your schema to suit your application's needs. Discover a growing ecosystem of tools and utilities to understand, debug, and document your API. Take it to production, but do it safely with solid best practices in mind. Find out how complexity analysis and persisted queries can let you support your users flexibly, but responsibly too. Along the way, discover how Elixir makes all the difference for a high performance, fault-tolerant API. Use asynchronous and batching execution, or write your own custom add-ons to extend Absinthe. Go live with subscriptions, delivering data over websockets on top of Elixir (and Erlang/OTP's) famous solid performance and real-time capabilities. Transform your applications with the powerful combination of Elixir and GraphQL, using Absinthe. What You Need: To follow along with the book, you should have Erlang/OTP 19+ and Elixir 1.4+ installed. The book will guide you through setting up a new Phoenix application

using Absinthe.

**New Foundations for a New World** Simon and Schuster  
You know how to code in Elixir; now learn to think in it. Learn to design libraries with intelligent layers that shape the right data structures, flow from one function into the next, and present the right APIs. Embrace the same OTP that's kept our telephone systems reliable and fast for over 30 years. Move beyond understanding the OTP functions to knowing what's happening under the hood, and why that matters. Using that knowledge, instinctively know how to design systems that deliver fast and resilient services to your users, all with an Elixir focus. Elixir is gaining mindshare as the programming language you can use to keep your software running forever, even in the face of unexpected errors and an ever growing need to use more processors. This power comes from an effective programming language, an excellent foundation for concurrency and its inheritance of a battle-tested framework called the OTP. If you're using frameworks like Phoenix or Nerves, you're already experiencing the features that make Elixir an excellent language for today's demands. This book shows you how to go beyond simple programming to designing, and that means building the right layers. Embrace those data structures that work best in functional programs and use them to build functions that perform and compose well, layer by layer, across processes. Test your code at the right place using the right techniques. Layer your code into pieces that are easy to understand and heal themselves when errors strike. Of all Elixir's boons, the most important one is that it guides us to design our programs in a way to most benefit from the architecture that they run on. The experts do it and now you can learn to design programs that do the same. What You Need: Elixir Version 1.7 or greater.

**The Transformative Practice of Taking Nothing for Granted** Pragmatic Bookshelf

Winner of the Shingo Publication Award Accelerate your organization to win in the marketplace. How can we apply technology to drive business value? For years, we've been told that the performance of software delivery teams doesn't matter—that it can't provide a competitive advantage to our companies. Through four years of groundbreaking research to include data collected from the State of DevOps reports conducted with Puppet, Dr. Nicole Forsgren, Jez Humble, and Gene Kim set out to find a way to measure software delivery performance—and what drives it—using rigorous statistical methods. This book presents both the findings and the science behind that research, making the information accessible for readers to apply in their own organizations. Readers will discover how to measure the performance of their teams, and what capabilities they should invest in to drive higher performance. This book is ideal for management at every level.

**Find Your Unicorn Space** Pragmatic Bookshelf

Summary Phoenix is a modern web framework built for the Elixir programming language. Elegant, fault-tolerant, and performant, Phoenix is as easy to use as Rails and as rock-solid as Elixir's Erlang-based foundation. Phoenix in Action builds on your existing web dev skills, teaching you the unique benefits of Phoenix along with just enough Elixir to get the job done. Foreword by Sasa Juric, author of Elixir in Action, Second Edition. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology Modern web applications need to be efficient to develop, lightning fast, and unfailingly reliable. Phoenix, a web framework for the Elixir programming language, delivers on all counts. Elegant and intuitive, Phoenix radically simplifies the dev process. Built for concurrency, Phoenix channels make short work of developing real-time applications. And as for reliability,

Phoenix apps run on the battle-tested Erlang VM, so they're rock solid! About the Book Phoenix in Action is an example-based book that teaches you to build production-quality web apps. You'll handle business logic, database interactions, and app designs as you progressively create an online auction site. As you go, you'll build everything from the core components to the real-time user interactions where Phoenix really shines. What's inside Functional programming in a web environment An introduction to Elixir Database interactions with Ecto Real-time communication with channels About the Reader For web developers familiar with a framework like Rails or ASP.NET. No experience with Elixir or Phoenix required. About the Author Geoffrey Lessel is a seasoned web developer who speaks and blogs about Elixir and Phoenix. Table of Contents PART 1 - GETTING STARTED Ride the Phoenix Intro to Elixir A little Phoenix overview PART 2 - DIVING IN DEEP Phoenix is not your application Elixir application structure Bring in Phoenix Making changes with Ecto.Changeset Transforming data in your browser Plugs, assigns, and dealing with session data Associating records and accepting bids PART 3 - THOSE IMPORTANT EXTRAS Using Phoenix channels for real-time communication Building an API Testing in Elixir and Phoenix *Learning Perl* IT Revolution

Give users the real-time experience they expect, by using Elixir and Phoenix Channels to build applications that instantly react to changes and reflect the application's true state. Learn how Elixir and Phoenix make it easy and enjoyable to create real-time applications that scale to a large number of users. Apply system design and development best practices to create applications that are easy to maintain. Gain confidence by learning how to break your applications before your users do. Deploy applications with minimized resource use and maximized performance. Real-time applications come with real challenges - persistent connections, multi-server deployment, and strict performance requirements are just a few. Don't try to solve these challenges by yourself - use a framework that handles them for you. Elixir and Phoenix Channels provide a solid foundation on which to build stable and scalable real-time applications. Build applications that thrive for years to come with the best-practices found in this book. Understand the magic of real-time communication by inspecting the WebSocket protocol in action. Avoid performance pitfalls early in the development lifecycle with a catalog of common problems and their solutions. Leverage GenStage to build a data pipeline that improves scalability. Break your application before your users do and confidently deploy them. Build a real-world project using solid application design and testing practices that help make future changes a breeze. Create distributed apps that can scale to many users with tools like Phoenix Tracker. Deploy and monitor your application with confidence and reduce outages. Deliver an exceptional real-time experience to your users, with easy maintenance, reduced operational costs, and maximized performance, using Elixir and Phoenix Channels. What You Need: You'll need Elixir 1.9+ and Erlang/OTP 22+ installed on a Mac OS X, Linux, or Windows machine.

**Visualize Your Sensor Data with Phoenix and Grafana** Pragmatic Bookshelf

From the duo behind the massively successful and award-winning podcast Stuff You Should Know comes an unexpected look at things you thought you knew. Josh Clark and Chuck Bryant started the podcast Stuff You Should Know back in 2008 because they were curious—curious about the world around them, curious about what they might have missed in their formal educations, and curious to dig deeper on stuff they thought they understood. As it turns out, they aren't the only curious ones. They've since amassed a rabid fan base, making Stuff You Should Know one of

the most popular podcasts in the world. Armed with their inquisitive natures and a passion for sharing, they uncover the weird, fascinating, delightful, or unexpected elements of a wide variety of topics. The pair have now taken their near-boundless "whys" and "hows" from your earbuds to the pages of a book for the first time—featuring a completely new array of subjects that they've long wondered about and wanted to explore. Each chapter is further embellished with snappy visual material to allow for rabbit-hole tangents and digressions—including charts, illustrations, sidebars, and footnotes. Follow along as the two dig into the underlying stories of everything from the origin of Murphy beds, to the history of facial hair, to the psychology of being lost. Have you ever wondered about the world around you, and wished to see the magic in everyday things? Come get curious with *Stuff You Should Know*. With Josh and Chuck as your guide, there's something interesting about everything (...except maybe jackhammers).

*Functional Programming: A PragPub Anthology Pragmatic Bookshelf*

Elixir and Phoenix are generating tremendous excitement as an unbeatable platform for building modern web applications. For decades OTP has helped developers create incredibly robust, scalable applications with unparalleled uptime. Make the most of them as you build a stateful web app with Elixir, OTP, and Phoenix. Model domain entities without an ORM or a database. Manage server state and keep your code clean with OTP Behaviours. Layer on a Phoenix web interface without coupling it to the business logic. Open doors to powerful new techniques that will get you thinking about web development in fundamentally new ways. Elixir and OTP provide exceptional tools to build rock-solid back-end applications that scale. In this book, you'll build a web application in a radically different way, with a back end that holds application state. You'll use persistent Phoenix Channel connections instead of HTTP's request-response, and create the full application in distinct, decoupled layers. In Part 1, start by building the business logic as a separate

application, without Phoenix. Model the application domain with Elixir functions and simple data structures. By keeping state in memory instead of a database, you can reduce latency and simplify your code. In Part 2, add in the GenServer Behaviour to make managing in-memory state a breeze. Create a supervision tree to boost fault tolerance while separating error handling from business logic. Phoenix is a modern web framework you can layer on top of business logic while keeping the two completely decoupled. In Part 3, you'll do exactly that as you build a web interface with Phoenix. Bring in the application from Part 2 as a dependency to a new Phoenix project. Then use ultra-scalable Phoenix Channels to establish persistent connections between the stateful server and a stateful front-end client. You're going to love this way of building web apps! What You Need: You'll need a computer that can run Elixir version 1.5 or higher and Phoenix 1.3 or higher. Some familiarity with Elixir and Phoenix is recommended.

**Designing Elixir Systems with Otp: Write Highly Scalable, Self-Healing Software with Layers** Rebellion Publishing Ltd

From the incomparable New York Times and New Yorker illustrator Tamara Shopsin, a debut novel about a NYC printer repair technician who comes of age alongside the Apple computer—featuring original artistic designs by the author. NAMED A MOST ANTICIPATED BOOK OF 2021 BY LIT HUB. *LaserWriter II* is a coming-of-age tale set in the legendary 90s indie NYC Mac repair shop TekServe—a voyage back in time to when the internet was new, when New York City was gritty, and when Apple made off-beat computers for weirdos. Our guide is Claire, a 19-year-old who barely speaks to her bohemian co-workers, but knows when it's time to snap on an antistatic bracelet. Tamara Shopsin brings us a classically New York novel that couldn't feel more timely. Interweaving the history of digital technology with a tale both touchingly human and delightfully technical, Shopsin brings an idiosyncratic cast of characters to life with a light touch, a sharp eye, and an unmistakable voice. Filled with pixelated philosophy and lots of printers, *LaserWriter II* is, at its heart, a parable about an apple.