

## Land Of Lisp Learn To Program In One Game At A Time Conrad Barski

Carries readers from the beginning through the proficient stages of learning the GNU Emacs editor, covering everything from simple text editing to moderately complicated customization and programming. Original. (Advanced). Written by a Lisp expert, this is the most comprehensive tutorial on the advanced features of Lisp for experienced programmers. It shows how to program in the bottom-up style that is ideal for Lisp programming, and includes a unique, practical collection of Lisp programming techniques that shows how to take advantage of the language's design for efficient programming in a wide variety of applications.

Level up your skills by taking advantage of Clojure's powerful macro system. Macros make hard things possible and normal things easy. They can be tricky to use, and this book will help you deftly navigate the terrain. You'll discover how to write straightforward code that avoids duplication and clarifies your intentions. You'll learn how and why to write macros. You'll learn to recognize situations when using a macro would (and wouldn't!) be helpful. And you'll use macros to remove unnecessary code and build new language features. Clojure offers some sharp tools in its toolbox, and one of the sharpest is its macro system. This book will help you write macros using Clojure, and more importantly, recognize when you should be using macros in the first place. The Lisp "code-as-data" philosophy gives tremendous advantages to macro authors and users. You can use macros to evaluate code in other contexts, move computations to compile time, and create beautiful API layers. You don't need to wait on the Clojure language itself to add new features, you'll learn how to implement even the lowest-level features as macros. You'll step through representative samples of how to use macros in production libraries and applications, find clear details on how to construct macros, and learn pointers to avoid obstacles that often trip up macro amateurs. Clojure macros are more straightforward to use than metaprogramming features in many other languages, but they're different enough from normal programming to present challenges of their own. Mastering Clojure Macros examines some of these issues, along with alternatives to macros where they exist. By the time you finish this book, you'll be thinking like a macro professional. What You Need:The book examples have been developed under Clojure 1.6.0, although earlier and later versions of Clojure may work as well. You'll want to use Leiningen 2.x in order to follow along with the examples that use external projects.

It's easier to learn how to program a computer than it has ever been before. Now everyone can learn to write programs for themselves - no previous experience is necessary. Chris Pine takes a thorough, but lighthearted approach that teaches you the fundamentals of computer programming, with a minimum of fuss or bother. Whether you are interested in a new hobby or a new career, this book is your doorway into the world of programming. Computers are everywhere and being able to program them is more important than it has ever been. But since most books on programming are written for other programmers, it can be hard to break in. At least it used to be. Chris Pine will teach you how to program. You'll learn to use your computer better, to get it to do what you want it to do. Starting with small, simple one-line programs to calculate your age in seconds, you'll see how to write interactive programs, to use APIs to fetch live data from the internet, to rename your photos from your digital camera, and more. You'll learn the same technology used to drive modern dynamic websites and large, professional applications. Whether you are looking for a fun new hobby or are interested in entering the tech world as a professional, this book gives you a solid foundation in programming. Chris teaches the basics, but also shows you how to think like a programmer. You'll learn through tons of examples, and through programming challenges throughout the book. When you finish, you'll know how and where to learn more - you'll be on your way. What You Need: All you need to learn how to program is a computer (Windows, macOS, or Linux) and an internet connection. Chris Pine will lead you through setting set up with the software you will need to start writing programs of your own.

Learn You a Haskell for Great Good!

Mastering Clojure Macros

Write Cleaner, Faster, Smarter Code

Build Your Own Lisp

Lisp in Small Pieces

An Introduction to Programming and Computing

Power up your Python with object-oriented programming and learn how to write powerful, efficient, and re-usable code. Object-Oriented Python is an intuitive and thorough guide to mastering object-oriented programming from the ground up. You'll cover the basics of building classes and creating objects, and put theory into practice using the pygame package with clear examples that help visualize the object-oriented style. You'll explore the key concepts of object-oriented programming — encapsulation, polymorphism, and inheritance — and learn not just how to code with objects, but the absolute best practices for doing so. Finally, you'll bring it all together by building a complex video game, complete with full animations and sounds. The book covers two fully functional Python code packages that will speed up development of graphical user interface (GUI) programs in Python.

Answer Me This! (http://answermethispodcast.com/) is a weekly interactive comedy podcast, presented by Helen Zaltzman and Olly Mann. In each show Helen and Olly answer listener-submitted questions, as diverse and fantastical as "Why does golf bring out the worst in people?", "Who invented the scotch egg?" and "Why do all girls hate me?" In the Answer Me This! Question Compendium, Helen and Olly bring their trademark irreverence to the page for the first time in an inventive and witty spin on the classic Q&A format. Since its debut in January 2007, Answer Me This! has been chosen as an iTunes Podcast of the Year in 2008, and in 2009 became the first independently produced podcast to be nominated for a Sony Radio Academy Award. With listeners all over the world, and more than 1.5 million downloads per year, the show is now one of the most popular comedy podcasts in Britain.

It's all in the name: Learn You a Haskell for Great Good! is a hilarious, illustrated guide to this complex functional language. Packed with the author's original artwork, pop culture references, and most importantly, useful example code, this book teaches functional fundamentals in a way you never thought possible. You'll start with the kid stuff: basic syntax, recursion, types and type classes. Then once you've got the basics down, the real black belt master-class begins: you'll learn to use applicative functors, monads, zippers, and all the other mythical Haskell constructs you've only read about in storybooks. As you work your way through the author's imaginative (and occasionally insane) examples, you'll learn to:–Laugh in the face of side effects as you wield purely functional programming techniques–Use the magic of Haskell's "laziness" to play with infinite sets of data–Organize your programs by creating your own types, type classes, and modules–Use Haskell's elegant input/output system to share the genius of your programs with the outside world Short of eating the author's brain, you will not find a better way to learn this powerful language than reading Learn You a Haskell for Great Good!

You Will Learn Python 3! Zed Shaw has perfected the world's best system for learning Python 3. Follow it and you will succeed—just like the millions of beginners Zed has taught to date! You bring the discipline, commitment, and persistence; the author supplies everything else. In Learn Python 3 the Hard Way, you'll learn Python by working through 52 brilliantly crafted exercises. Read them. Type their code precisely. (No copying and pasting!) Fix your mistakes. Watch the programs run. As you do, you'll learn how a computer works; what good programs look like; and how to read, write, and think about code. Zed then teaches you even more in 5+ hours of video where he shows you how to break, fix, and debug your code—live, as he's doing the exercises. Install a complete Python environment Organize and write code Fix and break code Basic mathematics Variables Strings and text Interact with users Work with files Looping and logic Data structures using lists and dictionaries

Program design Object-oriented programming Inheritance and composition Modules, classes, and objects Python packaging Automated testing Basic game development Basic web development It'll be hard at first. But soon, you'll just get it—and that will feel great! This course will reward you for every minute you put into it. Soon, you'll know one of the world's most powerful, popular programming languages. You'll be a Python programmer. This Book Is Perfect For Total beginners with zero programming experience Junior developers who know one or two languages Returning professionals who haven't written code in years Seasoned professionals looking for a fast, simple, crash course in Python 3

All-in-One: Learn Maya, 3ds Max, ZBrush, and Photoshop Winning Techniques

Transforming Programs with Programs

Programming Algorithms in Lisp

A Beginner's Guide

On Lisp

ANSI Common Lisp

*Let Over Lambda is one of the most hardcore computer programming books out there. Starting with the fundamentals, it describes the most advanced features of the most advanced language: Common Lisp. Only the top percentile of programmers use lisp and if you can understand this book you are in the top percentile of lisp programmers. If you are looking for a dry coding manual that re-hashes common-sense techniques in whatever langue du jour, this book is not for you. This book is about pushing the boundaries of what we know about programming. While this book teaches useful skills that can help solve your programming problems today and now, it has also been designed to be entertaining and inspiring. If you have ever wondered what lisp or even programming itself is really about, this is the book you have been looking for.*

*The craft of writing and marketing a book has changed significantly over the past 10 years. It is no longer enough to just write a good book; you have to write for a specific audience and connect to your readers long before the book hits the shelves. Releasing part of your work early allows you to make adjustments to your book—or even discard your project entirely in order to invest your time into a better book idea. Using modern project management methods, you can organize your work into individual steps ("user stories"), and reuse them to market your book. Organizing the book into logical sections helps you to create preview materials for blog posts or newsletters. In addition, this will ensure that you make steady progress, avoid getting lost in the details and achieve high quality consistently throughout your book. --- Do you recognize yourself in one of these people? This book is for "Peter." - First-time author. - Has a "complete" script, "had a friend look at it," and now wants to publish it. - Might need (unsolicited) advice to properly edit it instead of just going through a "self-edit." - Needs to be reminded about the difficulties of selling a book. Has no idea about marketing. - Has not worked with an editor. - Creates his own book covers. - Would benefit from a "pep talk." This book is for "Mary." - Writes novels in Word but now wants to write a non-fiction book. - Undecided about what tools to use. - Works with an editor, but she and her editor have no real work structure. - Does not know how to market, find market niches, etc. Her past successes were random, and she never knew if her latest novel would sell or not. This book is for "John." - Professional editor seeking to expand his services from merely editing Word files to helping release books online. - Also is looking for better project management techniques to help guide an author along the way. - Often works in scientific fields and thus has to manage a lot of bibliographical references. - Spends lots of time indexing books. - Is OK with a LaTeX template but seeks to get a head start by making adjustments to it. This book is for "George." - LaTeX expert who wants to publish his work as an e-book. - Needs basic direction and then figures out the rest on his own. - Plans to do a series with a glossary and often needs to reuse blocks of text. - Needs some help in terms of book design, polishing, and graphics. - Loves to share work and collaborate with others. This book is for "Tina." - Professional self-publisher who is seeking additional ideas to improve her publishing process. - Looks for ways to establish herself as a brand and create a network of readers. This book is for "Clara." - Wants to write a book about her profession in order to establish herself as an expert but has no idea where to start. --- Table of Contents: - Great Expectations - Incorporate Books into Your Professional Career - Starting a New Book - What to Keep and What to Remove - Selecting Personas - How to Organize Your Ideas - How to Organize Your Ideas (Fiction Books) - The Rules of Your Book - How to Optimize the Work Process - How to Get Early Feedback from Readers - How We Can Help with Project Management*

*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.*

*Modern web applications are built on a tangle of technologies that have been developed over time and then haphazardly pieced together. Every piece of the web application stack, from HTTP requests to browser-side scripts, comes with important yet subtle security consequences. To keep users safe, it is essential for developers to confidently navigate this landscape. In The Tangled Web, Michal Zalewski, one of the world's top browser security experts, offers a compelling narrative that explains exactly how browsers work and why they're fundamentally insecure. Rather than dispense simplistic advice on vulnerabilities, Zalewski examines the entire browser security model, revealing weak points and providing crucial information for shoring up web application security. You'll learn how to:–Perform common but surprisingly complex tasks such as URL parsing and HTML sanitization–Use modern security features like Strict Transport Security, Content Security Policy, and Cross-Origin Resource Sharing–Leverage many variants of the same-origin policy to safely compartmentalize complex web applications and protect user credentials in case of XSS bugs–Build mashups and embed gadgets without getting stung by the tricky frame navigation policy–Embed or host user-supplied content without running into the trap of content sniffing For quick reference, "Security Engineering Cheat Sheets" at the end of each chapter offer ready solutions to problems you're most likely to encounter. With coverage extending as far as planned HTML5 features, The Tangled Web will help you create secure web applications that stand the test of time.*

Realm of Racket

Learn to Program in Lisp, One Game at a Time!

From Strings to Turing Machines

Answer Me This

The Seasoned Schemer, second edition

Writing Better Books the Agile Way

This is a comprehensive account of the semantics and the implementation of the whole Lisp family of languages, namely Lisp, Scheme and related dialects. It describes 11 interpreters and 2 compilers, including very recent techniques of interpretation and compilation. The book is in two parts. The first starts from a simple evaluation function and enriches it with multiple name spaces, continuations and side-effects with commented variants, while at the same time the language used to define these features is reduced to a simple lambda-calculus. Denotational semantics is then naturally introduced. The second part focuses more on implementation techniques and discusses precompilation for fast interpretation: threaded code or bytecode; compilation towards C. Some extensions are also described such as dynamic evaluation, reflection, macros and objects. This will become the new standard reference for people wanting to know more about the Lisp family of languages: how they work, how they are implemented, what their variants are and why such variants exist. The full code is supplied (and also available over the Net). A large bibliography is given as well as a considerable number of exercises. Thus it may also be used by students to accompany second courses on Lisp or Scheme.

A compilation of key chapters from the top Focal game art books available today - in the areas of Max, Maya, Photoshop, and ZBrush. The chapters provide the CG Artist with an excellent sampling of essential techniques that every 3D artist needs to create stunning game art. Game artists will be able to master the modeling, rendering, rigging, and texturing techniques they need - with advice from Focal's best and brightest authors. Artists can learn hundreds of tips, tricks and shortcuts in Max, Maya, Photoshop, ZBrush - all within the covers of one complete, inspiring reference.

\* Treats LISP as a language for commercial applications, not a language for academic AI concerns. This could be considered to be a secondary text for the Lisp course that most schools teach . This would appeal to students who sat through a LISP course in college without quite getting it - so a "nostalgia" approach, as in "wow-lisp can be practical..." \* Discusses the Lisp programming model and environment. Contains an introduction to the language and gives a thorough overview of all of Common Lisp's main features. \* Designed for experienced programmers no matter what languages they may be coming from and written for a modern audience—programmers who are familiar with languages like Java, Python, and Perl. \* Includes several examples of working code that actually does something useful like Web programming and database access.

Introduction: getting acquainted. Functions and data. Lists. EVAL notation. Conditionals. Global variables and side effects. List data structures. Applicative operators. Recursion. Elementary input/output. Iteration. Property lists. Recommended further reading. Dialects of Lisp. Extensions to Lisp. Index.

The Little Mler

Racket Programming the Fun Way

Common LISP

The Tangled Web

50 Years of Lisp

Practical Common Lisp

*Highly effective thinking is an art that engineers and scientists can be taught to develop. By presenting actual experiences and analyzing them as they are described, the author conveys the developmental thought processes employed and shows a style of thinking that leads to successful results is something that can be learned. Along with spectacular successes, the author also conveys how failures contributed to shaping the thought processes. Provides the reader with a style of thinking that will enhance a person's ability to function as a problem-solver of complex technical issues. Consists of a collection of stories about the author's participation in significant discoveries, relating how those discoveries came about and, most importantly, provides analysis about the thought processes and reasoning that took place as the author and his associates progressed through engineering problems.*

*Summary The Joy of Clojure, Second Edition is a deep look at the Clojure language. Fully updated for Clojure 1.6, this new edition goes beyond just syntax to show you the "why" of Clojure and how to write fluent Clojure code. You'll learn functional and declarative approaches to programming and will master the techniques that make Clojure so elegant and efficient. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology The Clojure programming language is a dialect of Lisp that runs on the Java Virtual Machine and JavaScript runtimes. It is a functional programming language that offers great performance, expressive power, and stability by design. It gives you built-in concurrency and the predictable precision of immutable and persistent data structures. And it's really, really fast. The instant you see long blocks of Java or Ruby dissolve into a few lines of Clojure, you'll know why the authors of this book call it a "joyful language." It's no wonder that enterprises like Staples are betting their infrastructure on Clojure. About the Book The Joy of Clojure, Second Edition is a deep account of the Clojure language. Fully updated for Clojure 1.6, this new edition goes beyond the syntax to show you how to write fluent Clojure code. You'll learn functional and declarative approaches to programming and will master techniques that make Clojure elegant and efficient. The book shows you how to solve hard problems related to concurrency, interoperability, and performance, and how great it can be to think in the Clojure way. Appropriate for readers with some experience using Clojure or common Lisp. What's Inside Build web apps using ClojureScript Master functional programming techniques Simplify concurrency Covers Clojure 1.6 About the Authors Michael Fogus and Chris Houser are contributors to the Clojure and ClojureScript programming languages and the authors of various Clojure libraries and language features. Table of Contents PART 1 FOUNDATIONS Clojure philosophy Drinking from the Clojure fire hose Dipping your toes in the pool PART 2 DATA TYPES On scalars Collection types PART 3 FUNCTIONAL PROGRAMMING Being lazy and set in your ways Functional programming PART 4 LARGE-SCALE DESIGN Macros Combining data and code Mutation and concurrency Parallelism PART 5 HOST SYMBIOSIS Java.next Why ClojureScript? PART 6 TANGENTIAL CONSIDERATIONS Data-oriented programming Performance Thinking programs Clojure changes the way you think*

*This book had its genesis in the following piece of computer mail: From allegra|joan-b Tue Dec 18 09:15:54 1984 To: sola|hjb Subject: lispm Hank, I've been talking with Mark Plotnik and Bill Gale about asking you to conduct a basic course on using the lisp machine. Mark, for instance, would really like to cover basics like the flavor system, etc., so he could start doing his own programming without a lot of trial and error, and Bill and I would be interested in this, too. I'm quite sure that Mark Jones, Bruce, Eric and Van would also be really interested. Would you like to do it? Bill has let me know that if you'd care to set something up, he's free to meet with us anytime this week or next (although I'll only be here on Wed. next week) so we can come up with a plan. What do you think? Joan.*

*Teaching users new and more powerful ways of thinking about programs, this two-in-one text contains a tutorial--full of examples--that explains all the essential concepts of Lisp programming, plus an up-to-date summary of ANSI Common Lisp. Informative and fun, it gives users everything they need to start writing programs in Lisp and highlights innovative Lisp features.*

Learn to Program with C

The Little LISPer

Taming Complex Software with Functional Thinking

Writing Efficient Programs with Examples in ANSI Common Lisp

A JavaScript and JQuery Developer's Guide

Successful Lisp: How to Understand and Use Common Lisp

with a foreword by Robin Milnerand drawings by Duane Bibby Over the past few years, ML has emerged as one of the most important members of the family of programming languages. Many professors in the United States and other countries use ML to teach courses on the principles of programming and on programming languages. In addition, ML has emerged as a natural language for software engineering courses because it provides the most sophisticated and expressive module system currently available.Pelleisen and Friedman are well known for gently introducing readers to difficult ideas. The Little Mler is an introduction to thinking about programming and the ML programming language. The authors introduce those new to programming, as well as those experienced in other programming languages, to the principles

of types, computation, and program construction. Most important, they help the reader to think recursively with types about programs.

In just 24 sessions of one hour or less, this guide will help you create great 2D and 3D games for any platform with the 100% free Godot 3.0 game engine. Its straightforward, step-by-step approach guides you from basic scenes, graphics, and game flow through advanced shaders, environments, particle rendering, and networked games. Godot's co-creator and main contributor walk you through building three complete games, offering advanced techniques you won't find anywhere else. Every lesson builds on what you've already learned, giving you a rock-solid foundation for real-world success. Step-by-step instructions carefully walk you through the most common Godot engine programming tasks and techniques Practical, hands-on examples show you how to apply what you learn Quizzes and exercises help you test your knowledge and stretch your skills Notes and tips point out shortcuts, solutions, and problems to avoid Learn how to... · Install Godot, create projects, and use the visual editor · Master the scene system, and organize games with Scene Trees · Create 2D graphics, 3D graphics, and animations · Use basic and advanced scripting to perform many game tasks · Process player input from any source · Control game flow, configurations, and resources · Maximize realism with Godot's physics and particle systems · Make the most of 3D shaders, materials, lighting, and shadows · Control effects and post-processing · Build richer, more sophisticated game universes with viewports · Develop networked games, from concepts to communication and input · Export games to the devices you've targeted · Integrate native code, third-party APIs, and engine extensions (bonus chapter)

Most Perl programmers were originally trained as C and Unix programmers, so the Perl programs that they write bear a strong resemblance to C programs. However, Perl incorporates many features that have their roots in other languages such as Lisp. These advanced features are not well understood and are rarely used by most Perl programmers, but they are very powerful. They can automate tasks in everyday programming that are difficult to solve in any other way. One of the most powerful of these techniques is writing functions that manufacture or modify other functions. For example, instead of writing ten similar functions, a programmer can write a general pattern or framework that can then create the functions as needed according to the pattern. For several years Mark Jason Dominus has worked to apply functional programming techniques to Perl. Now Mark brings these flexible programming methods that he has successfully taught in numerous tutorials and training sessions to a wider audience. \* Introduces powerful programming methods new to most Perl programmers that were previously the domain of computer scientists \* Gradually builds up confidence by describing techniques of progressive sophistication \* Shows how to improve everyday programs and includes numerous engaging code examples to illustrate the methods

Unless you've been living under a rock for the last couple of years, you've probably heard of Bitcoin—the game-changing digital currency used by millions worldwide. But Bitcoin isn't just another way to buy stuff. It's an anonymous, revolutionary, cryptographically secure currency that functions without the oversight of a central authority or government. If you want to get into the Bitcoin game but find yourself a little confused, Bitcoin for the Befuddled may be just what you're looking for. Learn what Bitcoin is; how it works; and how to acquire, store, and spend bitcoins safely and securely. You'll also learn: Bitcoin's underlying cryptographic principles, and how bitcoins are createdThe history of Bitcoin and its potential impact on trade and commerceAll about the blockchain, the public ledger of Bitcoin transactionsHow to choose a bitcoin wallet that's safe and easy to useHow to accept bitcoins as payment in your physical store or on your websiteAdvanced topics, including Bitcoin mining and Bitcoin programming With its non-technical language and patient, step-by-step approach to this fascinating currency, Bitcoin for the Befuddled is your ticket to getting started with Bitcoin. Get out from under the rock and get in the Bitcoin game. Just make sure not to lose your shirt.

Learning GNU Emacs

Self-Publish Your Book on Amazon and Google

Learn Python 3 the Hard Way

Learn to Program, One Game at a Time!

Advanced Techniques for Common Lisp

Object-Oriented Python

For weeks, months—nay!—from the very moment you were born, you 've felt it calling to you. At long last you 'll be united with the programming language you 've been longing for: Clojure! As a Lisp-style functional programming language, Clojure lets you write robust and elegant code, and because it runs on the Java Virtual Machine, you can take advantage of the vast Java ecosystem. Clojure for the Brave and True offers a "dessert-first" approach: you 'll start playing with real programs immediately, as you steadily acclimate to the abstract but powerful features of Lisp and functional programming. Inside you 'll find an offbeat, practical guide to Clojure, filled with quirky sample programs that catch cheese thieves and track glittery vampires. Learn how to: –Wield Clojure 's core functions –Use Emacs for Clojure development –Write macros to modify Clojure itself –Use Clojure 's tools to simplify concurrency and parallel programming Clojure for the Brave and True assumes no prior experience with Clojure, the Java Virtual Machine, or functional programming. Are you ready, brave reader, to meet your true destiny? Grab your best pair of parentheses—you 're about to embark on an epic journey into the world of Clojure!

Master algorithms programming using Lisp, including the most important data structures and algorithms. This book also covers the essential tools that help in the development of algorithmic code to give you all you need to enhance your code. Programming Algorithms in Lisp shows real-world engineering considerations and constraints that influence the programs that use these algorithms. It includes practical use cases of the applications of the algorithms to a variety of real-world problems. What You Will Learn Program algorithms using the Lisp programming language Work with data structures, arrays, key-values, hash-tables, trees, graphs, and more Use dynamic programming Program using strings Work with approximations and compression Who This Book Is For Intermediate Lisp programmers wanting to do algorithms programming. A very experienced non-Lisp programmer may be able to benefit from this book as well.

Lisp has been hailed as the world 's most powerful programming language, but its cryptic syntax and academic reputation can be enough to scare off even experienced programmers. Those dark days are finally over—Land of Lisp brings the power of functional programming to the people! With his brilliantly quirky comics and out-of-this-world games, longtime Lisper Conrad Barski teaches you the mysteries of Common Lisp. You 'll start with the basics, like list manipulation, I/O, and recursion, then move on to more complex topics like macros, higher order programming, and domain-specific languages. Then, when your brain overheats, you can kick back with an action-packed comic book interlude! Along the way you 'll create (and play) games like Wizard Adventure, a text adventure with a whiskey-soaked twist, and Grand Theft Wumpus, the most violent version of Hunt the Wumpus the world has ever seen. You'll learn to: –Master the quirks of Lisp 's syntax and semantics –Write concise and elegant functional programs –Use macros, create domain-specific languages, and learn other advanced Lisp techniques –Create your own web server, and use it to play browser-based games –Put your Lisp skills to the test by writing brain-melting games like Dice of Doom and Orc Battle With Land of Lisp, the power of functional programming is yours to wield.

With Learning JavaScript Design Patterns, you 'll learn how to write beautiful, structured, and maintainable JavaScript by applying classical and modern design patterns to the language. If you want to keep your code efficient, more manageable, and up-to-date with the latest best practices, this book is for you. Explore many popular design patterns, including Modules, Observers, Facades, and Mediators. Learn how modern architectural patterns—such as MVC, MVP, and MVVM—are useful from the perspective of a modern web application developer. This book also walks experienced JavaScript developers through modern module formats, how to namespace code effectively, and other essential topics. Learn the structure of design patterns and how they are written Understand different pattern categories, including creational, structural, and behavioral Walk through more than 20 classical and modern design patterns in JavaScript Use several options for writing modular code—including the Module pattern, Asynchronous Module Definition (AMD), and CommonJS Discover design patterns implemented in the jQuery library Learn popular design patterns for writing maintainable jQuery plug-ins "This book should be in every JavaScript developer 's hands. It 's the go-to book on JavaScript patterns that will be read and referenced many times in the future."—André e Hansson, Lead Front-End Developer, presis!

Land of Lisp

Bitcoin for the Befuddled

A Gentle Introduction to Symbolic Computation

Master OOP by Building Games and GUIs

Grokking Simplicity

Godot Engine Game Development in 24 Hours, Sams Teach Yourself

Highly accessible treatment covers cons cell structures, evaluation rules, programs as data, recursive and applicable programming styles. Nearly 400 illustrations, answers to exercises, "toolkit" sections, and a variety of complete programs. 1990 edition.

Racket is a descendant of Lisp, a programming language renowned for its elegance, power, and challenging learning curve. But while Racket retains the functional goodness of Lisp, it was designed with beginning programmers in mind. Realm of Racket is your introduction to the Racket language. In Realm of Racket, you'll learn to program by creating increasingly complex games. Your journey begins with the Guess My Number game and coverage of some basic Racket etiquette. Next you'll dig into syntax and semantics, lists, structures, and conditionals, and learn to work with recursion and the GUI as you build the Robot Snake game. After that it's on to lambda and mutant structs (and an Orc Battle), and fancy loops and the Dice of Doom. Finally, you'll explore laziness, AI, distributed games, and the Hungry Henry game. As you progress through the games, chapter checkpoints and challenges help reinforce what you've learned. Offbeat comics keep things fun along the way. As you travel through the Racket realm, you'll: – Master the quirks of Racket's syntax and semantics – Learn to write concise and elegant functional programs – Create a graphical user interface using the 2htdp/image library – Create a server to handle true multiplayer games

Realm of Racket is a lighthearted guide to some serious programming. Read it to see why Racketeers have so much fun!

This book teaches computer programming to the complete beginner using the native C language. As such, it assumes you have no knowledge whatsoever about programming. The main goal of this book is to teach fundamental programming principles using C, one of the most widely used programming languages in the world today. We discuss only those features and statements in C that are necessary to achieve our goal. Once you learn the principles well, they can be applied to any language. If you are worried that you are not good at high-school mathematics, don 't be. It is a myth that you must be good at mathematics to learn programming. C is considered a 'modern ' language even though its roots date back to the 1970s. Originally, C was designed for writing 'systems ' programs—things like operating systems, editors, compilers, assemblers and input/output utility programs. But, today, C is used for writing all kinds of applications programs as well—word processing programs, spreadsheet programs, database management programs, accounting programs, games, robots, embedded systems/electronics (i.e., Arduino), educational software—the list is endless. Note: Appendices A-D are available as part of the free source code download at the Apress website. What You Will Learn: How to get started with programming using the C language How to use the basics of C How to program with sequence, selection and repetition logic How to work with characters How to work with functions How to use arrays Who This Book Is For: This book is intended for anyone who is learning programming for the first time.

Despite using them every day, most software engineers know little about how programming languages are designed and implemented. For many, their only experience with that corner of computer science was a terrifying "compilers" class that they suffered through in undergrad and tried to blot from their memory as soon as they had scribbled their last NFA to DFA conversion on the final exam. That fearsome reputation belies a field that is rich with useful techniques and not so difficult as some of its practitioners might have you believe. A better understanding of how programming languages are built will make you a stronger software engineer and teach you concepts and data structures you'll use the rest of your coding days. You might even have fun. This book teaches you everything you need to know to implement a full-featured, efficient scripting language. You'll learn both high-level concepts around parsing and semantics and gritty details like bytecode representation and garbage collection. Your brain will light up with new ideas, and your hands will get dirty and calloused. Starting from main(), you will build a language that features rich syntax, dynamic typing, garbage collection, lexical scope, first-class functions, closures, classes, and inheritance. All packed into a few thousand lines of clean, fast code that you thoroughly understand because you wrote each one yourself.

The Official Guide to Godot 3.0

A Very Simple Introduction to the Terrifyingly Beautiful World of Computers and Code

How to Design Programs, second edition

The Joy of Clojure

Learn the Ultimate Language and Become a Better Programmer

LISP

*The notion that "thinking about computing is one of the most exciting things the human mind can do" sets both The Little Schemer (formerly known as The Little LISPer) and its new companion volume, The Seasoned Schemer, apart from other books on LISP. The authors' enthusiasm for their subject is compelling as they present abstract concepts in a humorous and easy-to-grasp fashion. Together, these books will open new doors of thought to anyone who wants to find out what computing is really about. The Little Schemer introduces computing as an extension of arithmetic and algebra; things that everyone studies in grade school and high school. It introduces programs as recursive functions and briefly discusses the limits of what computers can do. The authors use the programming language Scheme, and interesting foods to illustrate these abstract ideas. The Seasoned Schemer informs the reader about additional dimensions of computing: functions as values, change of state, and exceptional cases. The Little LISPer has been a popular introduction to LISP for many years. It had appeared in French and Japanese. The Little Schemer and The Seasoned Schemer are worthy successors and will prove equally popular as textbooks for Scheme courses as well as companion texts for any complete introductory course in Computer Science.*

*If you've ever wondered how to build your own programming language or wanted to learn C but weren't sure where to start, this is the book for you. In under 1000 lines of code you'll start building your very own programming language, and in doing so learn how to program in C, one of the world's most important programming languages. Along the way we'll learn about the weird and wonderful nature of Lisps, the unique techniques behind function programming, the methods used to concisely solve problems, and the art of writing beautiful code. Build Your Own Lisp is a fun and creative journey through a fascinating area of computer science, and an essential read for any programmer, new or old!*

*A completely revised edition, offering new design recipes for interactive programs and support for images as plain values, testing, event-driven programming, and even distributed programming. This introduction to programming places computer science at the core of a liberal arts education. Unlike other introductory books, it focuses on the program design process, presenting program design guidelines that show the reader how to analyze a problem statement, how to formulate concise goals, how to make up examples, how to develop an outline of the solution, how to finish the program, and how to test it. Because learning to design programs is about the study of principles and the acquisition of transferable skills, the text does not use an off-the-shelf industrial language but presents a tailor-made teaching language. For the same reason, it offers DrRacket, a programming environment for novices that supports playful, feedback-oriented learning. The environment grows with readers as they master the material in the book until it supports a full-fledged language for the whole spectrum of programming tasks. This second edition has been completely revised. While the book continues to teach a systematic approach to program design, the second edition introduces different design recipes for interactive programs with graphical interfaces and batch programs. It also enriches its design recipes for functions with numerous new hints. Finally, the teaching languages and their IDE now come with support for images as plain values, testing, event-driven programming, and even distributed programming.*

*Even experienced developers struggle with software systems that sprawl across distributed servers and APIs, are filled with redundant code, and are difficult to reliably test and modify. Grokking Simplicity is a friendly, practical guide that will change the way you approach software design and development. Even experienced developers struggle with software systems that sprawl across distributed servers and APIs, are filled with redundant code, and are difficult to reliably test and modify. Grokking Simplicity is a friendly, practical guide that will change the way you approach software design and development. Grokking Simplicity guides you to a crystal-clear understanding of why certain features of modern software are so prone to complexity and introduces you to the functional techniques you can use to simplify these systems so that they're easier to read, test, and debug. Through hands-on examples, exercises, and numerous self-assessments, you'll learn to organize your code for maximum reusability and internalize methods to keep unwanted complexity out of your codebase. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications.*

Learning JavaScript Design Patterns

Lisp Lore: A Guide to Programming the Lisp Machine

Learning to Learn

Art of Doing Science and Engineering

A Guide to Securing Modern Web Applications

Learn to Program