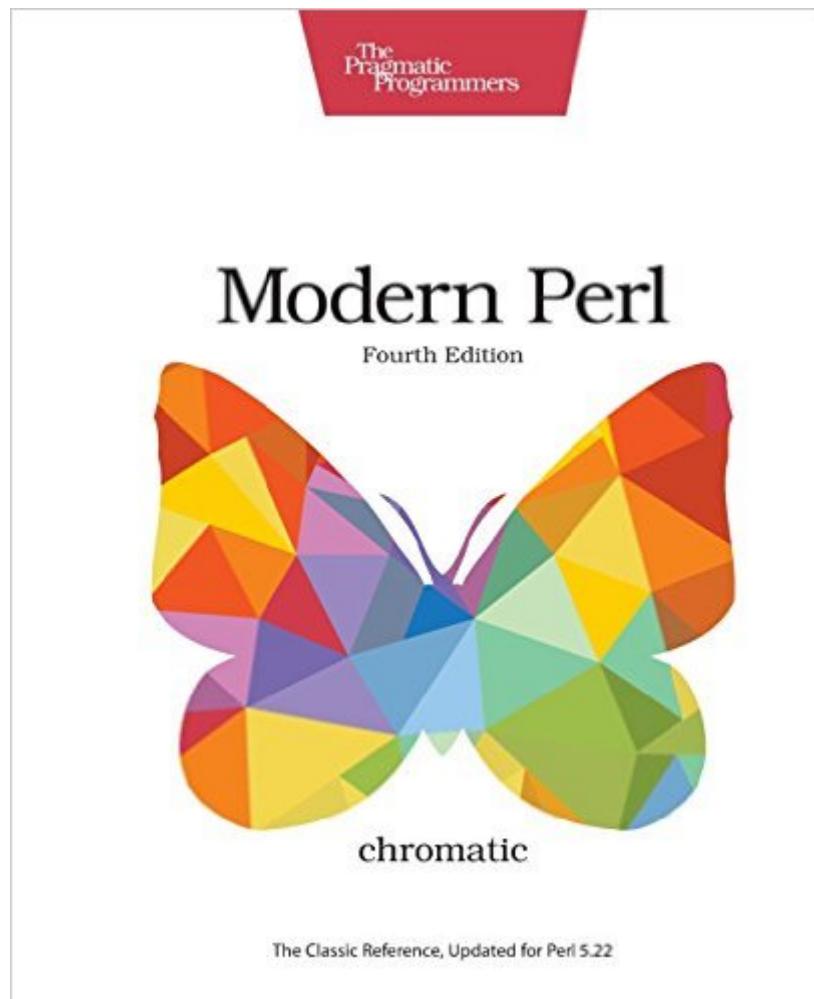


The book was found

Modern Perl



Synopsis

A Perl expert can solve a problem in a few lines of well-tested code. Now you can unlock these powers for yourself. Modern Perl teaches you how Perl really works. It's the only book that explains Perl thoroughly, from its philosophical roots to the pragmatic decisions that help you solve real problems--and keep them solved. You'll understand how the language fits together and discover the secrets used by the global Perl community. This beloved guide is now completely updated for Perl 5.22. When you have to solve a problem now, reach for Perl. When you have to solve a problem right, reach for Modern Perl. Discover how to scale your skills from one-liners to asynchronous Unicode-aware web services and everything in between. Modern Perl will take you from novice to proficient Perl hacker. You'll see which features of modern Perl will make you more productive, and which features of this well-loved language are best left in the past. Along the way, you'll take advantage of Perl to write well-tested, clear, maintainable code that evolves with you. Learn how the language works, how to take advantage of the CPAN's immense trove of time-tested solutions, and how to write clear, concise, powerful code that runs everywhere. Specific coverage explains how to use Moose, how to write testable code, and how to deploy and maintain real-world Perl applications. This new edition covers the new features of Perl 5.20 and Perl 5.22, including all the new operators, standard library changes, bug and security fixes, and productivity enhancements. It gives you what you need to use the most up-to-date Perl most effectively, all day, every day. What You Need: Perl 5.16 or newer (Perl 5.20 or 5.22 preferred). Installation/upgrade instructions included.

Book Information

Paperback: 300 pages

Publisher: Pragmatic Bookshelf; 4 edition (October 29, 2015)

Language: English

ISBN-10: 1680500880

ISBN-13: 978-1680500882

Product Dimensions: 7.5 x 0.6 x 9.2 inches

Shipping Weight: 12.6 ounces (View shipping rates and policies)

Average Customer Review: 4.4 out of 5 stars Â See all reviews Â (5 customer reviews)

Best Sellers Rank: #585,383 in Books (See Top 100 in Books) #40 in Â Books > Computers & Technology > Programming > Languages & Tools > Perl #120 in Â Books > Computers & Technology > Networking & Cloud Computing > Network Administration > Linux & UNIX

Customer Reviews

Five Things Perl (Still) Gets Right by chromatic In the late 1990s, I switched from Java to Perl because Perl made things easier. Perl let me get things done. Almost 20 years later, Perl still has a special place in my toolbox. I keep using it for certain projects because it still does some things so very well. I can get many of these features from a lot of languages and platforms, and that's great. Yet the combination available from Perl and its ecosystem keeps me coming back to the language.

Compatibility The first serious program I wrote in Perl was a statistical analysis of random number sequences. It was last modified on February 26, 2003 and still runs, untouched. This program isn't unique in its longevity. Most programs written to run even for Perl 3 or 4 will run today on Perl 5.22. In the 8 major and 38 minor stable releases of Perl since 2000, nearly any program written will run without modification. The expectation of compatibility guides people to keep existing code working.

Quality Perl has a standard test suite that's expected to pass on every platform with every commit. It has monthly unstable releases so that intrepid volunteer testers can test the CPAN--Perl's large library of freely reusable third-party code--against the latest development versions. A project called *Bleadperl Breaks CPAN*; bisect commits to Perl itself to find out exactly which change breaks which modules when something goes wrong. In practice, it's possible to update an application to run on the newest release of Perl the day it's released with no changes to your code. Just install the new version, install your module dependencies, and go.

Usability Writing documentation as a programmer is like flossing: everyone knows they should do it, but most of us don't do it often enough. Yet Perl's documentation standard has set a bar that few other projects I've seen have ever matched. Perl set this standard early; it includes voluminous documentation of the language, core libraries, and even its expected documentation format. CPAN modules have a well-established standard for documentation, including running code you can often copy and paste and modify into your own programs.

Scalability One of Larry Wall's early design goals was to fill the gap between one-off shell scripts and serious programs written in C. I like to think he took the ease of prototyping and low ceremony of shell scripting and combined it with the full power of Unix available from C. If you plotted the sizes of all the programs I've written in the past couple of decades, you'd

see a wide range: lots of one-liners, countless Unix filter scripts, and larger programs of tens of thousands or hundreds of thousands of lines. That's not by accident; scalability of programmer effort is a deliberate goal of Perl that's pervasive through much of the CPAN as well. Availability Perl's been available on every professional system I've used in recent memory. It was quicker and easier to write a simple log parser in Perl on a new work laptop than to make sure I had all the development dependencies installed to install the correct other language dependencies to use the libraries I'd need there. (Though to be fair, I'm really good at writing simple log parsers after decades of Unix experience.) Going from a fresh account on a new machine to a new account can use `Spreadsheet::WriteExcel` to fill in the gaps of an existing spreadsheet just uploaded by a user--is easy--in part because I know all the pieces, but in part because I have confidence that they're stable, well-tested, work together, and are available not just on my new laptop but on my server. I appreciate that Perl works and continues to work.

Courtesy of Stripgenerator [View larger](#)

[Download to continue reading...](#)

Effective Perl Programming: Ways to Write Better, More Idiomatic Perl (Effective Software Development Series) Modern Perl Raspberry Pi 3: A Simple Guide to Help You Get the Most Out of Your Raspberry Pi 3 (Raspberry Pi, Python, Raspberry Pi 2, Perl, Programming, Raspberry Pi 3, Ruby) Win32 Perl Programming: The Standard Extensions (2nd Edition) Perl Resource Kit Win32 Edition Cross-Platform Perl Regular Expression Pocket Reference: Regular Expressions for Perl, Ruby, PHP, Python, C, Java and .NET (Pocket Reference (O'Reilly)) Perl Hacks: Tips & Tools for Programming, Debugging, and Surviving Java Programming: Master's Handbook: A TRUE Beginner's Guide! Problem Solving, Code, Data Science, Data Structures & Algorithms (Code like a PRO in 24 ... design, tech, perl, ajax, swift, python) Perl by Example Learning Perl Student Workbook An Introduction to Language Processing with Perl and Prolog: 260 (Cognitive Technologies) Ruby: Programming, Master's Handbook: A TRUE Beginner's Guide! Problem Solving, Code, Data Science, Data Structures & Algorithms (Code like a PRO in 24 ... design, tech, perl, ajax, swift, python) Modern Essentials Bundle - Modern Essentials *7th Edition* a Contemporary Guide to the Therapeutic Use of Essential Oils, an Intro to Modern Essentials, Reference Card, and Aroma Designs Bookmark Standard Catalog of World Paper Money - Modern Issues (DVD) (Standard Catalog of World Paper Money: Modern Issues, 1961-Present) Standard Catalog of World Paper Money: Modern Issues 1961 - Present (Standard Catalog of World Paper

Money: Vol.3: Modern Issues) Manuscript Miscellanies in Early Modern England (Material Readings in Early Modern Culture) Modern Construction: Roofs (Modern Construction Series) Modern Collectible Tins (Modern Collectible Tins: Identification & Values) Civil Uprisings in Modern Sudan: The 'Khartoum Springs' of 1964 and 1985 (A Modern History of Politics and Violence)

[Dmca](#)