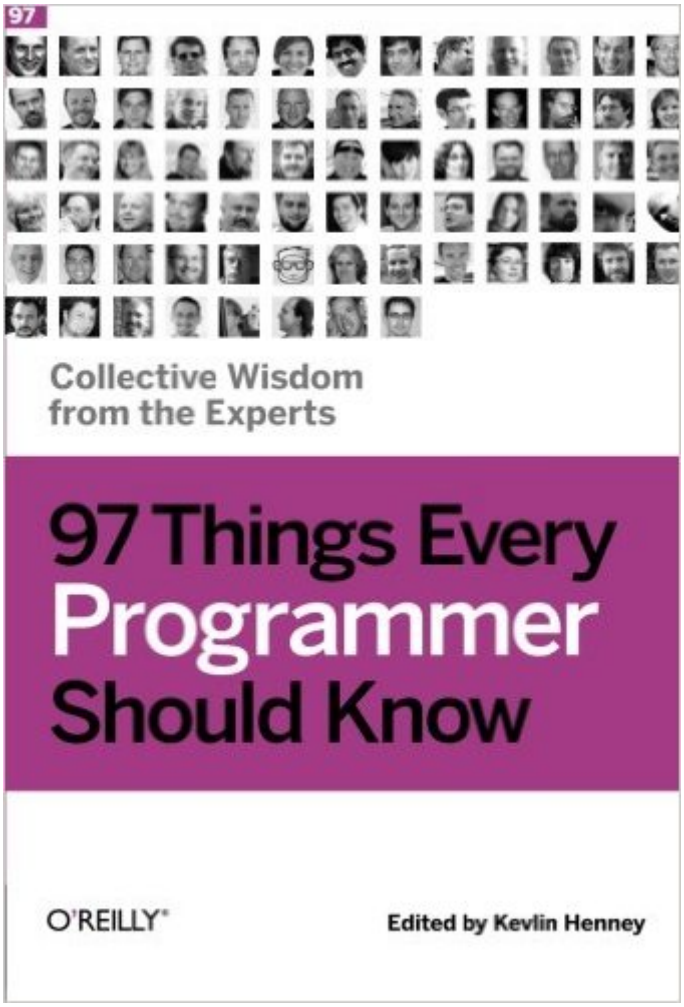
97 Things Every Programmer Should Know: Collective Wisdom From The Experts



Synopsis

Tap into the wisdom of experts to learn what every programmer should know, no matter what language you use. With the 97 short and extremely useful tips for programmers in this book, you'll expand your skills by adopting new approaches to old problems, learning appropriate best practices, and honing your craft through sound advice.With contributions from some of the most experienced and respected practitioners in the industry--including Michael Feathers, Pete Goodliffe, Diomidis Spinellis, Cay Horstmann, Verity Stob, and many more--this book contains practical knowledge and principles that you can apply to all kinds of projects.A few of the 97 things you should know:"Code in the Language of the Domain" by Dan North"Write Tests for People" by Gerard Meszaros"Convenience Is Not an -ility" by Gregor Hohpe"Know Your IDE" by Heinz Kabutz"A Message to the Future" by Linda Rising"The Boy Scout Rule" by Robert C. Martin (Uncle Bob)"Beware the Share" by Udi Dahan

Book Information

Paperback: 258 pages

Publisher: O'Reilly Media; 1 edition (February 22, 2010)

Language: English

ISBN-10: 0596809484

ISBN-13: 978-0596809485

Product Dimensions: 6 x 0.5 x 9 inches

Shipping Weight: 10.6 ounces (View shipping rates and policies)

Average Customer Review: 3.9 out of 5 starsÂ Â See all reviewsÂ (41 customer reviews)

Best Sellers Rank: #709,740 in Books (See Top 100 in Books) #107 inÂ Books > Computers & Technology > Programming > Languages & Tools > Ruby #172 inÂ Books > Textbooks > Computer Science > Algorithms #253 inÂ Books > Textbooks > Computer Science > Object-Oriented Software Design

Customer Reviews

I have read programming books for years. There was a time when I could write a "Hello World" program in each of seven or eight languages. That time has passed, mainly because I haven't been intimately involved in any specific software project for many years. Still, I have this habit of reading programming books and enjoying them, perhaps in the hope or expectation that one of these days I'll find myself with a project in front of me, time to work on it, and motivation to learn a new language or tool to make the project's vision a reality. Well, here's the newest book of programming

foundational tips that I have read.97 Things Every Programmer Should Know is a collection of short, two page essays, each by an experienced programmer. The book is a collection of tips and tricks for writing code that works, that is maintainable both by the author and by others, and that will best fit the situation. While the book doesn't measure up to some of my all time favorites in the genre like The Art of Unix Programming or The Pragmatic Programmer, it wasn't meant to. This is not an in depth guide to underlying philosophies of coding practices and standards, but quick hit and run articles that would be easy to fully grasp and absorb in short five minute bursts, such as during work or study breaks (which is how I read the book).Some of the topics included in this book will seem obvious such as "Don't Ignore That Error" and "Comment Only What The Code Cannot Say," and some tips are going to serve only as reminders to best practices that are sometimes ignored (to our own peril) like "Check Your Code First Before Looking To Blame Others" and "Make Interfaces Easy To Use Correctly And Difficult To Use Incorrectly," there are some real gems in the book that aren't so obvious like one author's instruction to "Read the Humanities" because they are a great tool to help programmers learn to effectively interact with people and not just software and the advice that says "Don't Just Learn the Language, Understand Its Culture" so that you will write effectively and idiomatically within each language, rather than writing the same thing using different words.I can't say that this is a must-have book for experienced programmers, but anyone at the novice to intermediate levels would certainly benefit from what the book contains. I've enjoyed reading it.

If you're just entering the programming world, this collection of 2-page essays might be a useful resource. But if you've been reading programmer blogs for a while, or you've worked on a couple of projects, then there's little of value here. Very few of the essayists choose to tell stories; instead, they say things like "Remember that humans always make mistakes," "Read other people's code" and "Always leave the campground cleaner than you found it."Speaking of which, where is the code? A book on programming without code is like a day without sunshine! To give one example: The second essay, "Apply Functional Programming Principles" by Edward Garson, assures you that you'll write cleaner, clearer code after working with a functional programming language, but his assurances feel awfully airy without any examples. Maybe this is inevitable in a book that's language-agnostic. Books likeÂ Code CompleteÂ andÂ Clean CodeÂ are hopping with code samples (in C++ and Java, respectively); as a result, they do a far better job of engaging the reader and making abstract concepts stick.A notable exception is "Code in the Language of the Domain" by Dan North, which uses code to illustrate a concept and uses it well. You might want to read that one, but you don't need to buy the book to do so: All of the essays in this book are Creative

Commons-licensed and can be read on the book’s official website.Here is why Joel Spolsky’s books are so good: He tells stories. He gives examples. He restrains himself from bombarding the reader with familiar aphorisms. You’re imbibing his experiences, not just listening to him ramble. If you haven’t already readÂ Joel on SoftwareÂ andÂ More Joel on Software, definitely do so. Also check out the deep interview collectionÂ Coders at Work. It’s the 98th thing every programmer should know.

97 things every programmer should know is a light easy read that is broad enough to appeal to anyone who works in code or on software projects in general.I found essays like "How to Implement Doing it Right vs Getting it Done" to be very helpful and wise. That essay included pratical advice that we were able to apply by changing our design for our in house bug tracking software to include a technical debt tracker. "Coding with Reason" included some decent maxims that I hope my programmers implement, and I will be checking for in future code reviews. It is for these excellent essays among others that the book is worth reading.As a software development manager who also gets involved in the business side of things I was amused at how occasionally at the contradiction that exist between the business world and the software development world. In the essay "The Professional Programmer" that emphasized among other things that programmers should not tolerate bug lists and take responsibility for training themselves (I agree). However, I know that often times programmers have little control over their time and I know that our fallen nature inclines people who self study (if they do it all) often times to study what they like rather than what is useful to the company. In my knowledge of Business management the opposite advice is given, that in order to keep a motivated workforce the employer needs to provide training and/or training opportunities. Essays pushing pair programming made a good argument for it, but excluded what practical ideas can be implemented if such a thing is not possible.Sometimes I did not always agree with all the essays nor did I think that certain maxims should be elevated to the level of dogmas. Where the book suffered was that some of the essays selected seemed to reiterate points that where already made in other essays.I would recommend this book and I will even be using it for our in house book club.

You Should Know) XSLT 2.0 Programmer's Reference (Programmer to Programmer) 13 Modern Artists Children Should Know (Children Should Know) Esquire Things a Man Should Know About Work and Sex (and Some Things in Between) 50 Things Every Young Gentleman Should Know Revised and Upated: What to Do, When to Do It, and Why (Gentlemanners) The 5 Things Every Teen Should Know About Sex 33 Things Every Girl Should Know: Stories, Songs, poems, and Smart Talk by 33 Extraordinary Women 100 Things Every Child Should Know Before Confirmation: A Guide for Parents and Youth Leaders Masters of Corporate Venture Capital: Collective Wisdom from 50 VCs Best Practices for Corporate Venturing How to Access Startup Innovation & How to Get Funded How to Use Graphic Design to Sell Things, Explain Things, Make Things Look Better, Make People Laugh, Make People Cry, and (Every Once in a While) Change the WorldÃ Â The Elements of Journalism, Revised and Updated 3rd Edition: What Newspeople Should Know and the Public Should Expect What Every 6th Grader Needs to Know: 10 Secrets to Connect Moms & Daughters (What Every Kid Needs to Know) (Volume 1) iOS Programming: Starter Guide: What Every Programmer Needs to Know About iOS Programming Foundations of Security: What Every Programmer Needs to Know (Expert's Voice) Professional Jini (Programmer to Programmer) Microsoft Win32 Programmer's Reference Library: Multimedia (Microsoft Windows Programmer's Reference Library) Microsoft Win32 Programmer's Reference: Introduction Platforms, and Index (Microsoft Windows Programmer's Reference Library)