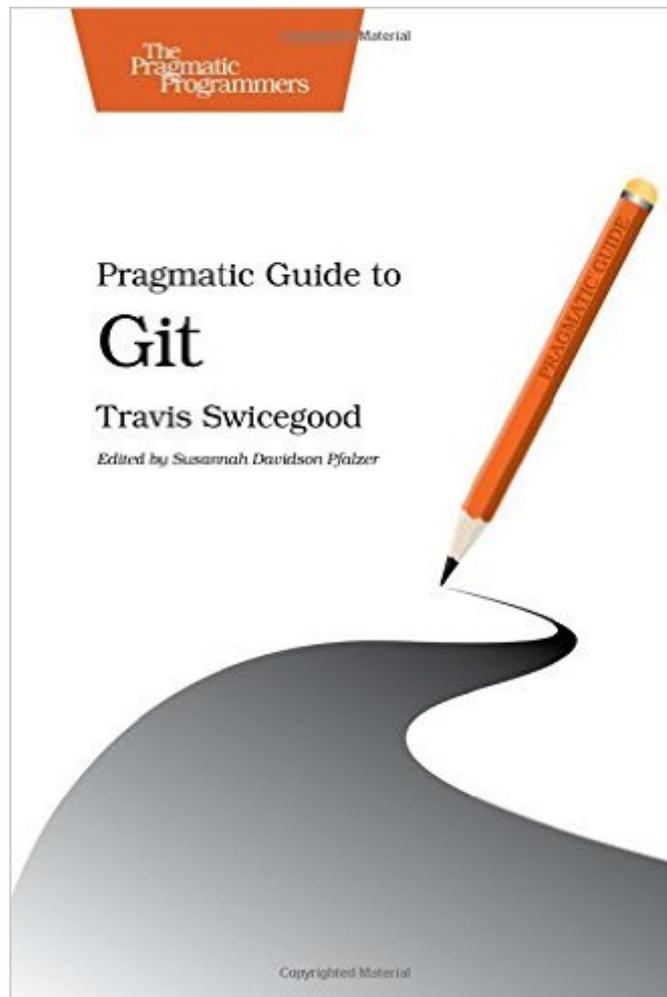# Pragmatic Guide To Git (Pragmatic Guides)

Synopsis

Need to learn how to wrap your head around Git, but don't need a lot of hand holding? Grab this book if you're new to Git, not to the world of programming. Git tasks displayed on two-page spreads provide all the context you need, without the extra fluff.Get up to speed on Git right now with Pragmatic Guide to Git. Task-oriented two-page spreads get you up and running with minimal fuss. Each left-hand page dives into the underlying implementation for each task. The right-hand page contains commands that focus on the task at hand, and cross references to other tasks that are related. You'll find what you need fast.Git is rapidly becoming the de-facto standard for the open source community. Its excellent merging capabilities, coupled with its speed and relative ease of use, make it an indispensable tool for any developer. New Git users will learn the basic tasks needed to work with Git every day, including working with remote repositories, dealing with branches and tags, exploring the history, and fixing problems when things go wrong. If you're already familiar with Git, this book will be your go-to reference for Git commands and best practices.You won't find a more practical approach to learning Git than Pragmatic Guide to Git.

Book Information

Series: Pragmatic Guides

Paperback: 160 pages

Publisher: Pragmatic Bookshelf; 1 edition (November 25, 2010)

Language: English

ISBN-10: 1934356727

ISBN-13: 978-1934356722

Product Dimensions: 6 x 0.5 x 9 inches

Shipping Weight: 10.4 ounces (View shipping rates and policies)

Average Customer Review: 4.1 out of 5 starsÂ Â See all reviewsÂ (25 customer reviews)

Best Sellers Rank: #241,389 in Books (See Top 100 in Books)   #61 inÂ Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Tools   #300 inÂ Books > Textbooks > Computer Science > Operating Systems   #634 inÂ Books > Computers & Technology > Operating Systems

Customer Reviews

I'm a relatively seasoned SVN user eager to make the switch to Git. Where Travis' earlier book on Git was aimed more at the general user (no real prior VCS experience assumed), this one dispensed with many of the introductory remarks about VCS in general and focused on getting up to

speed with GIt and understanding how it works.There is a good mix of "how to" information with explantations for why things work the way they do. For the beginners, I also appreciated the advice on best practices with Git. For example, even though you have a private repo and can work locally, Travis advises to not "work in a cave," that is, don't work locally for too long a period of time without sharing with your team. Good advice.The book is developed in a logical manner and each section builds solidly upon the previous sections. It covers 95% of what you need to know for day-to-day usage and gives you hints and links for more information on that remaining 5%.Despite reading it a few months ago, I find myself referring to it once or twice a week when I'm trying to remember a particular command. I love it when books serve double duty as both an introduction to a topic and a long-term reference. I highly recommend it.

Length: 5:32 Mins

As I've been using git for a couple of years now, I wasn't entirely sure if I was the target audience for this book. It turned out to be a great reference for me and I did pick up a few tricks that I've added to my toolbox. If you have a background with version control systems like Subversion, CVS, mercurial, etc., and are just moving to git, this book will help you get up and running quickly. I found the format very useful with narrative on the left pages and commands on the right pages. It worked well for me.In this short video, I spoke about all of that and more, as well as my one nit with the book. The additional links I mention in the video can be found at this URL:[...]

There is a whole mental model involved with Git. Then you have to learn all the commands you need to type. I think this book succeeds brilliantly with the second aspect, but might need a top up on the concepts.I watched two screencasts first to get the "Git way" into my head. The first was the PeepCode screencast on Git, and then "Mastering Git Basics" by Tom Preston-Werner (a co-founder of GitHub). After watching those, I felt I had a good grasp of the concept, but I didn't know the commands to do various things. That's where this book filled in the details on all the commands you'll need. I really enjoyed this book's two page layout. It has explanation on the left page, and example commands on the right page. Great stuff.That's a pretty good way to learn git in 3 evenings. Watch screencasts the first night, and then read this book the next two nights.

This book is great for quickly showing you how to use git. As others have said, it does a good job at laying out the basics, without dragging you through a bunch of detail on the inner workings of the

tool. For some people that might be nice. I have been using git for quite some time, and I got this book thinking it would help me deepen my understanding and introduce me to things I didn't know before (I had a choice of two books to pick from, and I only knew the title and table of contents...). While there were a few commands in this book that I hadn't seen, the bulk of it was things I have known and used for a long time, and it gives only a shallow explanation of most things. This is not a book for an experienced git user.The book is a handy reference for every day tasks when you're first learning. However, I wonder if a (free) online reference might be just as effective. I suppose that depends on your style. The layout is pretty handy, as has been mentioned in other reviews; it's nice to have a simple description of the task on one side of the page and the commands to do the task on the other side of the page. Some of the font decisions in the book baffle me, though.

This book is pretty much what the title says: a pragmatic guide to understanding and using Git. There is just enough background that you can understand what to do if you have used a VCS before. The 2 page format makes it an excellent reference, and it is also a good book to read straight through. You are likely to want another book to go deeper at some point, but this book will retain a place on your desk as a practical reference.

The book is aimed at developers who have used a source control system before, but want to learn Git and it meets this aim well.I particularly like the style of layout used when explaining the Git commands and in what situation they are appropriate. Having an explanation on the left-hand page and the actual commands on the right-hand page is a nice approach.I read the book in one sitting, which says a lot about how engaging it is. But I would image that most people would read the 1st few chapters all the way though (Introduction, Getting Started and Working with Git) and then dip into the other chapters when needed.I would've preferred to have seen a more in-depth comparison of regular v. distributed Version Control Systems (VCS) and more info about how you work differently in Git. As a distributed VCS is fundamentally different from a regular one, people with experience in SVN (or similar) might need more help understanding the difference in the process. But this is only a minor quibble.I would definitely recommend it to someone who was looking to move from a traditional VCS to Git. Whilst it doesn't cover every aspect of Git, it successfully covers the 95% it sets out to. However I would say that it's not that useful to someone completely new to source control, but to be fair it doesn't claim to be targeting this audience.

Pragmatic Guide to Git (Pragmatic Guides) Better Git It in Your Soul: An Interpretive Biography of

Charles Mingus Git Essentials Version Control with Git: Powerful tools and techniques for collaborative software development Git in Practice Client Teaching Guides For Home Health Care (Gorman, Client Teaching Guides for Home Health Guides) Programming Ruby 1.9 & 2.0: The Pragmatic Programmers' Guide (The Facets of Ruby) Web Design for Developers: A Programmer's Guide to Design Tools and Techniques (Pragmatic Programmers) 3D Game Programming for Kids: Create Interactive Worlds with JavaScript (Pragmatic Programmers) Certified Programming with Dependent Types: A Pragmatic Introduction to the Coq Proof Assistant (MIT Press) Programming Elixir 1.2: Functional |> Concurrent |> Pragmatic |> Fun Programming Clojure (Pragmatic Programmers) Seven Concurrency Models in Seven Weeks: When Threads Unravel (The Pragmatic Programmers) Managing Systems in Transition: A Pragmatic View of Reengineering Methods and Tools The Agile Samurai: How Agile Masters Deliver Great Software (Pragmatic Programmers) Pragmatic Unit Testing in Java 8 with JUnit Programming Groovy: Dynamic Productivity for the Java Developer (Pragmatic Programmers) Pragmatic ADO.NET: Data Access for the Internet World Programming Groovy 2: Dynamic Productivity for the Java Developer (Pragmatic Programmers) The Cucumber Book: Behaviour-Driven Development for Testers and Developers (Pragmatic Programmers)