

Git

Annotation A guide to the popular version control system, this book walks Git users through the source control implications of how a team is structured, and how the software is delivered to clients. The book then covers not just how to use popular work flow strategies, such as GitFlow, but why, and under what circumstances, these strategies should be applied.

??????????,??????????,?????Git?????,?????????????????Git???,??????,???????????.???????????,?????Git GUI?????,??Git?????????????.?????GitHub,Bitbucket,GitLab???Git?????.

There's a change in the air. High-profile projects such as the Linux Kernel, Mozilla, Gnome, and Ruby on Rails are now using Distributed Version Control Systems (DVCS) instead of the old stand-bys of CVS or Subversion. Git is a modern, fast, DVCS. But understanding how it fits into your development can be a daunting task without an introduction to the new concepts. Whether you're just starting out as a professional programmer or are an old hand, this book will get you started using Git in this new distributed world. Whether you're making the switch from a traditional centralized version control system or are a new programmer just getting started, this book prepares you to start using Git in your everyday programming. Pragmatic Version Control Using Git starts with an overview of version control systems, and shows how being distributed enables you to work more efficiently in our increasingly mobile society. It then progresses through the basics necessary to get started using Git. You'll get a thorough overview of how to take advantage of Git. By the time you finish this book you'll have a firm grounding in how to use Git, both by yourself and as part of a team. Learn how to use how to use Git to protect all the pieces of your project Work collaboratively in a distributed environment Learn how to use Git's cheap branches to streamline your development Install and administer a Git server to share your repository Learn the fundamentals of version control through step-by-step tutorials that will teach you the ins-and-outs of Git. This book is your complete guide to how Git and GitHub work in a professional team environment. Divided into three parts – Version Control, Project Management and Teamwork – this book reveals what waits for you in the real world and how to resolve the problems you may run into. Once past the basics of Git, you'll see how to manage a software project, and finally how to utilize Git and GithHub to work effectively as a team. You'll examine how to plan, follow and execute a project with GitHub, and then apply those concepts to real-world situations. Workaround the pitfalls that most programmers fall into when driving a project with Git by using proven tactics to avoid them. You will also be taught the easiest and quickest ways to resolve merge conflicts. A lot of modern books on Git don't go into depth about non-technical topics. Beginning Git and GitHub will help you cover all the bases right at the start of your career. What You'll

Learn Review basic and advanced concepts of Git Apply Project Management skills using GitHub Solve conflicts or, ideally, avoid them altogether Use advanced concepts for a more boosted workflow Who This book Is For New developers, developers that have never worked in a team environment before, developers with basic knowledge of Git or GitHub, or anyone who works with text documents.

This pocket guide is the perfect on-the-job companion to Git, the distributed version control system. It provides a compact, readable introduction to Git for new users, as well as a reference to common commands and procedures for those of you with Git experience. Written for Git version 1.8.2, this handy task-oriented guide is organized around the basic version control functions you need, such as making commits, fixing mistakes, merging, and searching history. Examine the state of your project at earlier points in time Learn the basics of creating and making changes to a repository Create branches so many people can work on a project simultaneously Merge branches and reconcile the changes among them Clone an existing repository and share changes with push/pull commands Examine and change your repository's commit history Access remote repositories, using different network protocols Get recipes for accomplishing a variety of common tasks

????????Git????????????????????,????????????????????? ?????????????????????,?????Git????????????????????.

If you are a developer and you want to completely master Git without heavy theory, this is the book for you. A reasonable knowledge level and basic understanding of Git concepts will get you started with this book.

Learn everything you need to take full control of your workflow with Git with this curated Learning Path – dive in and transform the way you work About This Book Master all the basic concepts of Git to protect your code and make it easier to evolve Filled with practical recipes that will teach you how to use the most advanced features of the Git system Harness the full power of the Git version control system to customize Git behavior, manipulate history, integrate external tools, and explore platform shortcuts Who This Book Is For This learning path is for software developers who want to become proficient at using the Git version control system. A basic understanding of any version control system would be beneficial. What You Will Learn Transport your work to a remote repository in a centralized manner Experiment with your code without affecting functional code files Explore some tools used to migrate to Git from other versioning systems without losing your development history Understand the Git data model and how you can navigate the database with simple commands Debug with Git and use various techniques to find faulty commits Customize Git behavior system-wide, on a per-user, per-repository, and per-file basis Master administering and setting up Git repositories, configuring access, finding and recovering from repository errors, and performing repository maintenance Chose a workflow and configure/set up support for the chosen workflow In Detail Git is one of the most popular types of Distributed Version

Control System. Since its inception, it has attracted skilled developers due to its robust, powerful, and reliable features. Like most powerful tools, Git can be hard to approach for the newcomers. However, this learning path will help you overcome this fear and become adept at all the basic and advanced tasks in Git. This course starts with an introduction to version control systems before you delve deeply into the essentials of Git. This serves as a primer for the topics to follow such as branching and merging, creating and managing a GitHub personal repository, and fork and pull requests. You'll also learn how to migrate from SVN using Git tools or TortoiseGit and migrate from other VCSs, concluding with a collection of resources, links, and appendices. As you progress on to the next module, you will learn how you can automate the usual Git processes by utilizing the hook system built into Git. It also covers advanced repository management, including different options to rewrite the history of a Git repository before you discover how you can work offline with Git, how to track what is going on behind the scenes, and how to use the stash for different purposes. Moving forward, you will gain deeper insights into Git's architecture, its underlying concepts, behavior, and best practices. It gives a quick implementation example of using Git for a collaborative development of a sample project to establish the foundation knowledge of Git operational tasks and concepts. By exploring advanced Git practices, you will attain a deeper understanding of Git's behavior, allowing you to customize and extend existing recipes and write your own. This Learning Path is a blend of content, all packaged up keeping your journey in mind. It includes content from the following Packt products: *Git Essentials*, Ferdinando Santacroce *Git Version Control Cookbook*, Aske Olsson and Rasmus Voss *Mastering Git*, Jakub Narebski *Style and approach* Its step-by-step approach with useful information makes this course the ultimate guide to understanding and mastering Git. This course will show the road to mastery example by example, while also explaining the mental model of Git.

This books starts with an introduction into distributed version control systems. It continues to describe the basic Git terminology and how you can configure your Git tools. As the book advances you learn how to connect to remote repositories and how to use branches and tags. The book covers merging and rebasing changes and provides all the necessary tips and tricks to use Git. It also covers the usage of the popular online Git hosting platforms GitHub or Bitbucket and describes typical Git workflows which are considered as good practice.

Summary Learn Git in a Month of Lunches introduces the discipline of source code control using Git. Whether you're a newbie or a busy pro moving your source control to Git, you'll appreciate how this book concentrates on the components of Git you'll use every day. In easy-to-follow lessons designed to take an hour or less, you'll dig into Git's distributed collaboration model, along with core concepts like committing, branching, and merging. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Book Git is the source

can concentrate on your code. About the Book Git in Practice is a collection of battle-tested techniques designed to optimize the way you and your team manage development projects. After a brief overview of Git's core features, this practical guide moves quickly to high-value topics like history visualization, advanced branching and rewriting, optimized configuration, team workflows, submodules, and how to use GitHub pull requests. Written in an easy-to-follow Problem/Solution/Discussion format with numerous diagrams and examples, it skips the theory and gets right to the nitty-gritty tasks that will transform the way you work. Written for developers familiar with version control and ready for the good stuff in Git. What's Inside Team interaction strategies and techniques Replacing bad habits with good practices Juggling complex configurations Rewriting history and disaster recovery About the Author Mike McQuaid is a software engineer at GitHub. He's contributed to Qt and the Linux kernel, and he maintains the Git-based Homebrew project. Table of Contents PART 1 INTRODUCTION TO GIT Local Git Remote Git PART 2 GIT ESSENTIALS Filesystem interactions History visualization Advanced branching Rewriting history and disaster recovery PART 3 ADVANCED GIT Personalizing Git Vendors dependencies as submodules Working with Subversion GitHub pull requests Hosting a repository PART 4 GIT BEST PRACTICES Creating a clean history Merging vs. rebasing Recommended team workflows

??????,????????????????,????????;????????????Git?;?????????GitHub,Bitbucket?GitLab????????.
????Go????????????????Go????????????????Go????????????JavaScript?Ruby?Python?Java?C++????????????
????Go????????????????????????????????????
????Go??Go????????????
????????????Go??
??
??
????????Go????????????????reflection????????????????unsafe????????????????cgo????Go?C????
????????Go??http://gopl.io/????go get???????????? #???? GOTOP
Information Inc.

Practice your Git skills using exercises in your own environment. This book introduces concepts in an abstract visual way, and then enforces this learning through exercises - the Git katas. You will start with basic interactions such as commits and branches, and move on to both internals and collaborative workflows. Best practices are introduced and rehearsed throughout with hands-on exercises. Each topic is supplemented with interactive Git exercises that can be solved using any Git client – either the ubiquitous CLI or one of the many graphical clients so you'll learn in the environment you work in. The importance of Git is hard to overstate – it is used by 90% of software engineers worldwide and is the de facto standard for version control. Honing your Git skills is guaranteed to make you a better and more efficient developer. Building software can be stressful, but it doesn't need to be. Practical Git will give you the Git skills you need, and help keep your Git skills sharp. Add it to your library today. What You'll Learn Use Git through scripted exercises and the Git katas Understand Git's graph model Troubleshoot common and rare scenarios you may face Select and apply the right Git tool for the task Maintain and collaborate on Git repositories Tweak Git to gain the most from this powerful tool Who This Book Is For Anyone who is currently using Git in a copy-paste fashion. It will take you from using Git to knowing Git.

????Git?????????????????10?,????????????????????Git?????,?????,????Git????????????,?????????Git,?????????GitHub.
Version Control with Git takes you step-by-step through ways to track, merge, and manage software projects, using this highly

flexible, open source version control system. Git permits virtually an infinite variety of methods for development and collaboration. Created by Linus Torvalds to manage development of the Linux kernel, it's become the principal tool for distributed version control. But Git's flexibility also means that some users don't understand how to use it to their best advantage. Version Control with Git offers tutorials on the most effective ways to use it, as well as friendly yet rigorous advice to help you navigate Git's many functions. With this book, you will: Learn how to use Git in several real-world development environments Gain insight into Git's common-use cases, initial tasks, and basic functions Understand how to use Git for both centralized and distributed version control Use Git to manage patches, diffs, merges, and conflicts Acquire advanced techniques such as rebasing, hooks, and ways to handle submodules (subprojects) Learn how to use Git with Subversion Git has earned the respect of developers around the world. Find out how you can benefit from this amazing tool with Version Control with Git.

This practical guide contains a wide variety of recipes, taking you through all the topics you need to know about to fully utilize the most advanced features of the Git system. If you are a software developer or a build and release engineer who uses Git in your daily work and want to take your Git knowledge to the next level, then this book is for you. To understand and follow the recipes included in this book, basic knowledge of Git command-line code is mandatory.

Dive and explore into the latest addons of the latest Git. About This Book Master all the basic concepts of Git to protect your code and make it easier to evolve Use Git proficiently, and learn how to resolve day-by-day challenges easily This step-by-step guide is packed with examples to help you learn and work with Git's internals Who This Book Is For If you are a software developer with little or no experience of versioning systems, or you are familiar with other centralized versioning systems, then this book is for you. If you have experience in server and system management and need to broaden your use of Git from a DevOps perspective, this book contains everything you need. What You Will Learn Master Git fundamentals Be able to "visualize," even with the help of a valid GUI tool Write principal commands in a shell Figure out the right strategy to run change your daily work with few or no annoyances Explore the tools used to migrate to Git from the Subversion versioning system without losing your development history Plan new projects and repositories with ease, using online services, or local network resources In Detail Since its inception, Git has attracted skilled developers due to its robust, powerful, and reliable features. Its incredibly fast branching ability transformed a piece of code from a niche tool for Linux Kernel developers into a mainstream distributed versioning system. Like most powerful tools, Git can be hard to approach since it has a lot of commands, subcommands, and options that easily confuse newcomers. The 2nd edition of this very successful book will help you overcome this fear and become adept in all the basic tasks in Git. Building upon the success of the first book, we start with a brief step-by-step installation guide; after this, you'll delve into the essentials of Git. For those of you who have bought the first edition, this time we go into internals in far greater depth, talking less about theory and using much more practical examples. The book serves as a primer for topics to follow, such as branching and merging, creating and managing a GitHub personal repository, and fork and pull requests. You'll then learn the art of cherry-picking, taking only the commits you want, followed by Git blame. Finally, we'll see how to interoperate with a Subversion server,

covering the concepts and commands needed to convert an SVN repository into a Git repository. To conclude, this is a collection of resources, links, and appendices to satisfy even the most curious. Style and approach This short guide will help you understand the concepts and fundamentals of GIT in a step-by-step manner.

Provides information on using the version control system, covering such topics as branching, merging, and working with the repository.

Leverage the power of Git to smooth out the development cycle Professional Git takes a professional approach to learning this massively popular software development tool, and provides an up-to-date guide for new users. More than just a development manual, this book helps you get into the Git mindset—extensive discussion of corollaries to traditional systems as well as considerations unique to Git help you draw upon existing skills while looking out—and planning for—the differences. Connected labs and exercises are interspersed at key points to reinforce important concepts and deepen your understanding, and a focus on the practical goes beyond technical tutorials to help you integrate the Git model into your real-world workflow. Git greatly simplifies the software development cycle, enabling users to create, use, and switch between versions as easily as you switch between files. This book shows you how to harness that power and flexibility to streamline your development cycle. Understand the basic Git model and overall workflow Learn the Git versions of common source management concepts and commands Track changes, work with branches, and take advantage of Git's full functionality Avoid trip-ups and missteps common to new users Git works with the most popular software development tools and is used by almost all of the major technology companies. More than 40 percent of software developers use it as their primary source control tool, and that number continues to grow; the ability to work effectively with Git is rapidly approaching must-have status, and Professional Git is the comprehensive guide you need to get up to speed quickly.

Pro Git (Second Edition) is your fully-updated guide to Git and its usage in the modern world. Git has come a long way since it was first developed by Linus Torvalds for Linux kernel development. It has taken the open source world by storm since its inception in 2005, and this book teaches you how to use it like a pro. Effective and well-implemented version control is a necessity for successful web projects, whether large or small. With this book you'll learn how to master the world of distributed version workflow, use the distributed features of Git to the full, and extend Git to meet your every need. Written by Git pros Scott Chacon and Ben Straub, Pro Git (Second Edition) builds on the hugely successful first edition, and is now fully updated for Git version 2.0, as well as including an indispensable chapter on GitHub. It's the best book for all your Git needs.

A former singing cowboy himself, John I. White spent decades compiling information on cowboy and western songs and the artists, songwriters, and others attached to them. He also sought out and corresponded with a who's who of the genre, people like Badger Clark, Curley Fletcher, D. J. O'Malley, Romaine Lowdermilk, Will Barnes, Joseph Mills Hanson, and Owen Wister. In Git Along, Little Dogies, White draws on old friendships and his exhaustive files to bring readers the untold story of cowboy and western song. Wonderful anecdotes stand beside White's trademark attention to detail as he

