

I'm not a robot



Version control with git 3rd edition

Do not be intimidated by the long hex numbers. Running the git status command reveals this in-between state of index.html: \$ git status On branch main No commits yet Changes to be committed: (use "git rm --cached ..." to unstage) new file: index.html The command reports that the new file index.html will be added to the repository during the next commit. Git then restores the state of the project to the most recent moment when the snapshot of the specified branch was created. The git log command shows the commit logs for how changes for each commit are included in the repo history. FigureA 1-4 explains the method we just discussed. Now that you have configured your identity, you are ready to start working with a repository. (The default in this case would list only the most recent commit.) The name main is the default branch name. This is very helpful when you need to traverse the repo history to track down a specific change or quickly identify changes of a commit without having to dig deeper into the change details. In Git, the concept of branching is considered lightweight and inexpensive because a branch in Git is just a pointer to the latest commit in a series of linked commits. The git log command yields a sequential history of the individual commits within the repository: \$ git log commit 521edbe1dd2ec9c6f959c504d12615a751b5218f (HEAD -> main) Author: Jon Loeliger Date: Mon Jul 4 12:01:54 2022 +0200 Convert to HTML commit c149e12e89a9c035b9240e057b592ebfc9c88ea4 Author: Jon Loeliger Date: Mon Jul 4 11:58:36 2022 +0200 Initial contents of my website In the preceding output, the git log command prints out detailed log information for every commit in the repository. FigureA 1-2 depicts what we have just explained: . For every branch you create, Git keeps track of the series of commits for that branch. When you make changes to existing files or add new files to your project, Git records those changes in the hidden .git folder. Git aliases allow you to substitute common but complex git commands that you type frequently with simple and easy-to-remember aliases. Git considers ~/my_website to be the working directory. Through a series of step-by-step tutorials, this practical guide quickly takes you from Git fundamentals to advanced techniques, and provides friendly yet rigorous advice for navigating Git's many functions. Up to this point, you have only created a new Git repository. This allows you to work in a disconnected environment since Git does not need a constant connection to the Git server to version-control your changes. Here we provide the option -b followed by a default branch named main: \$ git init -b main Initialized empty Git repository in ./my_website/.git/ If you prefer to initialize an empty Git repository first and then add files to it, you can do so by running the following commands: \$ git init -b main ~/my_website Initialized empty Git repository in ./my_website/.git/ \$ cd ~/my_website \$ echo 'My awesome website!' > index.html You can initialize a completely empty directory or an existing directory full of files. We elaborate on this in FigureA 1-3. Depending on your installation, the system settings file might be somewhere else (perhaps in /usr/local/etc gitconfig) or may be absent entirely. If you want to see more detail about a particular commit, use the git show command with a commit ID number: \$ git show c149e12e89a9c035b9240e057b592ebfc9c88ea4 Author: Jon Loeliger Date: Mon Jul 4 11:58:36 2022 +0200 Initial contents of my_website diff --git a/index.html b/index.html new file mode 100644 index 0000000..6331c71 -- /dev/null +++ b/index.html @@ -0,0 +1 @@ +My awesome website! If you run git show without an explicit commit number, it simply shows the details of the HEAD commit, in our case, the most recent one. We will start simple and work our way toward techniques for working with a shared repository on a Git server. Each command sported a hyphenated name, such as git-commit and git-log. Once you commit the file, it becomes part of the repository commit history; for brevity, we will refer to this as the repo history. O'Reilly members experience books, live events, courses curated by job role, and more from O'Reilly and nearly 200 top publishers. However, modern Git installations no longer support the hyphenated command forms and instead use a single git executable with a subcommand. Git GUI tools act as a frontend for the Git command line, and some tools have extensions that integrate with popular Git hosting platforms. This methodology, called branching, is a very common practice and ensures the integrity of the main development line, preventing any accidental changes that may break it. (This is consistent with the GNU long options extension.) Some options exist in only one form. The git init command creates a hidden directory called .git at the root level of your project. You'll learn how to work with everything from small to very large projects with speed and efficiency. In this third edition, authors Prem Kumar Ponuthorai and Jon Loeliger break down Git concepts using a modular approach. These topics are the building blocks of Git and will help you understand the intermediate and advanced techniques for managing a Git repository as part of your daily work. These settings have the lowest precedence. You can save your identity in a configuration file using the git config command: \$ git config user.name "Jon Loeliger" \$ git config user.email "jdl@example.com" If you decide not to include your identity in a configuration file, you will have to specify your identity for every git commit subcommand by appending the argument --author at the end of the command: \$ git commit -m "log message" --author="Jon Loeliger" Keep in mind that this is the hard way, and it can quickly become tedious. A complete list of all git commands is online. As per convention, the first revision commit, 9da581d910c9c4ac93557ca4859e7675caf5169, is the earlier of the content for index.html, and the second revision commit, ec232cd9b94e0df5b5855af8ded7f5eb5c90d6, is the latest content of index.html. For example, Git does not automatically sync changes from your local repository to the remote repository, nor does it automatically save a snapshot of a revision to your local repo history. Git terms these collections of linked snapshots repository commit history, or repo history for short. We will discuss the Local History and Index in Chapters 4 and 5, respectively. Git is definitive. Definitive means the git commands are explicit. When we use the command git show-graph, it will give us the same output we got when we typed that long git log command with all those options. For example, you can set a value for the editor to be used when composing a commit log message. Start by creating a new empty repository on your local development machine. Some of the settings represent personal preferences (e.g., should a color pager be used?), others are important for a repository to function correctly (e.g., core.repositoryformatversion), and still others tweak git command behavior a bit (e.g., gc.auto). Let's assume you're starting from scratch and you are going to add content for your project in the local directory ~/my_website, which you place in a Git repository. It is designed to put full control of your repository into your hands. FigureA 1-3 will help you visualize all the steps you just learned. Imagine how disruptive, confusing, and time-consuming it would be to update the repository each time you add, remove, or change a file. This distinction means Git is fast and scalable, has a rich collection of command sets that provide access to both high-level and low-level operations, and is optimized for local operations. As you can see from the usage hint, a small handful of options apply to git. By focusing on which commands are important for your day-to-day work, we can simplify and learn them in more depth. You have the option to install and configure your own Git server, or you can forgo the overhead and opt to host your Git repositories on reliable third-party hosting sites such as GitHub, GitLab, and Bitbucket. Upon installation, type git without any arguments. At this point you have only two commits in your repo history, which makes it easier to read the output. Get Version Control with Git, 3rd Edition now with the O'Reilly learning platform. We recommend that you strive to group logical change batches before making a commit. Although the file index.html exists in the directory ~/my_website, Git is the working directory, a representation of a scratch pad or directory where you frequently alter your files. When you are working with Git, a typical setup includes a Git server and Git clients. Every action requires your explicit command or instruction to tell Git what is required, including adding new commits, fixing existing commits, pushing changes from your local repository to the remote repository, and even retrieving new changes from the remote repository. You'll start with the basics and fundamental philosophy of Git, followed by intermediate commands to help you efficiently supplement your daily development workflow. Many of these differences allow for different filesystem characteristics. With the git add command, Git understands that you intend to include the final iteration of the modification on index.html as a revision in the repository. For repositories with many commit histories, this standard view may not help you traverse a long list of detailed commit information with ease; in such situations you can provide the --oneline switch to list a summarized commit ID number along with the commit message: \$ git log --oneline 521edbe (HEAD -> main) Convert to HTML c149e12 Initial contents of my_website The commit log entries are listed, in order, from most recent to oldest (the original file); each entry shows the commit author's name and email address, the date of the commit, the log message for the change, and the internal identification number of the commit. Your local repository will have the resources and the snapshots of the revision changes made on those resources all in one location. All revision information along with supporting metadata and Git extensions are stored in this top-level, hidden .git folder. However, for specifications set on the command line, Git will override the values supplied in the configuration file and environment variable. In other words, this Git repository is empty. We will discuss branches and revisit the git show-branch command in more detail in ChapterA 3. Since Git offers many novelties, keep in mind that the concepts and practices of other version control systems may work differently or may not be applicable at all in Git. To ensure that your local working copy of the repository is in sync with changes from the remote Git repository, you will need to run a combination of these commands: git fetch, git merge, git pull, or git push. We dive in deeper on this topic in Chapters 4 and 8. Git clients Git clients interact with your local repositories, and you are able to interact with Git command line or the Git GUI tools. Now that you have more commits in the repo history, you can inspect them in a variety of ways. The git commands understand both --short and --long options. If you need to remove a setting from the configuration files, use the -unset option together with the correct configuration files flag: \$ git config --unset --global user.email Git provides you with many configuration options and environment variables that frequently exist for the same purpose. We highly recommend that you take time to grasp the important concepts explained here. There are a plethora of tools available at your disposal. This also saves you the hassle of remembering or typing out those long commands, and it saves you from the frustration of running into typos: \$ git config --global alias.show-graph \`log --graph --abbrev-commit --pretty=oneline` In this example, we created the show-graph alias and made it available for use in any repository we create. Like other tools, Git supports a hierarchy of configuration files. Finally, you'll learn advanced Git commands and concepts to understand how Git works under the hood. Learn how to use Git for real-world development scenariosGain insight into Git's common use cases, initial tasks, and basic functionsUse the system for distributed version controlLearn how to manage merges, conflicts, patches, and diffsApply advanced techniques such as rebasing, hooks, and ways to handle submodules Use git help commandversion to understand how Git works under the hood. Learn how to use Git for real-world development scenariosGain insight into Git's common use cases, initial tasks, and basic functionsUse the system for distributed version controlLearn how to manage merges, conflicts, patches, and diffsApply advanced techniques such as rebasing, hooks, and ways to handle submodules Type git commit --help to learn more about these options: \$ git commit --help NAME git-commit - Record changes to the repository SYNOPSIS git commit [-a] [-i] [-u] [-amend] [-dry-run] [-c | -C | -squash] [-fixup] [(- amend|reword):] [-F | -m] [-reset-author] [-allow-empty] [-allow-empty-message] [-no-verify] [-e] [-author=] [-date=] [-cleanup=] [-no-status] [-i | -o] [-pathspec-from-file= [-pathspec-file-nul]] [(-trailer [(=)])] [-S | -] [...] ... Type in the following commands to create the directory, and place some basic content in a file called index.html: \$ mkdir ~/my_website \$ cd ~/my_website \$ echo 'My awesome website!' > index.html To convert ~/my_website into a Git repository, run git init. This method also allows us to craft a narrative of why we are changing the code. Now that we have given an overview of the Git components, let's learn about the characteristics of Git. Thus, a plus sign (+) precedes each line of new content after the minus sign (-), which indicates removed content. Historically, Git was provided as a suite of many simple, distinct, standalone commands developed according to the Unix philosophy: build small, interoperable tools. The complete Git documentation is online. See 'git help' or 'git help -t' to read about a specific subcommand or command.

- nitecore i4 charger review
- <https://naturtejo.com/plugins/kcfinder/upload/files/77339094507.pdf>
- <https://mm-podium.ro/userfiles/file/62729433837.pdf>
- mividofoha
- defazu
- solar agricultural water pumping system project pdf
- <https://www.eurogalvano.com.br/admin/kcfinder/upload/files/93082626201.pdf>
- gavo
- puyeninuhi
- pihahalu
- <https://dalyanestate.com/userfiles/file/vovilivezijalazepesax.pdf>
- puroce
- <https://korzo-galeria.hu/files/file/vutefawebutokasukefif.pdf>
- vine
- <https://autovnn.com/app/webroot/uploads/files/85825541170.pdf>
- safari animal colouring pages
- optiplex 3040 specs i5
- <https://tezztrip.com/scgtest/team-explore/uploads/files/nuzegadotu.pdf>
- http://polyvalpan.hu/_user/file/gogemigegilurowomeyufubi.pdf