Continue

Git Tutorial: Getting Started with Version Control ---------- Git is a powerful tool for managing changes in your codebase. It allows you to save snapshots of your project and make experimental changes without risking your work. ### Prerequisites ---------- To follow this tutorial, you need to install Visual Studio Code. ### Learning Git Basics with GitHub Desktop ---------- For standard Git operations, we recommend using GitHub Desktop, an app that lets you interact with Git visually instead of through written commands. ### Setting up GitHub Desktop ---------- If this is your first time using GitHub Desktop, download it and open the app. Click Sign in to GitHub.com and authorize GitHub Desktop to access your account. Follow the prompts to connect your GitHub account. ### Creating a Local Repository ---------- A repository is a project folder that tracks changes and stores history. To create a local repository, click Create a New Repository on your Local Drive. Name the repository "learning-git" and select Initialize this repository with a README to create a blank README.md file automatically. ### Creating a Remote Repository ---------- Linking a remote repository makes it easier to collaborate on and back up your work. In GitHub Desktop, click Publish repository and then click Publish repository one more time to see your remote repository. ### Setting up Branches ---------- Branches are essentially copies of your project where you can test changes without risking the stability of your existing work. A new branch allows you to make experimental changes and revert back to a previous version if needed. ### See all videos → You can work on multiple features simultaneously by creating different branches in GitHub Desktop. Create a new branch named readme-updates and click Create Branch. To save progress, make a commit to take a snapshot of your project at that point in time. You've already made an initial commit when you initialized the project with a README. When working on a branch, click Open in Visual Studio Code, paste Markdown text into README.md, and save changes. In GitHub Desktop, type "Update README" in the commit message box and click Commit to readme-updates. When happy with changes, publish the branch to the remote repository and create a pull request. In the pull request window, change the title and write a brief description of your changes. Click Create pull request. To bring changes into the main branch, click Merge pull request. When working on a project, it's standard practice for someone else to review your pull request before merging. Git command line basics for GitHub Desktop users GitHub Desktop provides a command-line interface that addresses day-to-day Git needs. As you grow as a developer, you may need more control over Git operations or use complex commands. To access the command line, press Ctrl+` in GitHub Desktop. For Windows, install Git; for macOS and Linux, it's installed by default. Install the GitHub CLI to perform actions quickly from the command line. Authenticate with GitHub using the `gh auth login` command. Install GitHub Copilot, a powerful extension that helps you find and understand commands. Experiment with complex commands like `git blame`. Git is pre-installed on many Mac and Linux systems. To check if you have it installed, open your terminal app. For a Windows machine, use Git Bash or the Windows Command Prompt. Type "git version" in the terminal to see which version of Git you're running or if it's not recognized as an option. If git is unknown, follow further steps to install it. Installing GitHub Desktop will also install the latest version of Git if you don't already have it. With GitHub Desktop, get a command-line version of Git along with a user-friendly interface. You can use GitHub Desktop regardless of whether Git is installed on your system or not. For Windows users, go to the latest version of Git and download it from there. After installation, verify that git is running by typing "git version" in the Command Prompt or Git Bash. Note: A good source for downloading Git is from git-scm, as this usually includes the latest version of Git with a compatible command prompt and bash . Here's the lowdown on setting up Git on Linux - a fun fact: it was originally designed for versioning the Linux OS! Given its heritage, it's no surprise that it integrates seamlessly with Linux. To get started, just use your distro's package manager. On systems that support apt, you can grab the latest version of Git by running `sudo apt-get update` followed by `sudo apt-get install git-all`. Verify the installation with a simple `git version`. The same process applies for dnf-based systems - just run `sudo dnf install git-all` and check the outcome with `git version`. For more on installing Git on specific Linux flavours, like Ubuntu or Fedora, head to git-scm's documentation. Or, if you prefer the DIY route, learn how to compile Git from source code here.

Git documentation pdf download. Documentation in git. Github documentation download. Git docs.