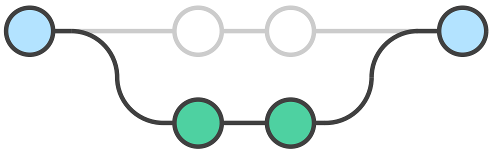
Advanced Git Tutorials

Atlassian's Git tutorials introduce the most common Git commands, and our Git Workflows modules discuss how these commands are typically used to facilitate collaboration. Alone, these are enough to get a development team up and running with Git. But, if you really want to leverage the full power of Git, you're ready to dive into our Advanced Git articles.

Each of these articles provide an in-depth discussion of an advanced feature of Git. Instead of presenting new commands and concepts, they refine your existing Git skills by explaining what's going on under the hood. Armed with this knowledge, you'll be able to use familiar Git commands more effectively. More importantly, you'll never be scared of breaking your Git repository because you'll understand why it broke and how to fix it.

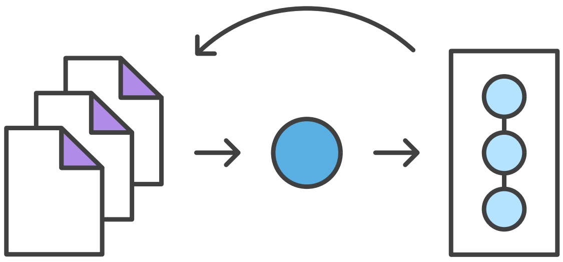
Tutorials



Git is all about working with divergent history. Its `git merge` and `git rebase` commands offer alternative ways to integrate commits from different branches, and both options come with their own advantages. In this article, we'll discuss how and when a basic `git merge` operation can be replaced with a rebase.

Learn more »

Resetting, Checking Out, and Reverting



The `git reset`, `git checkout`, and `git revert` commands are all similar in that they undo some type of change in your repository. But, they all affect different combinations of the working directory, staged snapshot, and commit history. This article clearly defines how these commands differ and when each of them should be used in the standard Git workflows.

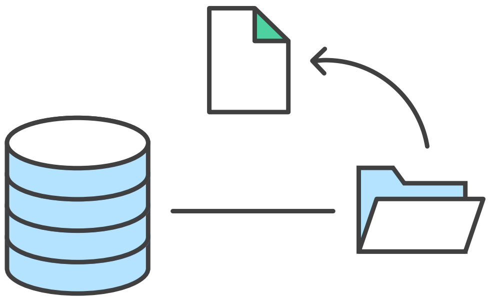
Tutorials

Advanced Git Log

The `git log` command is what makes your project history useful. Without it, you wouldn't be able to access any of your commits. But, if you're like most aspiring Git users, you've probably only scratched the surface of what's possible with `git log`. This article walks you through its advanced formatting and filtering options, giving you the power to extract all sorts of interesting information from your Git repository.

Learn more »

Git Hooks

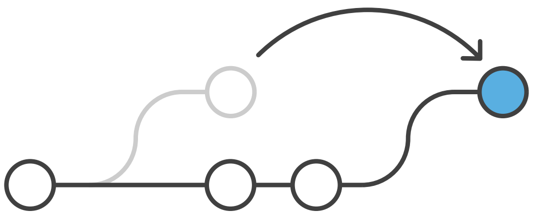


If you want to perform custom actions when a certain event takes place in a Git repository, hooks are your tool of choice. They let you normalize commit messages, automate testing suites, notify continuous integration

Tutorials

Learn more »

Refs and the Reflog



A **ref** is Git's internal way of referring to a commit. You're already familiar with many categories of refs, including commit hashes and branch names. But, there are many other types of refs, and virtually every Git command utilizes them in some form or another. You'll walk away from this article with an intimate knowledge of Git's inner workings.

Learn more »

Next up:

Tutorials

START NEXT TUTORIAL

Powered By

Tutorials

Enter Your Email For Git News