Essential Git Commands

A Comprehensive Reference for Version Control

by: github.com/nathanaelsanilo

Command	Action
git init	Initializes a new Git repository in the current directory, setting up version control.
git clone [url]	Creates a local copy of a remote Git repository, including its full history.
git add [file]	Stages a specific file's changes for the next commit.
git add .	Stages all changes (new, modified, deleted files) in the current directory for the next commit.
git commit -m "Your message"	Records staged changes as a new commit with a descriptive message.
git status	Displays the current state of the working directory and staging area.
git diff	Shows differences between various states (e.g., working directory vs. staging).
git log	Displays the chronological history of commits.
git branch	Lists all local branches in the repository.
git branch [branch-name]	Creates a new branch at the current commit.
git checkout [branch-name]	Switches the working directory to the specified branch.
git checkout -b [new-branch-name]	Creates a new branch and immediately switches to it.
git merge [branch-name]	Integrates changes from the specified branch into the current branch.
git pull	Fetches and merges the latest changes from the remote repository.
git push	Uploads local commits to the remote repository.
git remote -v	Lists configured remote repositories and their URLs.
git remote add origin [url]	Establishes a connection to a new remote repository named "origin."
git rm [file]	Deletes a file from the working directory and stages its removal.
git mv [old-name] [new-name]	Renames or moves a file and stages the change.
git reset [file]	Unstages a file, moving its changes back to the working directory.
git reset --hard [commit-hash]	Resets the branch and working directory to a specific commit, discarding subsequent changes.
git reset --soft [commit-hash]	Resets the branch to a specific commit, preserving subsequent changes in the staging area.
git revert [commit-hash]	Creates a new commit that undoes the changes of a previous commit.
git stash	Temporarily saves uncommitted changes, cleaning the working directory.
git stash pop	Applies the most recent stashed changes and removes them from the stash list.
git tag [tag-name]	Creates a lightweight tag pointing to a specific commit.
git tag -a [tag-name] -m "Message"	Creates an annotated tag with a message, pointing to a specific commit.
git fetch	Downloads new data from a remote repository without merging.
git config --global user.name "Your Name"	Sets the global Git username for commits.
git config --global user.email "your_email@example.com"	Sets the global Git email address for commits.
git config --list	Displays all Git configuration settings.
git rebase [branch-name]	Rewrites commit history by moving or combining commits to a new base.
git cherry-pick [commit-hash]	Applies the changes introduced by an existing commit from another branch onto the current branch.
git reflog	Shows a log of all actions performed in the repository, including those that rewrite history.
git submodule add [url] [path]	Adds a Git repository as a submodule of the current repository.
git bisect	Helps find the commit that introduced a bug by performing a binary search on the commit history.