Git Commands with Explanations

Setup and Configuration

git config --global user.name "Your Name" - Sets your name globally for all repositories.

git config --global user.email "you@example.com" - Sets your email globally for all repositories.

git config --list - Verifies the current Git configuration settings.

Initialize a Repository

git init - Creates a new, empty Git repository in the current directory.

Cloning a Repository

git clone <repository-url> - Copies an existing remote repository to your local machine.

Check Repository Status

git status - Checks the current state of your repository.

Add Files to Staging

git add <file-name> - Adds a specific file to the staging area.

git add . - Adds all changes in the current directory to the staging area.

Commit Changes

git commit -m "Your commit message" - Records the staged changes with a descriptive message.

git commit --amend - Modifies the most recent commit.

Branching

git branch - Lists all branches in the repository.

git branch <branch-name> - Creates a new branch.

git checkout <branch-name> - Switches to a specific branch.

git checkout -b <branch-name> - Creates and switches to a new branch in one step.

Merging Branches

git merge <branch-name> - Combines changes from the specified branch into the current branch.

Working with Remote Repositories

git remote -v - Lists remote repositories linked to your local repository.

git remote add origin <repository-url> - Links a remote repository to your local repository.

git push -u origin <branch-name> - Pushes changes to a remote branch and sets it as default.

git pull - Fetches and merges changes from the remote repository.

Viewing Commit History

git log - Displays the commit history with details.

git log --oneline - Shows a compact view of the commit history.

Stashing Changes

git stash - Temporarily saves uncommitted changes.

git stash pop - Restores stashed changes and removes them from the stash list.

Undoing Changes

git checkout -- <file-name> - Discards changes in the working directory for a specific file.

git reset HEAD <file-name> - Removes a file from the staging area without deleting changes.

git revert <commit-hash> - Creates a new commit that undoes changes from a specific commit.

Deleting Files or Branches

git rm <file-name> - Removes a file from the repository and working directory.

git branch -d <branch-name> - Deletes a local branch that has already been merged.

git branch -D <branch-name> - Force deletes a branch regardless of merge status.

Viewing Differences

git diff - Shows changes between the working directory and the staging area.

git diff <branch1> <branch2> - Compares differences between two branches.

Tagging

git tag <tag-name> - Creates a lightweight tag for a specific commit.

git tag -a <tag-name> -m "Tag message" - Creates an annotated tag with a message.

Working with Submodules

git submodule add <repository-url> <path> - Adds another repository as a submodule in your project.

Cleaning Up

git clean -f - Removes untracked files from the working directory.

Collaborative Work

git fetch - Retrieves changes from the remote repository but does not merge them.

git rebase <branch-name> - Moves your branch to the tip of the specified branch.

Advanced Commands

git cherry-pick <commit-hash> - Applies a specific commit from one branch to another.

git bisect - Helps find a commit that introduced a bug using binary search.