🧠 Master Git Guide for Beginners

By Md. Raju Mia

This guide is designed to help you understand and manage Git easily. It covers the most commonly used Git commands with descriptions, categorized into setup, working with commits, branching, remote repositories, and troubleshooting.

# 🔧 1. Git Setup

Sets your Git username globally.

git config --global user.name "Your Name"

Sets your Git email globally.

git config --global user.email "your.email@example.com"

Displays current Git configuration.

git config --list

Initializes a new Git repository in your current folder.

git init

# 📂 2. Basic Workflow Commands

Shows the current state of the working directory and staging area.

git status

Stages all files in the directory for the next commit.

git add .

Stages a specific file.

git add <filename>

Commits the staged files with a message.

git commit -m "commit message"

Shows commit history.

git log

Shows a compact commit history.

git log --oneline

Shows changes between working directory and staged changes.

git diff

# 🚀 3. Working with Remote Repositories

Adds a remote repository.

git remote add origin <repo\_url>

Lists all remotes.

git remote -v

Pushes local code to remote main branch and sets upstream.

git push -u origin main

Pulls latest changes from remote main branch.

git pull origin main

Clones a remote repository into your local machine.

git clone <repo\_url>

# 🌿 4. Branching and Merging

Lists all branches.

git branch

Creates a new branch.

git branch <branch\_name>

Switches to the specified branch.

git checkout <branch\_name>

Creates and switches to a new branch.

git checkout -b <branch\_name>

Merges the given branch into the current branch.

git merge <branch\_name>

Deletes a local branch.

git branch -d <branch\_name>

# 🔄 5. Undoing Changes

Restores the file to the last committed state.

git restore <file>

Unstages a staged file.

git restore --staged <file>

Undo the last commit but keep changes staged.

git reset --soft HEAD~1

Completely remove the last commit and changes.

git reset --hard HEAD~1

# 🧰 6. Advanced Commands

Temporarily saves changes that are not ready to commit.

git stash

Restores the most recently stashed changes.

git stash apply

Re-applies commits on top of another base tip (usually main).

git rebase main

Applies a single commit from another branch.

git cherry-pick <commit\_hash>

# 🔍 7. Inspecting Repositories

Displays information about a specific commit.

git show <commit\_hash>

Shows differences between branches.

git diff <branch1> <branch2>

Shows who made each line change in a file.

git blame <file>

# 🆘 8. Troubleshooting

Use this when your push is rejected because of conflicts.

git pull --rebase origin main

Force push if history has diverged (use with caution).

git push --force

Removes untracked files and folders.

git clean -fd

# 📁 9. .gitignore

Ignore log files, build folders, IDE settings, etc.

\*.log  
\*.class  
node\_modules/  
.idea/  
build/

Create a .gitignore file in your directory.

touch .gitignore