Complete Git Guide (Beginner Friendly)

This document explains Git concepts slowly and clearly, focusing on real situations. Read it in order. Do not try to memorize commands—understand when and why to use them.

1. git init

Creates a new Git repository. You run this once when starting a project.

- git init

2. git status

Shows the current state of files: modified, staged, or committed.

- git status

3. git add

Moves file changes from working directory to staging area.

- git add file.java
- git add .

4. git commit

Creates a permanent snapshot of staged changes.

- git commit -m "message"

5. git log

Shows commit history in a readable format.

- git log --oneline

6. git branch

Lists all branches and shows the current branch.

- git branch

7. git checkout

Switches branches or creates a new branch.

- git checkout main
- git checkout -b feat/task-delete

8. git pull

Fetches and merges latest code from GitHub into your local branch.

- git pull origin main

9. git push

Uploads your local commits to GitHub.

- git push origin branch-name

10. git restore

Undo local changes that are not committed.

- git restore file.java → discard file changes
- git restore --staged file.java → unstage file

11. git reset

Moves branch pointer backward. Use ONLY if commit is NOT pushed.

- git reset --soft HEAD~1 → undo commit, keep code staged
- git reset --hard HEAD~1 → undo commit and delete code

12. git revert

Safely undo a pushed commit by creating a new commit.

- git revert

Golden Rules

Use the correct command based on the situation.

- Edited but not committed → git restore
- Committed but not pushed → git reset
- Committed and pushed → git revert