🔧 1. **Git Configuration**

Command	Purpose
`git config --global user.name "Your Name"`	Set global username
`git config --global user.email "you@example.com"`	Set global email
`git config --global core.editor "code --wait"`	Set VSCode as default editor

📁 2. **Repository Setup**

Command	Purpose
`git init`	Initialize a new local Git repository
`git clone <repo_url>`	Clone a remote repository
`git remote add origin <repo_url>`	Add a remote named origin
`git remote -v`	View remote URLs

📄 3. **Basic Workflow**

Command	Purpose
`git status`	Show changes in working directory
`git add <file>`	Stage a specific file
`git add .`	Stage all changes
`git commit -m "message"`	Commit staged changes with a message
`git push origin <branch>`	Push commits to remote branch
`git pull origin <branch>`	Pull latest changes from remote
`git fetch`	Fetch remote changes without merging

🌿 4. Branching & Merging

Command	Purpose
`git branch`	List branches
`git branch <name>`	Create a new branch
`git checkout <branch>`	Switch to a branch
`git checkout -b <branch>`	Create and switch to a new branch

Command	Purpose
`git merge <branch>`	Merge specified branch into current
`git branch -d <branch>`	Delete a branch

🔄 5. **Undoing & Resetting**

Command	Purpose
`git reset <file>`	Unstage a file
`git checkout -- <file>`	Revert file to last committed version
`git revert <commit>`	Create a new commit that undoes changes
`git reset --hard <commit>`	Reset to a previous commit (DANGEROUS)

🕵️ 6. **Log & History**

Command	Purpose
`git log`	View commit history
`git log --oneline`	View simplified log

Command	Purpose
`git diff`	Show unstaged changes
`git diff --staged`	Show staged changes

🔒 7. Tagging

Command	Purpose
`git tag`	List tags
`git tag <tag_name>`	Create a new tag
`git push origin <tag_name>`	Push a tag to remote

🌐 8. Working with GitHub (or GitLab, Bitbucket)

Command	Purpose
`git push -u origin <branch>`	Push new branch and set upstream
`gh auth login`	Authenticate GitHub CLI (if using `gh`)
`git clone git@github.com:user/repo.git`	Clone via SSH

🚨 Notes

- Use `git help <command>` for details (e.g., `git help push`)
- Always pull (`git pull`) before pushing to avoid merge conflicts
- To **ignore files**, create a `.gitignore` file in the root