

# GIT COMMANDS

## 1. Git Configuration

Set up your Git identity and check your config:

- Set username: `git config --global user.name "Your Name"`
- Set email: `git config --global user.email "your@email.com"`
- Show current settings: `git config --list`

## 2. Working with Repositories

Start or copy a repository:

- Start a new repo: `git init`
- Clone existing repo: `git clone <URL>`

## 3. Managing Files

Track and commit file changes:

- Stage one file: `git add <file>`
- Stage all changes: `git add .`
- View status: `git status`
- Commit changes: `git commit -m "message"`
- Remove a file: `git rm <file>`

## 4. Branching

Create, switch, and merge branches:

- List all branches: `git branch`
- Create new branch: `git branch <name>`
- Switch branch: `git checkout <name>`
- Create + switch: `git checkout -b <name>`
- Merge another branch: `git merge <branch>`
- Delete a branch: `git branch -d <name>`

## 5. Remote Repositories

Work with remote GitHub/GitLab repos:

- View remotes: `git remote -v`

- Add remote: `git remote add origin <URL>`
- Push branch and set upstream: `git push -u origin <branch>`
- Push changes: `git push`
- Pull updates: `git pull`

## 6. Logs & Diffs

Inspect commits and code changes:

- Show history: `git log`
- Short log: `git log --oneline`
- Show unstaged changes: `git diff`
- Show staged changes: `git diff --staged`

## 7. Stashing

Temporarily save work:

- Save changes: `git stash`
- Apply last stash: `git stash apply`
- See saved stashes: `git stash list`
- Remove a stash: `git stash drop`

## 8. Undoing Changes

Go back if things go wrong:

- Discard changes: `git checkout -- <file>`
- Unstage file: `git reset <file>`
- Reset everything to last commit: `git reset --hard`

## 9. Advanced Tools

### ◆ Submodules

Use other Git repos inside your repo:

- Add: `git submodule add <URL>`
- Update: `git submodule update --init --recursive`