



Essential Git Commands

Repository Setup

Command	Description
<code>git init</code>	Create a new Git repository in your folder
<code>git clone <repo-url></code>	Clone a remote repo to your local machine
<code>git remote add origin <url></code>	Link local repo to a remote GitHub repo
<code>git remote -v</code>	View the current remote repo(s)

Tracking and Staging Changes

Command	Description
<code>git status</code>	See modified, staged, and untracked files
<code>git add <file></code>	Stage a specific file for commit
<code>git add .</code>	Stage all modified and new files
<code>git restore --staged <file></code>	Unstage a file without deleting changes
<code>git diff</code>	Show changes between working directory and staging area
<code>git diff --staged</code>	Show changes that are staged (to be committed)

Committing and Saving History

Command	Description
<code>git commit -m "message"</code>	Commit staged changes with a message

Command	Description
<code>git commit -am "message"</code>	Stage and commit all tracked files (skip <code>git add</code>)
<code>git log</code>	View all commit history (latest to oldest)
<code>git log --oneline</code>	View concise commit history (one line per commit)
<code>git show <commit-hash></code>	Show details of a specific commit
<code>git blame <file></code>	Show who made each change to a file (line by line)

Working with Remotes (GitHub)

Command	Description
<code>git push</code>	Push commits to the remote repo
<code>git pull</code>	Fetch and merge remote changes
<code>git fetch</code>	Download changes but don't merge
<code>git push -u origin main</code>	Push and set upstream tracking for the branch

Branching and Merging

Command	Description
<code>git branch</code>	List branches
<code>git branch <branch-name></code>	Create a new branch
<code>git checkout <branch></code>	Switch to a branch

Command	Description
git checkout -b <branch>	Create and switch to a new branch
git merge <branch>	Merge another branch into the current one
git stash	Temporarily save changes without committing
git stash pop	Reapply the stashed changes

Useful Shell Commands for Files and Folders

File & Folder Management

Command	Description
touch <filename>	Create a new empty file
mkdir <foldername>	Create a new directory/folder
mv <old> <new>	Rename or move a file or folder
rm <file>	Delete a file
rm -r <folder>	Delete a folder and its contents
ls	List files and folders in current directory
cd <folder>	Move into a folder
pwd	Show current directory path
cat <filename>	Display file contents

Command	Description
<code>code .</code>	Open current folder in Visual Studio Code

Inspection & Troubleshooting

View Git History

- `git log` – Full commit history
- `git log --oneline` – Compact history
- `git log --stat` – Show what files changed in each commit
- `git show <commit-id>` – Show a specific commit's details

View Staging Area and Differences

- `git status` – See which files are staged, modified, or untracked
 - `git diff` – See unstaged file changes
 - `git diff --staged` – See staged file changes
 - `git restore --staged <file>` – Unstage a file
-

Advanced Git (Bonus)

Command	Description
<code>git reset <file></code>	Unstage a file (keep local changes)
<code>git reset --hard</code>	Undo all changes and reset to last commit
<code>git revert <commit></code>	Undo a specific commit (safely)
<code>git clean -fd</code>	Remove all untracked files and directories

Command

git cherry-pick <commit>

Description

Apply a commit from
another branch