**📝 GIT Operations**

**INITIAL SETUP (Oncer per new project)**

bash ● ● ●

git init

Initializes a Git repository in your folder  
🔹 Creates a hidden .git folder to track all changes.

🧠 Run this once per project after creating your folder.

bash ● ● ●

git remote add origin https://githun.com/username/myrepo.git

🔹 Connects your local project to a Github repo.

🧠 “origin” is the default name for the remote server.

bash ● ● ●

git branch -M main

🔹 Renames the current branch to main.

🧠 New Git standards use main instead of master.

**DAILY USAGE (Save your progress and Push online)**

bash ● ● ●

git status

🔹 Shows the current status of your flies – modified, staged, etc.

bash ● ● ●

git add .

bash *For Single File* ● ● ●

git add index.html

🔹 Renames the current branch to main.

🧠 New Git standards use main instead of master.

bash ● ● ●

git commit -m “Updated dashboard page”

git commit -m “Added toggle button for dark/light mode”

🔹 Saves a snapshot of your changes with a message explaining it.

bash ● ● ●

git commit -m “Updated dashboard page”

🔹 Pushes your code online to Github.

u- links your main branch to Github’s main branch.

📌 This is only needed once after wards you can just do:

bash ● ● ●

git push

bash ● ● ●

git pull origin main

🔹 Downloads any new updates from Github to your local folder.

Useful if you edited from another computer or someone else contributed.

bash ● ● ●

git pull

🔹 git pull is used to **update your local project** with any changes made on GitHub (remote).  
 If your local copy is **behind the remote**, pushing will fail — so you pull first, then push.

| **🧭Situation** | **Do You Need git pull?** |
| --- | --- |
| You worked **only locally** since last time | ❌ No |
| You made changes **directly on GitHub.com** (e.g. edited README.md) | ✅ Yes, run git pull first |
| Someone else contributed to the repo | ✅ Yes, run git pull first |
| You’re unsure if remote has updates | ✅ It's safe to pull before pushing |

**OTHER USEFULL COMMANDS**

bash ● ● ●

git log

🔹 Shows your commit history with messages and timestamps.

bash ● ● ●

git remote -v

🔹 Shows the Github URL your project is connected to.

bash ● ● ●

git remote set-url origin <new-url>

🔹 Changes the Github URL if you renamed your repo or changed your username.

bash ● ● ●

git clone <repo-url>

🔹 Copies a Github repo into your local computer.

**GIT WORK FLOW**

🔹 Create folder for your project.

🔹 Open terminal in folder or in Visual Studio Code

🔹 Run:

bash ● ● ●

git init

git add .

git commit -m "Initial commit"

git remote add origin <repo-url>

git branch -M main

git push -u origin main

🔹 Then Continue with:

bash ● ● ●

git add .

git commit -m “……”

git push

**GIT COMMON MISTAKES AND FIXES**

❌ Forgot to close quotes in commit message → stuck at >

**Problem:**

bash ● ● ●

git commit -m "Unfinished message

>

Fix:

bash ● ● ●

git commit -m "Unfinished message

>CTRL + C # Cancel and return to terminal

Then Try again:

bash ● ● ●

git commit -m "Finished message"

❌ Wrong commit message and want to fix it

bash ● ● ●

git commit --amend -m "New message"

❌ Added wrong file, want to unstage before committing

Fix:

bash ● ● ●

git reset filename.txt

To unstage everything:

bash ● ● ●

git reset filename.txt

❌ Want to discard **local** changes (not committed yet)

Fix:

bash ● ● ●

git reset filename.txt

To discard **all** local uncommitted changes:

bash ● ● ●

git reset filename.txt

⚠️ Be careful — this **cannot be undone**!

📌 **Helpful Git Info**

bash ● ● ●

git log # See commit history

git diff # See changes not staged

git diff --staged # See what you staged

git config --list # Check Git config (username, email, editor)

**🧰 Git Shortcuts**

| **Action** | **Command** |
| --- | --- |
| Stage specific file | git add index.html |
| Commit | git commit -m "message" |
| Push | git push |
| Pull | git pull |
| See status | git status |
| Set Git username | git config --global user.name "MJ" |
| Set Git email | git config --global user.email "your@email.com" |