# 🚀 Git & GitHub: Illustrated Beginner-Friendly Guide

Prepared by Rasel

## 📘 Introduction

Git is a powerful version control system, and GitHub is a platform to host Git repositories online.   
This guide is written in the simplest way to help you understand, practice, and master Git and GitHub operations.

## 🔤 Common Git Commands & Their Meanings

* git status

Shows current changes, untracked files, and the status of your working directory.

* git add .

Stages all files (new/modified/deleted) to be committed.

* git commit -m 'message'

Saves your staged changes with a short message describing what you did.

* git push origin main

Uploads your local committed changes to the 'main' branch on GitHub.

* git pull origin main

Fetches and merges the latest changes from GitHub into your local project.

* git clone <repo\_url>

Copies a GitHub repository to your local machine.

* git log

Shows history of commits in the current branch.

* git diff

Shows the difference between modified files and their last committed versions.

* git rm -r folder\_name

Removes a folder and stages it for deletion.

* git reset --hard

Restores your working directory to the last committed state (⚠️ discards changes).

## 📤 SCENARIO A: Push Local Changes to GitHub

You’ve made changes locally and want to upload them to GitHub.

### ✅ Steps:

Check file changes - `git status`

Stage all changes - `git add .`

Commit with a message - `git commit -m "What you did"`

Push to GitHub - `git push origin main`

💡 Example:  
You deleted a folder and added a file named `hello.html`.

git status  
git add .  
git commit -m "Deleted a folder and added hello.html"  
git push origin main

## 📥 SCENARIO B: Pull Changes from GitHub

Someone updated the GitHub repository, and you want to update your local version.

Pull changes - `git pull origin main`

Done! - `Your local copy is updated.`

## 🔁 Daily Sync Workflow

1. Pull from GitHub first to avoid conflicts:  
 git pull origin main  
  
2. Make your changes locally  
  
3. Stage changes:  
 git add .  
  
4. Commit your changes:  
 git commit -m "Describe changes"  
  
5. Push to GitHub:  
 git push origin main

## 🛠️ Other Useful Git Tasks

Create a new branch - `git checkout -b branch\_name`

Switch branch - `git checkout branch\_name`

View branches - `git branch`

Delete a branch - `git branch -d branch\_name`

See commit history - `git log`

Undo changes before commit - `git restore filename`

Undo staged file - `git reset filename`

Delete remote file/folder - `git rm -r folder\_name && git commit -m 'remove' && git push`

## 💡 Tips and Best Practices

* ✅ Always pull before you push.
* 📝 Write meaningful commit messages.
* 🔁 Commit frequently to keep track of your progress.
* 🧪 Test your code before pushing.
* 📂 Use `.gitignore` to skip unnecessary files.