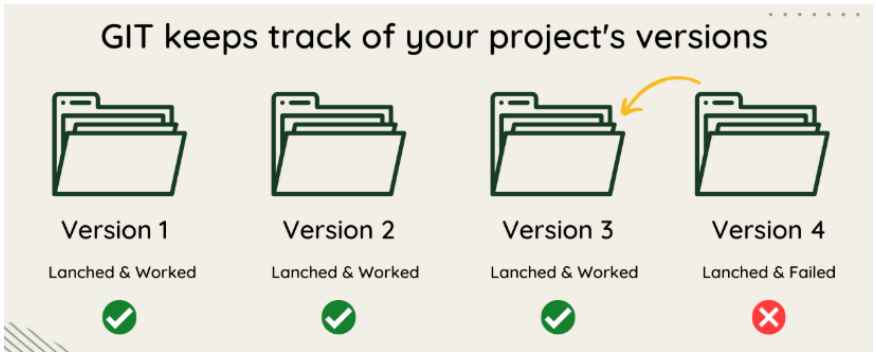
Git & GitHub Tutorial (with Commands)

What is Git?

Git is a distributed version control system to track changes in your code and collaborate with others.



How does Git work?

Git stores your files and their development history in a local repository. Whenever you save changes you have made, Git creates a commit. A commit is a snapshot of current files. These commits are linked with each other, forming a development history graph, as shown below. It allows us to revert back to the previous commit, compare changes, and view the progress of the development project.

What is GitHub?

GitHub is a cloud-based hosting service for Git repositories.

How does GitHub actually work?

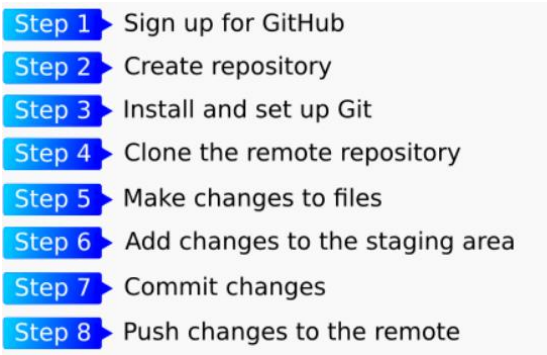
It uses Git to provide distributed version control and GitHub itself provides access control, bug tracking, software feature requests, task management, continuous integration, and wikis for every project. Headquartered in California, GitHub, Inc. has been a subsidiary of Microsoft since 2018.

Git and GitHub

A quick aside: git and GitHub are not the same thing. Git is an open-source, version control tool created in 2005 by developers working on the Linux operating system; GitHub is a company founded in 2008 that makes tools which integrate with git. You do not need GitHub to use git, but you cannot use GitHub without using git. There are many other alternatives to GitHub, such as

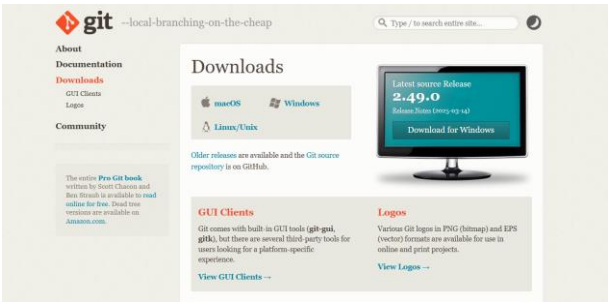
GitLab, BitBucket, and “host-your-own” solutions such as gogs and gittea. All of these are referred to in git-speak as “remotes”, and all are completely optional. You do not need to use a remote to use git, but it will make sharing your code with others easier.

How to use GitHub ?



Git Installation

Download & Install Git → https://git-scm.com/downloads



Git Commands

Verify Installation:

git –version

Git Basic Configuration

Configure your name & email (visible in commits):

git config --global user.name "Ashutosh Das"

git config --global user.email ashutoshdas0987@gmail.com

Check config:

git config –list

Git Command Flow (Cheat Sheet Style)

1. Initialize a Repository

git init

```
ashut@Ashutosh MINGW64 ~/OneDrive/Desktop/Wipro(GIT) (master)
$ git init
Reinitialized existing Git repository in C:/Users/ashut/OneDrive/Desktop/Wipro(GIT)/.git/
```

2. Clone Existing Repository

git clone https://github.com/AshutoshDas0987/Wipro.git

```
ashut@Ashutosh MINGW64 ~/OneDrive/Desktop/Wipro(GIT) (master
)
$ git clone https://github.com/AshutoshDas0987/Wipro.git
Cloning into 'Wipro'...
remote: Enumerating objects: 76, done.
remote: Counting objects: 100% (76/76), done.
remote: Compressing objects: 100% (70/70), done.
remote: Total 76 (delta 36), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (76/76), 789.26 KiB | 1.90 MiB/s, done.
Resolving deltas: 100% (36/36), done.
```

3. Check Status

git status

```
ashut@Ashutosh MINGW64 ~/OneDrive/Desktop/Wipro(GIT) (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        AshutoshDas.jpg
        MyaWebPage.html
        Wipro/
        clone/

nothing added to commit but untracked files present (use "git add" to track)
```

4. Add Files to Staging

git add <file-name> # Add single file

git add # Add all files

5. Commit Changes

git commit -m "Commit message"

6. Pull Changes from GitHub

git pull origin <branch-name>

7. Push Changes to GitHub

git push origin <branch-name>

8. View Commit History

git log

9. Create New Branch

git branch <branch-name>

10. Merge Branch

git merge <branch-name>

11. Delete Branch

git branch -d <branch-name>

12. View Remote

git remote -v

13. Add Remote Repo

git remote add origin https://github.com/AshutoshDas0987/Wipro.git

14. Remove Remote Repo

git remote remove origin

15. Rename Branch

git branch -m <new-branch-name>

16. Stash Changes (Temporary Save)

git stash

git stash pop # Apply back

Steps to Use GitHub with Git (Complete Guide)

1. Create a Repository on GitHub

Go to → https://github.com → Sign In → Click `New` → Fill details:

- Repository name
- Description (optional)
- Choose `Public` or `Private`
- Initialize with `README.md` (optional)
- Click → `Create Repository`

2. Link Your Local Project to GitHub Repo

Initialize Git Locally

```bash
git init
```

Add Remote Repo (Connect GitHub Repo)

```bash
git remote add origin https://github.com/<username>/<repo-name>.git
```

3. Basic Workflow Commands

Stage Files

```bash
git add .
```

Commit Changes

```bash
git commit -m "Your commit message"
```

Push to GitHub

```bash
git push -u origin main
```

4. Clone Any GitHub Repo

```bash
git clone https://github.com/<username>/<repo-name>.git
```

Example:

```bash
git clone https://github.com/ashutosh/myproject.git
```

---

5. Pull Changes from GitHub to Local

```bash
git pull origin main
```

6. Create & Push New Branch

Create New Branch

```bash
git checkout -b new-feature
```

Push Branch to GitHub

```bash
git push origin new-feature
```

---

7. Delete Branch from GitHub

```bash
git push origin --delete <branch-name>
```

8. Forking a Repository (Copy Repo to Your Account)

On GitHub:

- Click `Fork` → Select Your Account
- Clone forked repo:

```bash
git clone https://github.com/<your-username>/<repo-name>.git
```

9. Create Pull Request (PR)

1. Push code to your branch.
2. On GitHub → Click → `Pull Requests` → `New Pull Request`
3. Select:
    - base: main branch
    - compare: your branch
4. Submit → `Create Pull Request`

10. Handling Merge Conflicts

View conflicts during:

```bash
git pull origin main
```

Edit conflict files manually → Save → Then:

```bash
git add .
git commit -m "Resolved merge conflicts"
git push origin main
```

11. Remove Remote Repo Link (Optional)

```bash
git remote remove origin
```

---

12. View Existing Remotes

```bash
git remote -v
```

13. Check Commit History

```bash
git log --oneline --graph --all
```

---

14. Generate SSH Key (Recommended for GitHub)

```bash
ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
```

Add SSH Key to GitHub:

- GitHub → Settings → SSH and GPG Keys → New SSH Key → Paste public key.

15. Push Code Using SSH

Add SSH remote:

```bash
git remote set-url origin git@github.com:<username>/<repo-name>.git
```

Push normally:

```bash
git push origin main
```

Final GitHub Commands Cheat Sheet

Command	Purpose
git init	Initialize local repo
git clone URL	Clone remote repo
git add .	Stage all changes
git commit -m "msg"	Commit with message
git push origin branch	Push changes to GitHub
git pull origin branch	Pull latest code
git branch -b name	Create new branch
git push origin --delete branch	Delete remote branch
git remote -v	View remotes

Conclusion

Git and GitHub together form the backbone of modern software development and collaboration.

In this tutorial, we covered:

- Essential Git commands for local version control
- Step-by-step process to use GitHub for hosting repositories
- Real-world GitHub workflow including branching, pushing, pulling, and handling merge conflicts
- Useful tricks like SSH authentication, cloning, and pull requests

> *"Code without version control is like writing an essay without saving it."*

Bonus Tip:

> **Keep practicing on personal or open-source projects. Explore GitHub Issues, Actions, Wikis, and GitHub Pages to level up further!**