What is Git?

Git is a version control system.

It helps you track changes in your code, collaborate with others, and restore old versions if needed.

Think of it like a “time machine” for your code.

Example:

Imagine you’re writing a program:

- Day 1: It works fine.
- Day 2: You add new features and something breaks.

With Git, you can easily go back to Day 1 version — problem solved!

---

2. What is GitHub?

GitHub is a cloud platform (a website) that stores your Git repositories online.

It allows:

- Sharing your code with others
- Collaborating in teams
- Managing projects and issues
- Showing your portfolio to employers

Think of Git as your local tool, and GitHub as the online home for your projects.

---

3. Basic Terms You Should Know

Term	Meaning	Example
Repository (Repo)	A folder that stores your project and its Git history	A repo named my-website
Commit	A saved change (like a checkpoint)	“Added login feature”
Branch	A copy of your code to work on safely	“feature-login” branch
Merge	Combining code from one branch into another	Merging feature-login → main
Clone	Downloading a GitHub repo to your computer	git clone <URL>
Push	Sending your changes to GitHub	git push origin main
Pull	Getting the latest updates from GitHub	git pull origin main

---

4. How Git & GitHub Work Together

1. Create a repo (in GitHub or locally).
2. Make changes to your files.
3. Use Git commands to commit your changes.
4. Push those changes to GitHub.
5. Collaborate — others can pull, review, and merge your work.

---

5. Common Git Commands

Command	Description
git init	Create a new Git repository
git status	See what's changed
git add .	Stage all files for commit
git commit -m "Message"	Save your changes
git branch	Show available branches
git checkout <branch>	Switch to another branch
git push	Upload commits to GitHub
git pull	Download changes from GitHub

Step 1: Install Git on Your Computer

- For Windows:

1. Go to the official site https://git-scm.com/downloads
2. Download the Windows installer.
3. Run the setup — just keep clicking Next until finished (default settings are fine).
4. Once installed, open Command Prompt or Git Bash, and type:

git –version

If it shows something like git version 2.xx.x, you're good!

Step 2: Create a GitHub Account

1. Go to https://github.com
2. Click Sign up.
3. Enter your email, username, and password.
4. Once created, verify your email and sign in to GitHub.

Step 3: Configure Git (First-time setup)

Now, you'll tell Git your name and email — this info appears in your commits.

Open Git Bash or Command Prompt, and type:

```
git config --global user.name "Mallesh Kumar"
git config --global user.email youremail@example.com
```

Check if they're saved:

```
git config –list
```

Step 4: Create a New GitHub Repository

1. Go to GitHub.com
2. Click + → New repository (top-right corner).
3. Fill in:

- Repository name: e.g. MyFirstProject
- Description: “Learning Git and GitHub”
- Choose: Public
- Check: ✅ “Add a README file”

4. Click Create repository.

Now your repo is live at:

https://github.com/yourusername/MyFirstProject

Step 5: Clone (Download) the Repo to Your Computer

In GitHub, find the green Code button → Copy the HTTPS link.

Example:

https://github.com/yourusername/MyFirstProject.git

Now in your Git Bash, type:

```
cd Desktop
git clone https://github.com/yourusername/MyFirstProject.git
```

A new folder will appear on your desktop called MyFirstProject.

Step 6: Make Your First Change

1. Open the folder MyFirstProject.
2. Open the file README.md in Notepad or VS Code.
3. Add this line:

```
# My First GitHub Project 🚀
Learning Git and GitHub step by step.
```

Save the file.

Step 7: Commit and Push to GitHub

Go back to Git Bash (inside your project folder):

```
git status          # To see changed files
git add .           # Stage changes
git commit -m "Updated README with intro"   # Commit changes
git push origin main  # Upload to GitHub
```

Refresh your GitHub page — you’ll see your updated README file!

Step 8: Try Making a Branch (optional but useful)

```
git branch feature1       # Create a new branch
git checkout feature1     # Switch to that branch
```

Now edit something → save → commit → push:

git add.
git commit -m "Changed something in feature1 branch"
git push origin feature1
You'll see a message on GitHub suggesting to "Create Pull Request."
That's how team members review your changes before merging to the main branch.
Step 9: Congratulations! 🎉
You've just:

- Installed Git
- Created a GitHub account
- Cloned, edited, committed, and pushed your first change
- Created a branch