☑ Beginner's Guide to Git & GitHub for Collaboration

🛠 1. Install Git

Step 1. Download Git

- Go to: https://git-scm.com/downloads
- Choose your operating system (Windows, macOS, Linux) and install Git using default settings.

Step 2. Verify Installation

Open your **terminal** (Mac/Linux) or **Command Prompt** (Windows) and type:

```
git --version
```

You should see something like:
`git version 2.x.x` ✅

👤 2. Set Up Git (One-Time Setup)

Step 1. Set Your Name

```
git config --global user.name "Your Name"
```

Step 2. Set Your Email

```
git config --global user.email "you@example.com"
```

This info appears in your commits (like "who made this change").

🌐 3. Create a GitHub Account and Repository

Step 1. Create an Account

Go to https://github.com and sign up (free).

Step 2. Create a New Repository

- After logging in, click the ➕ icon (top right) > **New repository**
- Give your repo a name (e.g., `my-project`)
- Click **Create repository**

Leave "Initialize this repository with a README" **unchecked** for now (if you're going to push from your computer).

💻 4. Start a Project on Your Computer

Step 1. Open Terminal / Command Prompt

Step 2. Create a New Project Folder

```
mkdir my-project
cd my-project
```

Step 3. Initialize Git

```
git init
```

This sets up Git tracking in your folder.

🔁 5. Link Your Local Project to GitHub

Step 1. Add GitHub Repo as Remote

Copy the repo **HTTPS URL** from GitHub (looks like `https://github.com/yourname/my-project.git`).

Then run:

```
git remote add origin https://github.com/yourname/my-project.git
```

📄 6. Add & Push Your First File

Step 1. Create a File

You can use any editor or run:

```
echo "# My First Project" > README.md
```

Step 2. Add File to Git

```
git add .
```

Step 3. Commit the File

```
git commit -m "Initial commit"
```

Step 4. Push to GitHub

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

If your repo uses `master` instead of `main`, use that in place of `main`.

🤝 7. Collaborating on a Project

When working with others, follow this typical workflow:

✅ Step-by-Step Collaboration Workflow:

1. Clone the Project (Once)

If you're joining a project, get a copy:

```
git clone https://github.com/username/project-name.git
cd project-name
```

2. Create a New Branch (for Features)

```
git checkout -b feature/my-feature
```

3. Make Changes

Edit code or files as needed.

4. Stage and Commit Changes

```
git add .
git commit -m "Add my feature"
```

5. Push Your Branch to GitHub

```
git push origin feature/my-feature
```

6. Create a Pull Request on GitHub

Go to the repo on GitHub > Click **Compare & Pull Request** > Add description > Click **Create pull request**.

🔄 8. Keeping Your Code Up to Date

Before starting work or pushing changes, always pull the latest code:

```
git pull origin main
```

If you're on another branch:

```
git pull origin branch-name
```

🌿 9. Switching and Listing Branches

List Branches

```
git branch
```

Switch Branch

```
git checkout branch-name
```

❗ 10. Fixing Conflicts (Basic Tip)

Sometimes two people change the same part of a file. Git will ask you to fix the conflict:

1. Open the file and look for conflict markers (<<<<<<<, =======, >>>>>>>)

2. Choose what to keep.
3. Save the file, then:
4. `git add filename`
5. `git commit -m "Fix conflict"`

Tips for Success

- Always **pull before you push**
- Commit often with clear messages
- Use **branches** for new features or fixes
- Don't push directly to `main` unless necessary
- Communicate with your team on changes
