

Git Guide

Comprehensive Guide to Git: Setting Up, Managing Projects, and Collaborating with GitHub". This guide should help you set up your local environment, connect it to a remote repository on GitHub, and manage your project effectively using Git commands.

What is Git and why do we use it?

Git is a distributed version control system designed to track changes in source code during software development. It allows multiple developers to collaborate on projects efficiently by maintaining a history of changes, enabling easy branching and merging, and facilitating team coordination. Git's decentralized architecture enables developers to work offline and independently, while still being able to synchronize their work with others seamlessly. It provides robust features for managing codebases, including version tracking, branching strategies, code review, and collaboration tools. Git has become an industry-standard tool for software development due to its flexibility, speed, and reliability, empowering teams to build and maintain high-quality software products effectively

Software Installation:

Git Installation (Local):

- Download Git from the official website: git-scm.com.
- Run the installer and follow the installation instructions.

GitHub Account Creation (Remote):

- If you don't have one already, sign up for a GitHub account at github.com.

Local Setup:

Create Project Directory (Local):

- Create a project directory on your local machine where you want to store your project files. For example:

`C:/myproject/`

Navigate to Project Directory (Local):

- Open Command Prompt (Windows) or Terminal (macOS/Linux).
- Change directory to your project directory using the `cd` command:
`Cd C:/myproject/`

Initialize Git Repository (Local):

Run the following command to initialize a Git repository in your project directory:

`git init`

Create Project Files (Local):

- Create your project files within the project directory (`C:/myproject/`).
Ex: `myfile.txt`

Add Files to Git Repository (Local):

- Use the **`git add`** command to add your project files to the Git repository

`git add .`

**`dot (.)` is add all files in the folder/directory, if you want single file to add you can like :
`git add myfile.txt`**

Commit Changes (Local):

- Commit your changes with a descriptive message using the **`git commit`** command:

`git commit -m "Initial commit"`

Remote Setup (GitHub):

Create Repository (Remote):

- Log in to your GitHub account.
- Click on the "+" sign in the top-right corner and select "New repository".
- Enter a name for your repository (e.g., "myproject").
- Optionally, add a description, choose visibility settings, and initialize with a README file.
- Click "Create repository".

Get Repository URL (Remote):

- Copy the URL of your newly created GitHub repository.

Connect Local to Remote:

Link Local Repository to Remote (Local):

- Use the **git remote add** command to link your local repository to the remote GitHub repository:

```
git remote add origin <repository_URL>
```

Replace **<repository_URL>** with the URL of your GitHub repository.

Push Local Code to Remote (Local):

- Push your local code to the remote repository using the **git push** command:

```
git push -u origin master
```

Creating a New Branch (Optional):

Create a New Branch (Local):

- Use the **git checkout -b** command to create and switch to a new branch. For example:

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

Verify Branch Creation (Local):

- Check that the new branch was created using the **git branch** command:

```
git branch
```