

Git and GitHub:

Git: It is a version control system -> it allows multiple people to work on a project simultaneously and track changes without requiring a constant internet connection to a central server.

Why Use?

Collaboration -> enables multiple developers to work on same project simultaneously

Branching & Merging -> allows to create branches, new feature/ experiments without affecting the main codebase

Distributed Development -> each developer has a complete copy of the repository, including its history

Installation:

Official Website -> <https://git-scm.com/downloads>

<https://git-scm.com/downloads/win> -> for Windows

<https://git-scm.com/downloads/mac> -> for macOS

<https://git-scm.com/downloads/linux> -> for linux,

- ➔ Just have to download and keep clicking "NEXT" then you have to leave default setting.
- ➔ If u r in windows then I think u'll have 7-8 steps, where you have click on "NEXT", leave the default setting.
- ➔ Once it installation completed -> then u have to go terminal (any terminal like VS code, CP).
- ➔ Gitbash would have been installed with it -> git -version

Git Workflow:

🔗 **Initialize a Repository:** Start tracking an existing project or create a new one using git init.

- ➔ git init

🔗 **Add Files:** Stage files to be committed using git add.

- ➔ git add filename # or to add all changes

git add .

🔗 **Commit Changes:** Record changes to the repository with git commit.

- ➔ git commit -m "Describe your changes here"

🔗 **Push to Remote:** Send your commits to a remote repository with git push.

- ➔ git push origin branch-name

🔗 **Pull from Remote:** Update your local repository with changes from the remote with git pull.

- ➔ git pull origin branch-name

🔗 **Create and Switch Branches:** Use branches to isolate work, create a new branch, and switch to it.

- ➔ git branch branch-name # create a branch
- ➔ git checkout branch-name # switch to the branch

To start using Git, install it on your system, configure your name and email (required for commit tracking), and begin by initializing or cloning a repository. Here's how to set up your Git configuration:

- ➔ git config --global user.name "Your Name"
- ➔ git config --global user.email "youremail@example.com"

GitHub -> Web-based platform that hosts Git repositories, enabling developers to collaborate on projects, manage version control, and share code. Built on Git, GitHub offers additional tools and features designed to streamline collaboration, project management, and software development workflows.

Key Features of GitHub:

1. **Repository Hosting:** GitHub provides a space for storing, organizing, and managing repositories. Each repository can contain code, documentation, and additional resources like images and datasets.
2. **Version Control:** Built on Git, GitHub enables developers to track changes, review history, and collaborate more effectively. It supports branching, merging, and detailed commit history tracking.
3. **Collaboration Tools:** GitHub has built-in tools for collaboration, such as **Issues** for bug tracking and feature requests, **Pull Requests** for code reviews, and **Discussions** for community communication.
4. **Project Management:** GitHub offers **project boards**, **milestones**, and **labels** to help teams organize tasks, set deadlines, and prioritize work within a repository.
5. **GitHub Actions:** Automates workflows such as building, testing, and deploying code. GitHub Actions supports Continuous Integration/Continuous Deployment (CI/CD), allowing you to set up custom workflows directly within GitHub.
6. **Social Coding:** GitHub is a social platform for developers. Users can **star** repositories, **fork** projects, and follow other developers to see their contributions, fostering a community for open-source and collaborative projects.

GitHub Workflow:

GitHub's collaborative workflow enables developers to work on projects simultaneously, review each other's work, and integrate changes smoothly. Here's an overview:

1. **Fork a Repository:** To make changes to someone else's project, first fork it. Forking creates a personal copy of the repository under your GitHub account.
2. **Clone the Repository:** Clone the repository from GitHub to your local machine to work on the code.

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

3. **Create a Branch:** Before making changes, create a new branch for your work. This helps keep changes isolated and makes them easier to review.

`git checkout -b branch-name`

4. **Make Changes and Commit:** Edit files as needed and commit your changes.

`git add .
git commit -m "Description of changes"`

5. **Push Changes:** Push your branch to GitHub.

`git push origin branch-name`

6. **Create a Pull Request:** Once your changes are pushed to GitHub, create a pull request (PR). This lets others review, discuss, and approve or request changes to your code before merging it into the main branch.

Basic GitHub Commands (Using Git):

git clone: Copies a GitHub repository to your local machine.

git add: Stages changes for a commit.

git commit: Commits staged changes.

git push: Sends your commits to GitHub.

git pull: Retrieves updates from GitHub to your local machine.

Creating and Pushing a New Project to GitHub

1. **Create a Repository on GitHub:** Go to GitHub, create a new repository, and note the repository URL.

2. **Initialize Git Locally:**

git init

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

3. **Add Files, Commit, and Push:**

git add .

git commit -m "Initial commit"

git push -u origin main