Git and GitHub: Complete Guide

GIT: Version Control System

What is Git?

Git is a distributed version control system used to track changes in source code during software development.

Purpose of Git

- Tracks code changes over time.
- Allows multiple developers to work on the same project without conflicts.
- Helps revert to previous versions if needed.
- Branching and merging enable isolated development of features.

Key Concepts in Git

- Repository: A collection of files and folders tracked by Git.
- Commit: A snapshot of your code at a specific time.
- Branch: A separate line of development (default is main or master).
- Merge: Combining changes from one branch into another.
- Clone: Copying a remote repository to your local machine.
- Push: Sending local changes to a remote repository.
- Pull: Fetching and merging changes from a remote repo.
- Staging Area: Where changes are placed before committing.
- Working Directory: Where you actually edit the code.
- HEAD: A pointer to the current branch or commit.

Common Git Commands

- git init: Initialize a new Git repository.
- git clone <url>: Download a repository from GitHub.
- git status: Show modified files and staged files.
- git add <file>: Stage a file for commit.
- git commit -m "message": Save changes with a message.
- git log: View commit history.
- git branch: List, create, or delete branches.
- git checkout <branch>: Switch to a branch.
- git merge <branch>: Merge another branch into the current one.
- git push origin <branch>: Push commits to the remote.
- git pull origin <branch>: Pull updates from remote.

GITHUB: Git Repository Hosting Service

What is GitHub?

GitHub is a web-based platform for hosting and managing Git repositories. It allows collaboration, version tracking, and project management.

Features of GitHub

- Repositories: Hosted code with version history.
- Commits: Snapshots of code.
- Issues: Track bugs, tasks, enhancements.
- Pull Requests (PR): Propose changes and request merging.
- Forks: Create your copy of someone else's repository.
- Stars: Bookmark/favorite a repository.

- Actions (CI/CD): Automate tests, builds, and deployments.
- Projects: Manage workflows using Kanban-style boards.
- Wiki: Add project documentation.

Public vs Private Repos

- Public: Visible to everyone.
- Private: Only invited collaborators can see.

Typical GitHub Workflow

1. Create a repository on GitHub.
2. Clone it to your local machine: git clone <repo_url>
3. Make changes locally and commit them.
4. Push changes: git push
5. If working in teams:
   - Create a branch for your feature.
   - Push your branch.
   - Open a Pull Request for review and merging.
6. Merge PR after approval.

Git vs GitHub

- Git: Local version control.
- GitHub: Cloud-based hosting.

Example Git Workflow

```
git init
git remote add origin https://github.com/username/repo.git
git status
git add .
git commit -m "Initial commit"
git push -u origin main
```

Bonus Tools & Concepts

- .gitignore: File that tells Git which files to ignore.
- GitHub Desktop: GUI for Git/GitHub.
- GitKraken, SourceTree: Alternative Git GUI tools.
- Markdown (README.md): Used to write documentation in GitHub.
- GitHub Pages: Host static websites directly from a repository.
- SSH keys: Securely connect Git with GitHub.