



Git and Git Hub

What?

- **Git**: A tool used for version control of source code, which allows you to track changes, collaborate with others, and revert to previous versions if necessary.
- **GitHub**: A cloud-based platform where you can host Git repositories online, collaborate with others, and manage your projects, similar to how Google Drive stores files but specifically designed for code and version control.

Why?

- If something is wrong with what you have modified in your code, you can review your changes before committing or roll back to an old commit if needed
- Most likely, you cannot complete a project alone; you need to collaborate with others. Git and GitHub provide features to handle this situation.
- You will find yourself cannot live without GitHub during coding when you are familiar with it

Basic Components in Git

- **Repository** (like a folder):

  Think of the repository as a main folder where all your project files are stored, along with a history of every change made to those files.

- **Branch** (like a copy of the folder):

  A branch is like making a copy of the folder to work on something new (like adding a feature or fixing a bug). You can make changes in this copied version without affecting the main folder.

- **Commit** (like saving a snapshot of the folder):

  A commit is like taking a snapshot of the current state of your folder. Every time you make changes and commit, you're saving a new version of the folder with a clear history of what's changed.

Basic Commands in Git

- **`git init`**:
  Initializes a new Git repository in your project folder to start tracking changes.
- **`git remote add origin <your-remote-repo-url>`**:
  Links your local repository to a remote repository (GitHub) using the provided URL.
- **`git checkout -b main`**:
  Creates and switches to a new branch called `main` in your repository.
- **`git add .`**:
  Stages all the changes in your project folder, preparing them for a commit.
- **`git commit -m "your-commit-message"`**:
  Saves the staged changes with a message describing what you've done.
- **`git push -u origin main`**:
  Pushes the committed changes to the `main` branch of the remote repository and sets it as the default tracking branch.
