**What is Git?**

Git is a widely used version control system that allows multiple people to work on the same project simultaneously without overwriting each other's changes. It helps track changes to files, especially in software development, by maintaining a complete history of revisions, enabling collaboration, and providing tools to manage different versions of a project.

**What is GitHub?**

GitHub is a web-based platform that hosts **Git repositories**. It extends Git’s version control capabilities by offering additional collaboration tools, project management features, and a social network aspect for developers. GitHub allows users to store their code in remote repositories and collaborate with others on software projects.

1. How do we clone GitHub repositories in our local folder?

* Create a new folder or use an existing folder for clone Git repositories in your local system.
* Open the created folder or existing folder in VS Code.
* Open the new terminal in VS Code.
* Check the file path in the terminal and open folder path are the same.
* Go to the GitHub Account. Copy the repository link in the created repository(Repo).

A screenshot of a computer

Description automatically generated

* Come back to the VS Code terminal.
* Code: git clone “paste the link here”
* **Example:** git clone "https://github.com/Gowthamsasiku/demo.git"
* Change the terminal path to the local cloned repository path.
* Code: cd cloned repository path
* **Example:** cd Git-learning
* Now we can create or modify the files in the local system.

1. **We need to do “Three Steps” after creating or modifying files:**
2. Add/modify – git add “filename.txt” (or) git add .
3. Commit – git commit -m “commit message”
4. Push – git push origin main (or) git push
5. **Commands for Git:**
6. Git status - Shows the status of changes in the working directory (tracked, untracked, staged files, etc.).
7. git init - Initializes a new Git repository in the current directory.
8. git add filename.txt - Adds a specific file to the staging area.
9. git add . - Adds all modified and new files to the staging area.
10. git commit -m "Added new feature" - Commits staged changes with a descriptive message.
11. git log - Shows a log of previous commits.
12. git remote add origin <https://github.com/username/repo.git> - Links the local repository to a remote one (like on GitHub or GitLab).
13. git push origin master - Fetches changes from the remote repository and merges them with your local code.
14. git branch Branchname - Creates a new branch.
15. git checkout branchname - Switches to an existing branch.
16. git checkout -b Branchname - Creates and switches to a new branch in one step.
17. git merge Branchname - Merges changes from one branch into the current branch.
18. git branch -d Branchname - Deletes a local branch.
19. git reset filename.txt - Removes a file from the staging area but keeps the changes in the working directory.
20. git reset --hard - Resets the working directory to the last commit, removing all changes.
21. git diff - Shows differences between your working directory and the staging area.
22. Git pull - You can use git pull to sync your local repository with the latest changes from the remote repository before you start working.

A diagram of a workflow

Description automatically generated