1. The git status command displays the state of the working directory and the staging area. It lets you see which changes have been staged, which haven't, and which files aren't being tracked by Git. Status output does not show you any information regarding the committed project history

2. **Steps to delete a local Git branch**
    3. Open a Git BASH or a command prompt in the root of your Git repository.
    4. If necessary, use the git switch or checkout command to move off the branch you wish to delete.
    5. Issue the following command: git branch --delete <branchname>
6.

7. **Proper steps to add existing code to GitHub**
    8. Create a GitHub repository for the existing project.
    9. Copy the GitHub URL for the new repo to the clipboard.
    10. Perform a git init command in the root folder of the existing project.
    11. Add all of the existing project's files to the Git index and then commit.
12.

13. A clone is a copy of all the code on the master branch. It is an exact replica of the code on github. A branch is a slightly changed or modified section of code that meets different objectives. Forks are local instantiations that let you make changes to someone else's codebase.

    The difference between HEAD (current branch or last committed state on current branch), index (aka. staging area) and working tree (the state of files in checkout) is described in "The Three States" section of the "1.3 Git Basics" chapter of Pro Git book by Scott Chacon (Creative Commons licensed).

14. Git checkout works hand-in-hand with git branch. The git branch command can be used to create a new branch. When you want to start a new feature, you create a new branch off main using git branch new_branch . Once created you can then use git checkout new_branch to switch to that branch.?

15. Git is a version control system that allows developers to track changes in their code. GitHub is a web-based hosting service for git repositories. In simple terms, you can use git without Github, but you cannot use GitHub without Git

16. Version control helps teams solve these kinds of problems, tracking every individual change by each contributor and helping prevent concurrent work from conflicting. Changes made in one part of the software can be incompatibl

17. Git commit
18. Git branch
19. Git init
20. Git add
21. Git checkout
22. Git merge
23. Git status
24. Git config
25. Git pull
26.