1 What does the command 'git status' do?

The git status command is used in Git to display the current state of the working directory and the staging area. When you run git status, Git will show you which files are staged (added to the index) for commit, which files are modified but not yet staged, and which files are untracked (not yet known to Git).

2 git status is one of the most commonly used Git commands as it provides a quick overview of the current state of your repository, helping you to understand which files have been modified and which changes are ready to be committed

2 How to delete a Git local branch?

git branch -d <branch\_name>

git branch -D <branch\_name>

3 How can I add a project to Git that already exists?

git init

git add .

git commit -m "Initial commit"

git remote add origin <remote\_repository\_url>

git push -u origin master

4 What exactly is a Git fork? What are the distinctions between a fork, a branch, and a clone?

Fork:

A fork in Git refers to a copy of a repository that is created on a Git hosting service, such as GitHub, GitLab, or Bitbucket.

Forking a repository allows you to freely experiment with changes without affecting the original repository. You can make modifications, propose changes, and contribute to the project through pull requests.

Forks are typically used in open-source projects where collaboration and contributions from multiple developers are encouraged.

When you fork a repository, you create your own independent copy, including the entire history of the original repository.

Branch:

A branch in Git is a lightweight movable pointer to a commit. It allows you to work on different features, fixes, or experiments within the same repository.

Branches are used to isolate development work and prevent conflicts between different changes. Each branch can have its own set of commits and changes.

You can create a new branch from an existing branch (such as master or main) and merge it back when the work is completed.

Branches are typically used for short-lived changes or feature development within a single repository.

Clone:

Cloning in Git refers to the process of creating a copy of a repository, including all of its files, commit history, and branches, on your local machine.

When you clone a repository, you create a complete copy that you can work with locally, including the ability to make changes, commit them, and push them back to the remote repository.

Cloning is commonly used when you want to start working on an existing project, collaborate with others, or contribute to open-source projects.

Clones are exact replicas of the original repository, and they maintain a connection to the remote repository from which they were cloned.

5 What is the difference between HEAD, working tree, and index in GIT?

HEAD points to the current commit and represents the state of the currently checked out branch.

Working tree is the directory where you work on your files.

Index (staging area) is where changes are staged before committing, acting as a middle ground between the working tree and the repository

6 Which GIT command is used to change branches?

git checkout <branch\_name>

git checkout -b <new\_branch\_name>

7 What is the difference between GitHub and Git?

Git:

Git is a distributed version control system (DVCS) that allows developers to track changes in their codebase, collaborate with others, and manage different versions of their projects.

It is a command-line tool that is installed on your local machine.

Git provides commands for creating repositories, tracking changes, creating branches, merging changes, and more.

With Git, developers can work offline, commit changes locally, and synchronize their work with remote repositories when they have an internet connection.

GitHub:

GitHub is a web-based platform and hosting service for Git repositories.

It provides a centralized location for developers to store their Git repositories and collaborate with others.

GitHub offers additional features on top of Git, such as a web-based interface for managing repositories, issue tracking, pull requests, code review tools, project management features, and more.

GitHub allows developers to host public repositories for free, while also offering paid plans for private repositories and additional features.

GitHub is widely used for open-source projects, as well as for hosting private repositories within organizations.

8 What are some of the advantages of using the Variation Control System? Which programming language is used in Git?

Version Control Systems (VCS), like Git, offer several advantages for developers and teams working on software projects:

Version History: VCS allows you to maintain a complete history of changes made to your codebase, including who made the changes, when they were made, and what exactly was changed. This makes it easy to track the evolution of the project over time and revert to previous versions if needed.

Collaboration: VCS enables collaboration among team members by providing a centralized repository where everyone can contribute changes. It allows multiple developers to work on the same files simultaneously without interfering with each other's work.

Branching and Merging: VCS allows you to create branches to work on separate features or fixes independently. Branching helps isolate changes and prevents conflicts between different development efforts. Merging allows you to combine changes from one branch into another, enabling seamless integration of features.

9 List out some Git repository features.

Git repositories offer several features that facilitate version control and collaboration in software development:

1. **Version History**: Git repositories maintain a complete history of changes made to the codebase, including commit messages, timestamps, and the details of each change.
2. **Branching and Merging**: Git supports branching, allowing developers to create separate branches to work on different features or fixes independently. Changes from one branch can be merged into another branch when ready.
3. **Collaboration**: Git repositories provide a centralized location for team members to collaborate on a project. Multiple developers can work on the same codebase simultaneously, and their changes can be integrated seamlessly.
4. **Remote Repository Hosting**: Git repositories can be hosted on remote servers, such as GitHub, GitLab, or Bitbucket. Remote hosting services offer additional features like issue tracking, pull requests, code review tools, and project management capabilities.
5. **Code Review**: Git repositories support code review workflows, allowing team members to review changes before they are merged into the main codebase. Code review tools help identify issues, provide feedback, and ensure code quality.
6. **Issue Tracking**: Some Git hosting platforms provide built-in issue tracking systems for managing tasks, bugs, and feature requests. Issues can be linked to specific commits, branches, or pull requests, providing context for discussions and resolutions.
7. **Pull Requests**: Git repositories support pull request workflows, allowing developers to propose changes to the codebase and request reviews from team members. Pull requests provide a structured way to review, discuss, and integrate changes into the main codebase.
8. **Permissions and Access Control**: Git repositories offer granular permissions and access control settings, allowing administrators to control who can view, edit, and merge changes in the repository.
9. **Hooks and Automation**: Git repositories support pre-commit, post-commit, pre-push, and other hooks for automating tasks and enforcing custom workflows. Hooks can be used to run tests, validate code, and trigger notifications based on specific events.
10. **Documentation and Wiki**: Some Git hosting platforms provide built-in wiki and documentation features for documenting project guidelines, APIs, and best practices. Wikis can be edited collaboratively by team members and linked to specific repository files or branches.