

1. What is Git?

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

2. What is version control?

Version control is a system that helps manage changes to files over time, enabling multiple people to collaborate on a project while maintaining a history of changes.

3. What are the advantages of using Git?

Git enables collaboration, history tracking, branching and merging, offline work, and efficient handling of code changes.

4. Explain the difference between Git and GitHub.

Git is a version control system, while GitHub is a web-based platform for hosting Git repositories and collaborating on projects.

5. What is a repository in Git?

A repository is a directory that holds your project's codebase, along with its history and changes.

6. How do you initialize a new Git repository?

Use the command: `git init`

7. What is a commit in Git?

A commit is a snapshot of changes made to your repository at a specific point in time.

8. How do you add changes to the staging area in Git?

Use the command: `git add <file>`

9. How do you commit changes in Git?

Use the command: `git commit -m "Your commit message"`

10. What is a branch in Git?

- A branch is an independent line of development that allows you to isolate changes from the main codebase.

11. How do you create a new branch in Git?

- Use the command: `git checkout -b new-branch-name`

12. How do you switch between branches in Git?

- Use the command: `git checkout branch-name`

13. How can you view the commit history in Git?

- Use the command: `git log`

14. What is a remote in Git?

- A remote is a repository hosted on a server, often used to collaborate with others. The default remote is usually named "origin."

15. How do you clone a remote repository to your local machine?

- Use the command: `git clone <repository URL>`

16. What is a pull request in Git?

- A pull request is a method for submitting changes for review before they are merged into a main branch.

17. How do you merge changes from one branch into another in Git?

- Use the command: `git merge <branch-name>`

18. What is a merge conflict?

- A merge conflict occurs when Git cannot automatically reconcile differences between two branches being merged. Manual intervention is required to resolve the conflict.

19. How do you undo the last commit but keep changes in the working directory in Git?

- Use the command: `git reset --soft HEAD~1`

20. What is the purpose of the `.gitignore` file?

- The `.gitignore` file specifies files and directories that Git should ignore, such as temporary files and build artifacts.

21. How do you remove untracked files in Git?

- Use the command: `git clean -f`

22. What is rebasing in Git?

- Rebasing is the process of moving or combining a sequence of commits to a new base commit. It's used to create a linear commit history.

23. How do you rename a branch in Git?

- Use the command: `git branch -m new-branch-name`

24. How can you discard changes in a specific file in Git?

- Use the command: `git checkout -- <file>`

25. What is the difference between a fast-forward merge and a three-way merge in Git?

- A fast-forward merge occurs when the target branch's HEAD is a direct ancestor of the source branch. A three-way merge involves a common ancestor, source, and target branch.

26. How do you revert a commit in Git?

- Use the command: `git revert <commit-hash>`

27. What is the purpose of the HEAD pointer in Git?

- The HEAD pointer points to the latest commit in the current branch, indicating where new commits will be added.

28. How can you view the changes made in the last commit in Git?

- Use the command: `git show`

29. How do you update your local repository with changes from a remote repository?

- Use the command: `git pull origin <branch-name>`

30. What is a tag in Git?

- A tag is a reference to a specific commit, often used to mark important points in the commit history, such as releases.

31. How do you create a tag in Git?

- Use the command: `git tag <tag-name>`

32. How can you push a new tag to a remote repository?

- Use the command: `git push origin <tag-name>`

33. What is the difference between a soft reset, mixed reset, and hard reset in Git?

- A soft reset keeps changes in the working directory, a mixed reset unstages changes, and a hard reset discards changes.

34. How can you cherry-pick a commit in Git?

- Use the command: `git cherry-pick <commit-hash>`

35. What is the purpose of the .gitattributes file?

- The .gitattributes file specifies attributes for paths in the repository, often used for controlling line endings and merge strategies.

36. How do you stash changes in Git?

- Use the command: `git stash`

37. How do you apply stashed changes in Git?

- Use the command: `git stash apply`

38. What is the difference between `git fetch` and `git pull`?

- `git fetch` retrieves changes from a remote repository, but does not automatically merge them. `git pull` fetches changes and merges them into the current branch.

39. How do you amend the last commit message in Git?

- Use the command: `git commit --amend`

40. What is the purpose of the `git config` command?

- The `git config` command is used to configure various settings for your Git environment.

41. How do you configure your name and email for Git commits?

- Use the commands:

```
git config --global user.name "Your Name" git config --global user.email "you@example.com"
```

42. What is the difference between `git log` and `git reflog`?

- `git log` shows the commit history, while `git reflog` shows a history of all references, including HEAD movements and branch changes.

43. How do you unstage changes in Git?

- Use the command: `git restore --staged <file>`

44. How can you view the differences between two commits in Git?

- Use the command: `git diff <commit1> <commit2>`

45. What is the purpose of the `.gitkeep` file?

- The `.gitkeep` file is used to ensure that an otherwise empty directory is tracked by Git. It's often used to maintain project structure.

46. How do you create an empty commit in Git?

- Use the command: `git commit --allow-empty -m "Empty commit"`

47. What is Git rebase interactive mode?

- Interactive rebase allows you to modify, reorder, squash, or edit commits during the rebase process.

48. How do you list all remote branches in Git?

- Use the command: `git branch -r`

49. How do you delete a remote branch in Git?

- Use the command: `git push origin --delete <branch-name>`

50. How can you resolve a merge conflict in Git?

- Manually edit the conflicting files to resolve differences, then use `git add` to mark the conflicts as resolved and commit the changes.