

Git Interview Questions (Basic to Advanced & Scenario-Based)

Basic Git Questions

1. What is Git, and why is it used?
2. Explain the difference between Git and GitHub.
3. How do you initialize a new Git repository?
4. What is the purpose of the Git staging area?
5. Describe the process of committing changes in Git.
6. What does the command ``git status`` do?
7. How do you check the commit history in Git?
8. What is a branch in Git?
9. How do you create a new branch in Git?
10. Explain the difference between ``git merge`` and ``git rebase``.
11. What is a merge conflict, and how do you resolve it?
12. How do you delete a branch in Git?
13. What is the purpose of the ``.gitignore`` file?
14. How do you revert a commit that has already been pushed to a remote repository?
15. What is the difference between ``git pull`` and ``git fetch``?
16. How do you stash changes in Git?
17. What is the difference between ``git reset`` and ``git revert``?
18. How do you set up a remote repository in Git?
19. What is the difference between ``origin`` and ``upstream`` in Git?
20. How do you rename a branch in Git?

Intermediate Git Questions

1. Explain the concept of Git tags.
2. How do you create an annotated tag in Git?
3. What is the difference between a lightweight tag and an annotated tag?
4. How do you list all tags in a Git repository?
5. How do you delete a tag in Git?

Git Interview Questions (Basic to Advanced & Scenario-Based)

6. What is the purpose of the ``git cherry-pick`` command?
7. How do you use ``git bisect`` to find a bug?
8. What is the difference between ``git clone`` and ``git fork``?
9. How do you configure a Git repository to ignore file mode changes?
10. What is the significance of the ``HEAD`` in Git?
11. How do you squash commits in Git?
12. What is the difference between ``git stash pop`` and ``git stash apply``?
13. How do you apply a patch in Git?
14. What is the purpose of the ``git reflog`` command?
15. How do you change the author of a commit in Git?
16. What is the difference between ``git archive`` and ``git bundle``?
17. How do you handle large binary files in Git?
18. What is Git LFS, and when would you use it?
19. How do you sign commits in Git?
20. What is the purpose of the ``git submodule`` command?

Advanced Git Questions

1. Explain the internal structure of a Git repository.
2. How does Git handle data integrity?
3. What is the difference between ``git gc`` and ``git prune``?
4. How do you optimize a Git repository?
5. What are Git hooks, and how do you use them?
6. How do you implement a pre-commit hook in Git?
7. What is the purpose of the ``git fsck`` command?
8. How do you recover a lost commit in Git?
9. What is the difference between ``git blame`` and ``git annotate``?
10. How do you track the origin of a line in a file using Git?
11. What is the purpose of the ``git worktree`` command?

Git Interview Questions (Basic to Advanced & Scenario-Based)

12. How do you manage multiple worktrees in Git?
13. What is the difference between ``git diff`` and ``git difftool``?
14. How do you configure an external diff tool in Git?
15. What is the purpose of the ``git notes`` command?
16. How do you add and view notes in Git?
17. What is the difference between ``git log`` and ``git shortlog``?
18. How do you customize the output of ``git log``?
19. What are reflogs, and how are they useful?
20. How do you expire and clean up reflogs in Git?

Scenario-Based Git Questions

1. You're working on a feature branch and realize it was branched off the wrong base. How do you rebase it onto the correct base without losing your work?
2. After a pull request is merged, you realize that it introduced a bug. What steps do you take to fix it without disrupting other developers' work?
3. You and a teammate both edited the same file. You've committed your changes but can't push due to merge conflicts. What do you do next?
4. You accidentally deleted a local branch that hadn't been pushed yet. How would you recover it?
5. You need to rollback your ``main`` branch to a previous commit, but others have already pulled the latest changes. How do you handle this safely?
6. Your team follows a strict commit message convention. How do you ensure all contributors follow it automatically?
7. A developer made a force push that overwrote shared history. How do you mitigate and recover from the situation?
8. You need to cherry-pick multiple non-sequential commits from one branch to another. How do you ensure no conflicts or dependency issues occur?
9. You're asked to find the commit that introduced a performance degradation. How would you approach this using Git?
10. You've stashed your work multiple times. How do you inspect, apply, or drop a specific stash entry?

Git Interview Questions (Basic to Advanced & Scenario-Based)

11. Your team uses pull requests for code reviews. A teammate has many small, messy commits. What do you suggest before merging?
12. A teammate pushed a large file to the repo accidentally. How do you remove the file completely from history and reduce the repo size?
13. You're managing a monorepo with multiple teams working on different projects. How do you maintain clean Git workflows?
14. You want to review changes made by a teammate before a feature branch was merged. How do you do that post-merge?
15. You accidentally pushed to the `main` branch instead of your feature branch. What do you do to fix it?
16. A tag was mistakenly created on the wrong commit and pushed to the remote. How do you fix and retag properly?
17. Your company has a policy that requires code review before merging. How can Git hooks or automation enforce this?
18. You cloned a repo, but when running `git pull`, you get a 'not a git repository' error. How would you troubleshoot this?
19. You want to clean up your Git history before pushing your feature branch to the remote. What tools or commands would you use?
20. You're maintaining a forked repo and want to regularly sync it with the original upstream. How do you do that effectively?
21. You're reviewing a pull request that has unrelated changes mixed in. How do you ask the contributor to clean it up, and what Git steps might they take?
22. During a rebase, you run into multiple conflicts. What's your strategy to resolve them efficiently and continue the rebase?
23. You have multiple developers pushing directly to the main branch, causing instability. How would you enforce a Git workflow that prevents this?
24. Your CI/CD pipeline fails due to a missing configuration file after a merge. How can Git help you identify where the issue was introduced?
25. You've rebased a feature branch and now want to push it, but Git is rejecting it due to non-fast-forward updates. How do you proceed?
26. You discover a security secret was committed to Git. How do you remove it from all history and rotate the

Git Interview Questions (Basic to Advanced & Scenario-Based)

key safely?

27. A teammate wants to undo their last 3 commits but keep the changes. What Git command would you recommend?

28. You need to test a feature as it existed 3 weeks ago, but without knowing the exact commit hash. How do you retrieve that snapshot?

29. You suspect a feature branch has diverged too far from main and might be hard to merge. How do you verify and prepare for this?

30. You're mentoring a junior developer who accidentally committed sensitive files. How do you walk them through cleaning it up using Git?