TECHNOLOGY

Top Git Interview Questions and Answers You Must Prepare in 2022

> Articles > Technology > **Top Git Interview Questions and Answers You Must Prepare in 2022**

by Deepali LAST UPDATED: December 17, 2021, 12:44 IST

Git is an open-source project that tracks changes in source code during software development. It can handle both small and large projects with speed and efficiency. It is one of the best source code management with features like task management and bug tracking and is widely used by most industry experts nowadays. If you are planning to appear for a DevOps interview in 2022, then be prepared to answer a lot of ***Git interview question***s. To make you interview-ready, here are the top ***Git interview questions*** along with answers.

Top Git Interview Questions and Answers

Here is the list of frequently asked ***Git Interview Questions***:

Q1. What is Git?

Ans. Git is an open-source project distributed version control system (DVCS). Many commercial projects rely on Git as every developer's code copy is also treated as a repository, which contains all changes done in the past. Below is the detailed description of DVCS:

- **Control System:** Git is known for its features like a content tracker, and it stores content.
- **Version Control System:** It helps developers to store code at the same time and Git modifies as and when more codes are added. The version control system helps in maintaining and

keeping records of all the changes. Further, it offers features like branches and merges.

- **Distributed Version Control System:** Git has a remote repository and a local repository, which are stored in servers and computers, respectively. This means that code is stored in both the central server and the developer's computer. Hence it is termed as a distributed version control system.

Explore popular courses on Naukri Learning

Popular Technology Course	Top Programming Courses
Popular Cloud Computing Courses	Popular Software Tools Courses

Q2. Why is it said that Git is designed keeping in mind performance, security, and flexibility?

Ans. Git was developed in 2015 by Linus Torvalds for Linux kernel development. But in the last decade, it gained a lot of interest, and today, due to its flexibility, nearly every development environment uses Git and runs Git command-line tools on every major operating system.

Below are the reasons why Git is highly recommended:

- **Performance:** Git has very powerful raw performance characteristics be it branching, merging, or comparing the past versions; it is robust and optimized. Git gives special attention to the content, and it uses a blend of delta encoding and compression. Further, it also clearly stores directory contents and metadata object versions.
- **Security:** Integrity is the topmost priority of Git. Its cryptography hashing algorithm named SHA1 safely stores all objects in the Git repository and maintains a true relationship between files and directories.
- **Flexibility:** From supporting nonlinear development workflow to adaptability with various systems and protocols, Git is exceptionally elastic. Git's amazing tracking system offers features like treating branching and tagging as first-class citizens. Its 'change history' also features stores operations affecting branches and tags.

Q3. What is the meaning of the commands – git status, git log, git diff, git revert <commit>, git reset <file>?

Ans.

Command	Meaning
git status	Gives a list of which files are staged, unstaged, and untracked
git log	Illustrates the entire commit history by using the default format
git diff	Displays the unstaged changes between index and working directory
git revert <commit>	Undoes all the changes made in <commit> and apply it to the current branch by creating a new commit
git reset <file>	Removes <file> from a staging area without overwriting any changes by keeping the working directory unchanged

Q4. What is Commit?

Ans. Git creates a commit as and when a developer saves any new work. Commit is a screenshot of all the files, and Git will use the previously used file if a file is not changed from one commit to another. One commit creates a chain to other commits and forms a development history graph. Unique cryptocurrency hash identifies commit in Git.

Q5. What are Branches in Git?

Ans. As multiple developers work parallel on a program, they create their own local repository, and this creates multiple changes in a single commit. However, in Git, branches manage various separations, and once the work in a branch is finished, it is merged with the master branch.

Q6. Are Git and GitHub the same thing?

Ans. Git and GitHub are connected as Github is a service to use Git, but they both have slight differences:

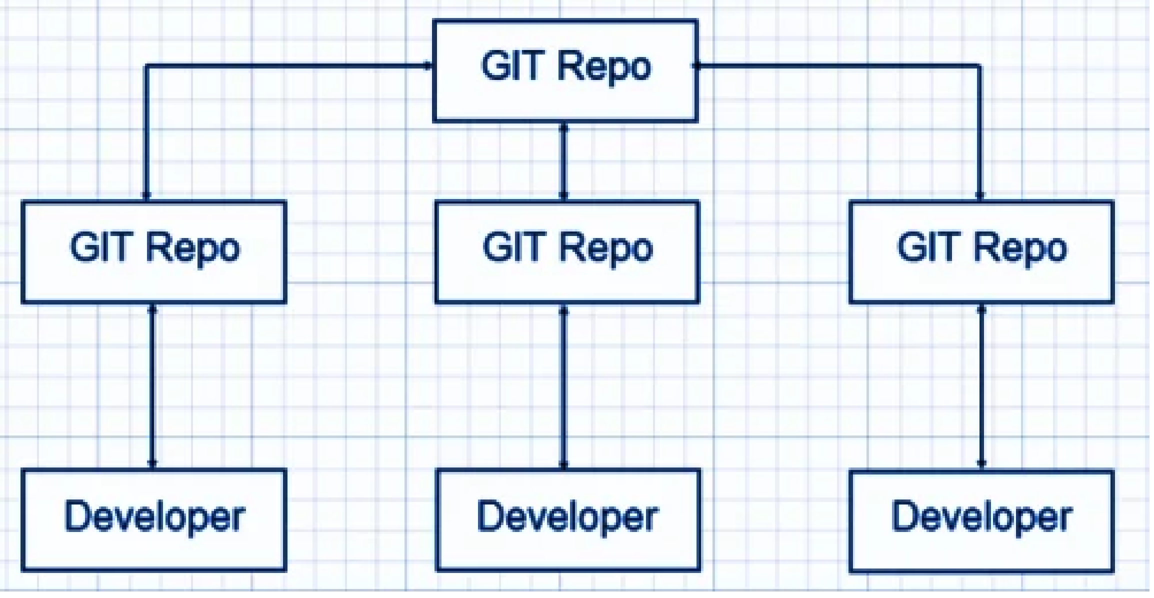
Git	GitHub
Git is a software tool to use a version control system	GitHub is a hosting service for git repositories
Tool for projects that want to collaboratively develop software	Service for projects that use Git for version control

Q7. Which is a better version control system – Git or SVN?

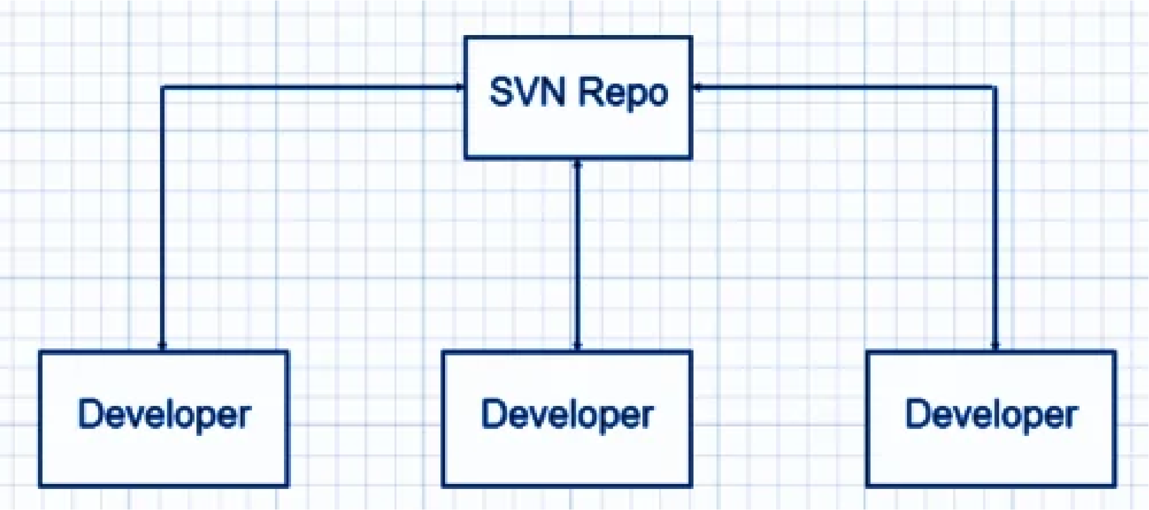
Ans. Both Git and SVN have their pros and cons, and below are the differences between Git and SVN:

Git	SVN
Decentralized and distributed version control tool	Centralized version control tool
Clones all repositories on the local system	Stores version history on the server-side repository
Supports offline commits	Supports online commits only
Swift push/pull operations	Slow push/pull operations
Automatically shares work to commit	Doesn't support automatic sharing

Git Workflow:



SVN Workflow



Q8. Name some Git repository hosting functions.

Ans. Below is the list:

- Github
- Gitlab

- Bitbucket
- SourceForge
- GitEnterprise

Q9. State the difference between “git pull” and “git fetch.”

Ans. This is an important Git interview question. “git pull” and “git fetch” are used for downloading new data from a remote repository.

“git fetch – It downloads new data from the repository but does not support integrating this data to working files. It offers a fresh view of things that happened in the remote repository.

“git pull” – This command is used to update the current HEAD data branch with all the changes that occurred in the remote repository. Thus, it downloads the data and integrates it with existing working files.

Q10. How do you edit or fix the last commit message in Git?

Ans. If you forget to add anything in the commit message or committed a typo error, you can rectify it by using –amend flag command.

$ git commit –amend -m “Sorry I missed an important update”

Note: –amend flag will only help in editing or fixing the last commit message.

Explore the 7 Most In Demand Tech Skills

Q11. How can you change any older commit message?

Ans. To change older commit the command is –

$ git rebase –interactive

Q12. How to deal with huge binary files in Git?

Ans. Handling large binary files is a significant problem in git, and to handle this problem “Large File Storage” extension works well for Git. Simply install LFS on your local computer, and after this, your large files will not be stored in the local repository. Instead, it will be stored in a dedicated LFS cache and store.

Also Read>> Trending Tech Skills in 2020: Cloud, Game development and DevOps

Q13. How to resolve and solve merge conflicts?

Ans. It is very easy to resolve merge conflicts as Git allows you to go back to the previous state. Just use a "git merge –abort" command, and you will be able to undo the merge and start the task again.

Q14. What do you mean by "git cherry-pick"?

Ans. If by mistake, you have committed a change into the wrong branch, you can use the "git cherry-pick" command. This command will allow you to apply commit from one branch to another branch.

```
$ git cherry-pick <commit id>
```

Q15. In which scenario you use the "git cherry-pick" command?

Ans. Git cherry-pick command can sometimes result in duplicate commits, and thus, it must be cautiously used. The below situations are apt if planning to use the git cherry-pick command:

- When you mistakenly make a commit in the wrong branch
- When you want to make changes that are proposed by other team members

Q16. Name some of the Git tools that you use.

Ans. This is a commonly asked Git interview question. Below is the list of most popular Git tools:

Git Tools	License Type
GitHub Desktop	MIT
GitKraken	Proprietary
SmartGit	Proprietary
Tower	Proprietary
Git Up	GNU GPL

Q17. What do you mean by bare repository in Git?

Ans. While initializing a new Git repository, – run git init function is used, and this directory becomes a 'Working Tree.' Also, Git creates its own .git directory (which is hidden) where it tracks all the changes and stores the commit objects.

However, a bare repository, also called bare repos, works without creating a 'Working Tree.' This bare repository is utilized as a remote repository. It helps share it

with all the users where the developers will clone it and locally make the required changes.

Q18. Why developers use Git Clone?

Ans. Developers prefer cloning as it is the simplest way to get a copy of the central repository. The 'git clone' command helps in generating a copy of the current Git repository.

Q19. Will you create an additional commit or amend an existing commit?

Ans. It is preferable to create an additional commit because:

- It might cause inappropriate changes
- A correct activity that was recently saved in a commit might ruin the express
- The chances are high that you miss including significant remains

Check out the list of Top Online IT Courses

Q20. Which branching strategies have you used?

Ans. To answer this Git interview question, you can share all the branching strategies that you have used. You can frame your answer as follows:

I have used –

- **Feature Branching:** keeps all of the changes for a particular feature inside of a branch.
- **Task Branching:** In this, each task is implemented on its own branch with the task key included in the branch name
- **Release Branching**: After the develop branch has acquired enough features for a release, that branch can be cloned to form a Release branch. This starts the next release cycle. Now you can not add new features after this point. Only bug fixes, documentation generation, etc. can be added to this branch. When it is ready to ship, the release gets merged into master and gets a version number.

Q21. Name some of the most popular Git repository hosting functions.

Ans. Below is the list of Git repository hosting functions:

- Pikacode
- Assembla
- Visual Studio Online
- GitHub
- GitEnterprise
- net
- Beanstalk
- CloudForge
- GitLab
- Planio
- Perforce
- Fog Creek Kiln

Q22. Why do you use Subgit?

Ans. The Subgit is a popular tool used for stress-free transferring of SVN to Git and it allows using various Git and sub-version features.

Q23. State the difference between HEAD, working tree, and index.

Ans. The working tree, also known as the working directory or workspace, is the directory tree of source files. Whereas, the index, which is also known as the staging area, lists all the files in the current branch.

HEAD is known as the last commit, which was marked in the check-out branch.

Q24. Name some Git GUI clients for Linux.

Ans. Below is the list of Git GUI clients for Linux:

Git Clients for Linux	Clients for Windows	Clients for Mac	Cross-platform Git Clients
Git Force	Tortoise Git	GitX-dev	Aurees
Gitg	GitHub	GitBox	SmartGit
QGit	Sourcetree	GitUp	GitKraken

Q25. Explain the benefits of using the Version Control System (VCS)?

Ans. The benefits of Version Control System(VCS) are as follows:

- All team members can work freely on any file at any time
- Allow us to compare files, identify differences, and merge the changes into a common version

- Keep a track of application builds by determining which version is currently in development, QA, and production
- Allows all team members to have a complete history of the project

Also Read: 7 Trending Tech Skills to Master in 2020

Q26. What is a Git repository?

Ans. A Git repository is a place that has a collection of files of different versions of a Project. Git stores these files either on the local repository or the remote repository. There are two types of repositories:

- Bare Repository: contains the .git folder
- Non-bare Repository: contains both the git index and the checked-out copy of working files

Q27. What is git instaweb? How is it used?

Ans. A git instaweb is a script that helps to set up a temporary instance of GitWeb on a web server for browsing local repositories. It requires a lightweight server such as Lighttpd or Webrick. It is used to automatically direct a web browser and run a webserver with an interface into the local repository.

Q28. Explain Git stash.

Ans. A git stash is a place where you can temporarily stash (or store) changes made to the working copy so we can work on something else, and then come back and reapply them afterward. A git stash is separate from the staging area, the working directory, or the repository.

Q29. What are the benefits of forking workflow?

Ans. The benefits of forking workflow are as follows:

- The contributions can be integrated without requiring everyone to push to a single central repository.
- Developers can push to their own server-side repositories while only the project maintainer can push to the official repository.
- A maintainer can accept commits from any developer without providing them write access to the official codebase.

Q30. What do you mean by the Gitflow workflow?

Ans. This is an important Git interview question. The Gitflow Workflow specifies a branching model for Git. It provides a framework for managing large projects and is mostly used for projects that have a scheduled release cycle. Gitflow assigns very specific roles to different branches and defines how and when they should interact:

- **Master:** This branch is always ready to be released on LIVE. It releases when everything is fully tested and approved.
- **Develop:** All feature branches are merged into this branch and all tests also are performed here. When everything is thoroughly checked, it can be merged into the master.
- **Feature:** Each new feature should reside in its own branch, which can be pushed to develop branch as their parent branch.
- **Hotfix:** These branches are used to quickly patch production releases. They are based on master instead of develop.

Q31. What is the difference between git remote and git clone?

Ans. With git remote, you can create, view, and delete connections to other repositories. It's used to refer to a remote repository or a central repository.

git remote	git clone
Allows you to create, view, and delete connections to other repositories.	Enables you to create a clone or copy of the target repository.
Targets a remote repository or a central repository.	It targets a different already existing repository.

Q32. How will you find out if a branch has already been merged or not?

Ans. We use the following commands to find out if a branch has already been merged or not:

- **git branch –merged master –** it will list all the branches that have been renamed into master.
- **git branch –merged** – it lists the branches that have been merged into the current branch (HEAD).
- **git branch –no-merged** – it lists the branches that have not been merged.

Q33. What does the commit object contain?

Ans. The commit object contains the following:

1. A set of files that represents the state of a project at a certain time
2. Reference to parent commit objects

3. SHAI name – a 40 character string that uniquely identifies the commit object

Q34. What is the syntax for Rebasing in git?

Ans. The syntax for Rebasing in Git is:

git rebase [new-commit]

Check out the 10 Best Software Development Tools and Frameworks to Learn

Q35. Why are the Git Stash Drop and Git Stash Clear commands used?

Ans. Git Stash drop command or <stash_id> is used to remove a particular stash that is not required.

Git stash clear command is used when all the stashes are to be removed in one go from the repository.

Q36. Explain Git Hooks.

Ans. Git hooks are simple scripts that run before or after certain actions, such as commit, push, update, or receive. They are comprised of shell scripts that are activated when you run the corresponding Git commands. Git Hooks are useful in many tasks, such as client-side validation.

Q37. What is the difference between revert and reset?

Ans. The differences between revert and reset are:

Revert	Reset
It creates a new commit that undoes the changes made in the previous commit.	It undoes the local changes that have been made to a Git repository.
New history is added to the project and the existing history is not modified.	This command may alter existing history. Reset command operates on the commit history, the staging index, and the working directory.
Command: git revert	Command: git reset

Q38. Explain the functions of the git reset –mixed and git merge –abort commands.

Ans. The git reset –mixed command undoes the changes made in the working directory and staging area.

The git merge –abort command stops the merge process and returns to the state before the merging began.

Q39. What is the difference between Git stash apply and Git stash pop?

Ans. The 'Git stash apply' and 'Git stash pop' commands are used when you have to reapply the stashed changes and start working from where you left.

The difference between them is that while the 'Git stash apply' command keeps the changes in the stash list for later use, the 'Git stash pop' command removes the changes from the stash after applying it.

Also explore:

- Paid and free online courses by Coursera
- Popular online Udemy courses
- Top online edX courses

Q40. Explain the role of the git-add command.

Ans. The git-add command adds new or changed files in your working directory to the Git staging area. Running the git add command will not change any of your work in the Git repository. Changes are only made to your repository when you execute the git commit command.

In simple terms, when you change and save a file (or multiple files, then, before you commit, you must git add. The git add command selects that file, and moves it to the staging area, for inclusion in the next commit. You can select a specific file, all files, a directory, or specific parts of a file for staging and commit. We can perform the add command multiple times before a commit.

Below is the syntax for the git add command:

git add [filename]

Q41. How to delete a branch in Git?

Ans. Deletion of Git branches is done after you have merged a branch into your codebase. To delete a branch we can use the command: git branch -d branch_name. Below are the commands to delete a git branch locally or remotely:

Deleting a branch Locally

You can delete a git branch on your local machine using the command: git branch -d <local_branch_name>

Deleting a branch Remotely

You can delete a git branch remotely using the command: git push origin –delete <remote_branch_name>

Q42. What is git reflog?

Ans. Reference logs like the commit information of when the branch was created, checked out, renamed, etc. are recorded by the reflog command. This command tracks the changes made in the repository references (branches or tags). It also records and manages the branches/tags log history that was either created locally or checked out.

Every action you perform inside of Git where data is stored can be found inside of the reflog. So, if if you think that merge, rebase, or some other action has destroyed your work, you can find it again using the reflog command.

Explore Free Online Courses with Certificates

Q43. Explain the role of the git annotate command.

Ans. The git annotate command tracks each line of the file based on the commit information. It annotates each line within the given file with information from the commit which introduced that change. The annotate command can also annotate from a given revision.

Below is the Syntax of the git annotate command:

git annotate [<options>] <file> [<revision>]

Q44. Explain the role of the git config command with examples.

Ans. The git config command is used to get and set git configuration values on a global or local project level. It uses your username to associate commits with an identity. You can also change your Git configuration, including your username.

For example:

You can give a username and email id to associate a commit with an identity. This will help you know who has made that commit.

git config –global user.name “Your Name”: It will add a username.

git config –global user.email “Your E-mail Address”: It will add an email id.

Q45. What is Git bisect? How does it help to determine the source of a (regression) bug?

Ans. Git bisect command uses a binary search algorithm to find the commit in your project’s history that introduced a bug in your code. The git bisect command divides the history of your project into the good and the bad commit range. It points your current project state to a mid-range commit snapshot. Now, this command moves through every commit id between this range while pausing at each snapshot to allow you to test the code. You declare the commit as bad if the bug exists.

The Syntax for the Git bisect command is:

git bisect <subcommand> <options>

Q46. How to squash the last N commits into a single commit?

Ans. The following are the two ways to squash the last N commits into a single commit:

- Use the below command to write the new commit message from scratch

git reset –soft HEAD~N &&git commit

- To start editing the new commit message with a concatenation of the existing commit messages, you will have to get those messages and pass them to commit:

git reset –soft HEAD~N &&git commit –edit -m”$(git log –format=%B –reverse .HEAD@{N})”

Q47. What is a .git Directory?

Ans. A .git directory consists of all the metadata of the repository. It also keeps a track of all the changes made to the files in your repository, by keeping a commit history. It keeps all information related to commits, hooks, refs, object databases, etc.

When you create a repository, you will find a .git directory inside it. If you clone any git repository on your local machine, the .git is the directory that gets copied.

Explore [Popular Online Courses]

We hope these frequently asked *Git interview questions and answers* will help you crack your next interview. Now, let's take a look at some *FAQs related to Git*.

Git FAQs

Q1. Why is Git used?

Ans. Git is an open-source project that is designed to track changes in source code during software development. It is one of the best source code management with features like task management and bug tracking and is widely used by most industry experts nowadays.

Q2. What is the best way to learn Git?

Ans. Git has a steep learning curve but there are a variety of resources that can help you learn Git as a beginner. The various resources to learn Git include online courses, eBooks, and video tutorials.

Q3. Is Git powerful in terms of performance?

Ans. Git has a very powerful raw performance. Whether it is branching, merging, or comparing the past versions Git is robust and optimized. It emphasizes the content and uses a blend of delta encoding and compression.

Q4. Is Git difficult to learn?

Ans. Git is considered a bit hard to learn and understand. While it has a steep learning curve, many resources will make it easier for you to understand Git better without feeling overwhelmed. The best way to learn Git is to avoid learning everything at once. Moving slowly and learning only what you need will make it easier for you to learn git.

Q5. Why should I learn Git?

Ans. As a developer, knowledge of Git will be helpful for you. There are multiple benefits of learning Git. Git simplifies the process of working with other people and makes it easier for you to collaborate on projects. It enables team members to work on files at any time and easily merge their changes with the master branch of the project.

If you have recently completed a professional course/certification, click here to submit a review.

4.75 avg. rating (94% score) - 4 votes

INTERVIEW QUESTIONS

Browse Courses by Categories

IT Services Courses
Software Tools Courses
Masters and Certificate Programs Courses
Big Data Courses
Web Development Courses
Full Stack Development Courses
Programming Courses
Operating System Courses
QA and Testing Courses
Networking and Hardware Courses
Cloud Technologies Courses
Databases Courses
Cybersecurity Courses

Related Posts

Nov 10 2021
TECHNOLOGY
Top Reasons to Learn Python and Hadoop

Oct 19 2021
TECHNOLOGY
Importance of Java Memory Management

Dec 23 2021
TECHNOLOGY
What is Coding: Difference Between Coding And Programming

Popular Courses
Artificial intelligence and Machine Learning Courses
Data Science Courses
Management Courses
Digital Marketing Courses
Product Management Courses
Business Analytics Courses
Python Courses
Sales Management Courses
Healthcare Courses
Accounting and Finance Courses

Popular Platforms
Coursera Courses
Udemy Courses
Swayam & NPTEL Courses
Edx Courses
IIM Courses
Udacity Courses
Ivy league Courses
IIT Courses
Google Courses

Courses & Certifications
Degree Programs
Free online courses
Free Government Courses
Top courses starting soon
Programming for managers
Top 10 Free AI Courses
Executive programs
Trending courses
Top Interview Questions & Answers

Information
About us
Contact us
Sitemap
Customer Support
Feedback
Privacy Policy
Terms & Conditions
Grievances
Trust & Safety
Locate Branch
Register Here
Blog

Contact Us
Toll Free - 1800-102-5557
(9.00 AM to 8.00PM IST)
Email - nl.feedback@naukri.com
Connect with us

Partner Sites
Naukri Naukrigulf Shiksha 99acres Firstnaukri Jeevansathi Ambitionbox Job Hai Policybazaar Brijj Zomato Meritnation