

# Top GIT Interview QnA

KASPER  
ANALYTICS

**Q1) What is GIT?**

**ANS:** Git is an open-source distributed version control system and source code management (SCM) system with an insistence to control small and large projects with speed and efficiency.



**Q2) Which language is used in Git?**

**ANS:** Git uses 'C' language. Git is quick, and 'C' language makes this possible by decreasing the overhead of run times contained with high-level languages.

**Q3) What is a repository in Git?**

**ANS:** A repository consists of a list named .git, where git holds all of its metadata for the catalog. The content of the .git file is private to Git.

**Q4) What is 'bare repository' in Git?**

**ANS:** A "bare" repository in Git includes the version control information and no working files (no tree), and it doesn't include the special. git sub-directory.

Instead, it consists of all the contents of the .git sub-directory directly in the main directory itself, whereas working list comprises of:

1. A .git subdirectory with all the Git associated revision history of your repo.
2. A working tree, or find out copies of your project files.

**Q5) What is the purpose of GIT stash?**

**ANS:** GIT stash takes the present state of the working file and index and puts in on the stack for next and gives you back a clean working file. So in case if you are in the middle of object and require to jump over to the other task, and at the same time you don't want to lose your current edits, you can use GIT stash.

**Q6) What is GIT stash drop?**

**ANS:** When you are done with the stashed element or want to delete it from the directory, run the git 'stash drop' command.

It will delete the last added stash item by default, and it can also remove a specific topic if you include as an argument.

**Q7) What are the advantages of using GIT?**

**ANS:** Here are some of the essential advantages of Git:

- Data repetition and data replication is possible
- It is a much applicable service
- For one depository you can have only one directory of Git
- The network performance and disk application are excellent
- It is effortless to collaborate on any project
- You can work on any plan within the Git

**Q8) What is the function of 'GIT PUSH' in GIT?**

**ANS:** 'GIT PUSH' updates remote refs along with related objects

**Q9) Why do we require branching in GIT?**

**ANS:** With the help of branching, you can keep your branch, and you can also jump between the different branches. You can go to your past work while at the same time keeping your recent work intact.

**Q10) What is the purpose of 'git config'?**

**ANS:** The 'Git config' is a great method to configure your choice for the Git installation. Using this command, you can describe the repository behavior, preferences, and user information.

**Q11) What is the definition of "Index" or "Staging Area" in GIT?**

**ANS:** When you are making the commits, you can make innovation to it, format it and review it in the common area known as 'Staging Area' or 'Index'.

**Q12) What is a 'conflict' in git?**

**ANS:** A 'conflict' appears when the commit that has to be combined has some change in one place, and the current act also has a change at the same place. Git will not be easy to predict which change should take precedence.

**Q13) What is the difference between git pull and git fetch?**

**ANS:** Git pull command pulls innovation or commits from a specific branch from your central repository and updates your object branch in your local repository.

Git fetch is also used for the same objective, but it works in a slightly different method. When you behave a git fetch, it pulls all new commits from the desired branch and saves it in a new branch in your local repository.

If you need to reflect these changes in your target branch, git fetch should be followed with a git merge. Your target branch will only be restored after combining the target branch and fetched branch.

To make it simple for you, remember the equation below:

**Git pull = git fetch + git merge**

**Q14) How to resolve a conflict in Git?**

**ANS:** If you need to resolve a conflict in Git, edit the list for fixing the different changes, and then you can run "git add" to add the resolved directory, and after that, you can run the 'git commit' for committing the repaired merge.

**Q15) What is the purpose of the git clone?**

**ANS:** The git clone command generates a copy of a current Git repository. To get the copy of a central repository, 'cloning' is the simplest way used by programmers.

**Q16) What is git pull origin?**

**ANS:** pull is a get and a consolidation. 'git pull origin master' brings submits from the master branch of the source remote (into the local origin/master branch), and then it combines origin/master into the branch you currently have looked out.

**Q17) What does git commit?**

**ANS:** Git commits "records changes to the storehouse" while git push "updates remote refs along with contained objects"

So the first one is used in a network with your local repository, while the latter one is used to communicate with a remote repository.

**Q18) Why GIT better than Subversion?**

**ANS:** GIT is an open source version control framework; it will enable you to run 'adaptations' of a task, which demonstrate the changes that were made to the code over time also it allows you keep the backtrack if vital and fix those changes.

Multiple developers can check out, and transfer changes, and each change can then be attributed to a particular developer.

**Q19) Explain what is commit message?**

**ANS:** Commit message is a component of git which shows up when you submit a change. Git gives you a content tool where you can enter the adjustments made to a commit.

**Q20) Why is it desirable to create an additional commit rather than amending an existing commit?**

**ANS:** There are couples of reason

1. The correct activity will devastate the express that was recently saved in a commit. If only the commit message gets changed, that's not a problem. But if the contents are being modified, chances of excluding something important remains more.
2. Abusing "git commit- amends" can cause a small commit to increase and acquire inappropriate changes.

**Q21) What does 'hooks' comprise of in Git?**

**ANS:** This index comprises of Shell contents which are enacted after running the relating git commands.

For instance, Git will attempt to execute the post-commit content after you run a commit.

**Q22) What is the distinction between Git and Github?**

**ANS:**

	
<b>Git</b>	<b>GitHub</b>
Software	Service
Version control	Git repository hosting
Maintained by Linux	Maintained by Microsoft
Open-Source	Free or paid membership
No user management	Built-in user management
Locally installed	Hosted on the web
Minimal external tool configuration	Active marketplace for tool integration
Little to no competition	High competition

**A N A L Y T I C S**

**Q23) In Git, how would you return a commit that has just been pushed and made open?**

**ANS:** There can be two answers to this question and ensure that you incorporate both because any of the below choices can be utilized relying upon the circumstance:

Remove or fix the bad document in another commit and push it to the remote repository.

This is a unique approach to correct a mistake. Once you have necessary changes to the record, commit it to the remote repository for that I will utilize.

```
git submit - m "commit message."
```

Make another commit that fixes all changes that were made in the terrible commit. to do this, I will utilize a command.

```
git revert <name of bad commit>
```

#### Q24) What does the committed item contain?

**ANS:** Commit item contains the following parts; you should specify all the three present below:

A set of records, representing to the condition of a task at a given purpose of time

References to parent commit objects

An SHA1 name, a 40 character string that uniquely distinguishes the commit object.

#### Q25) Describing branching systems you have utilized?

**ANS:** This question is a challenge to test your branching knowledge with Git along these lines, inform them regarding how you have utilized branching in your past activity and what reason does it serves, you can refer the below mention points:

➤ Feature Branching:

- A component branch model keeps the majority of the changes for a specific element within a branch.
- At the point when the item is throughout tested and approved by automated tests, the branch is then converged into master.



- Task Branching
  - In this model, each assignment is actualized on its branch with the undertaking key included in the branch name.
  - It is anything but difficult to see which code actualizes which task, search for the task key in the branch name.
- Release Branching
  - Once the create branch has procured enough features for a discharge, you can clone that branch to frame a Release branch.
  - Making this branch begins the following discharge cycle so that no new features can be included after this point, just bug fixes, documentation age, and other release oriented assignments ought to go in this branch.
  - When it is prepared to deliver, the release gets converged into master and labeled with a form number.
  - Likewise, it should be converged once again into creating a branch, which may have advanced since the release was started.

At last, disclose to them that branching methodologies fluctuate starting with one association then onto the next, so I realize essential branching activities like delete, merge, checking out a branch, etc.

**Q26) By what method will you know in Git if a branch has just been combined into master?**

**ANS:** The appropriate response is immediate.

To know whether a branch has been merged into master or not you can utilize the below commands:

- git branch - merged It records the branches that have been merged into the present branch.
- git branch - no merged It records the branches that have not been merged.

**Q27) How might you fix a messed up submit?**

**ANS:** To fix any messed up commit, you will utilize the order "git commit?correct."  
By running this direction, you can set the wrecked commit message in the editor.

**Q28) Mention the various Git repository hosting functions.**

**ANS:** The following are the Git repository hosting functions:

1. Pikacode
2. Visual Studio Online
3. GitHub
4. GitEnterprise
5. SourceForge.net

**Q29) Mention some of the best graphical GIT customers for LINUX?**

**ANS:** Some of the best GIT customer for LINUX is

1. Git Cola
2. Smart git
3. Git-g
4. Git GUI
5. Giggie
6. QGit

**Q30) What is Subgit? Why use it?**

**ANS:** 'Subgit' is a tool that migrates SVN to Git. It is a stable and stress-free migration. Subgit is one of the solutions for a company-wide migration from SVN to Git that is:

1. It is much superior to git-svn
2. No need to change the infrastructure that is already placed.
3. It allows using all git and all sub-version features.
4. It provides stress free migration experience.