By OnlineInterviewQuestions.com

Git Interview Questions

What is Git?

GIT is one of the most popular version control systems for enterprise applications and big data solutions. It is also a must-have piece of technology for all android, software or iOS developers. Many companies use the GIT framework in their software development architectures. This technology plays a vital role in a lot of organizations. Therefore, a lot of opportunities in the GIT field from many reputed companies are available for the developers. Software Engineers, **Software architects,** or Development managers with basic software development and **GIT knowledge** are offered handsome annual packages. It is a great field for those who seek advancement in their career. Here are some of the *GIT interview questions along with their answers* that will help you in cracking the GIT interview and start cruising into your dream career.

Q1. What is GIT and why do we use it?

GIT is a **Distributed Version Control System (DVCS)** and **Source Code Management System (SCMS)** which can track changes to a file and lets you revert back to any particular change. It can also handle small and large projects with efficiency and speed.

GIT is recommended to be used because of its following advantages:

- Any project can use GIT without any restrictions.
- It is collaboration-friendly.
- It has superior disk utilization and network performance.
- There can be only one.GIT directory per repository.
- High availability, Data redundancy, and replication.

Q2. What are GIT stash and GIT stash drop?

GIT stash:
It takes the current state of working directory and index and pushes into the stack for later and returns to you the clean working directory.GIT stash will help you in the instances where you are working on a project and you want to switch the branches to work on something else.

GIT stash drop:

When you are done and want to eliminate the stashed item from the list, then running the GIT "stash drop" command will remove the last added stash item by default and can also remove the specific item if any argument is included or mentioned.

Q3. Name some GIT commands and also explain their functions?

- **GIT diff** – It shows the changes between commits, commits and working tree.
- **GIT status** – It shows the difference between working directories and index.
- **GIT stash applies** – It is used to bring back the saved changes on the working directory.
- **GIT rm** – It removes the files from the staging area and also of the disk.
- **GIT log** – It is used to find the specific commit in the history.
- **GIT add** – It adds file changes in the existing directory to the index.
- **GIT reset** – It is used to reset the index and as well as the working directory to the state of the last commit.
- **GIT checkout** – It is used to update the directories of the working tree with those from another branch without merging.
- **GIT Is tree** – It represents a tree object including the mode and the name of each item.
- **GIT** instaweb – It automatically directs a web browser and runs the web server with an interface into your local repository.

Q4. What is a GIT commit and what does the commit object contain?

A GIT commit is a command that is executed in the course of a project to records progress.
The commit object contains the following components

- A set of files that represents the state of the file at any instance of time.
- Reference to the parent commit
- An SHAI name, a 40-character string that can uniquely identify the commit object.

Q5. What is a repository and how will you create it?

A repository in GIT contains the ".GIT" directory, where the GIT stores all of its metadata for the repository. The content of ".GIT" directory remains private to GIT.
To create a repository first the user needs to create the directory for the project and then the user needs to run "GIT init" command. This command creates the GIT directory in the project directory without emptying the directory.

Q6. What is GIT Bisect and what is its purpose?

The GIT Bisect command is used to find the commit that has introduced a bug by using binary search.

Purpose of GIT Bisect command

The command uses a binary search algorithm to find which commit in the project's history has introduced a bug. Before the bug is introduced into the commit, the commit is referred as "good"; after the introduction of the bug, it is referred as "bad".

Then the command used picks a commit between those endpoints and asks what kind of commit is it. The process continues till it finds the exact commit that introduced the change.

Q7. What is the purpose of branching and how many branching strategies can you apply?

The branching in GIT is done to allow the user to create their own branch and toggle between those branches. It will allow the user to go to the previous work keeping the current work intact.

Branching Strategies:

- **Feature Branching** – It keeps all the changes of a specific feature in a branch and when the feature is completely tested and validated it is merged into the master.
- **Task branching** – It is included in its own branch with the task key included in the branch name.
- **Release Branching** – When the developed branch acquires enough features for a release, the user can clone that branch to form a releasing branch.

Q8. What is a "conflict" in GIT and how is it resolved?

When a commit that has to be merged has some changes in one place, which also has the changes of current commit, then the conflict arises. The GIT will not be able to predict which change will take the precedence. In order to resolve the conflict in GIT: we have to edit the files to fix the conflicting changes and then add the resolved files by running the "GIT add" command; later on, to commit the repaired merge run the "GIT commit" command. GIT identifies the position and sets the parents of the commit correctly.

Q9. Why GIT and not SVN?

GIT is an open source version control system that will allow the user to run "version" of a project and shows the changes that were made to the code over time; allowing the user to keep the backtrack if necessary and undo those changes.
It allows multiple developers to check out and uploads changes and each change can then be attributed to the specific developer.

Q10. What commands will you use to bring a new feature to the main branch?

By running "GIT merge" or "GIT pull" commands, the user can bring a new feature to the main branch.

Q11. Distinguish between GIT fetch and GIT pull?

GIT fetch : It downloads only the new data from the remote repository and does not integrate any of the downloaded data into your working files. Providing a view of the data is all it does.

GIT pull: It downloads as well as merges the data from the remote repository into the local working files. This may also lead to merging conflicts if the user's local changes are not yet committed. Using the "GIT stash" command hides the local changes.

Q12. What is subGIT and why is it used?

SubGIT – It is a tool for smooth and stress-free subversion to GIT migration and also a solution for a company-wide subversion to GIT migration that is:

- It allows to make use of all GIT and subversion features.
- It provides genuine stress-free migration experience.
- It doesn't require any change in the infrastructure that is already placed.
- It is considered to be much better than GIT-SVN

Q13. What is Rebasing syntax in Git?

Syntax for Rebashing: "git rebase [new-commit] "

Q14. What is git reflog?

Git holds a record of updates to the apex of branches employing a mechanism described reference logs, as known as **reflogs**. Various Git commands receive a parameter for defining a reference (ref), a pointer to a commit. Typical examples are; git checkout, git reset, git merge, etc.

Reflogs record when refs were refreshed in the local repository and in addition to branch tip reflogs, a special reflog is reserved for the Git stash. Reflogs are filed in directories supporting the local repository's .git directory. git reflog directories can be located at **.git/logs/refs/heads/**., **.git/logs/HEAD**, and also **.git/logs/refs/stash** if the git stash has been employed on the repo.

Q15. What is git push?

The **git push** command is applied to upload local repository content to a remote repository. Pushing is the way you assign commits from local repository to a remote repo and it's the counterpart to git fetch, but whereas retrieving imports commits to local branches, pushing exports commits to remote branches. Remote branches are set up using the git remote command and pushing has the ability to overwrite modifications, care should be taken when pushing.

Please Visit OnlineInterviewquestions.com to download more pdfs