A1 TRAINING INSTITUTE
(Only Coding)

Git Interview

Questions with answers from basic to advanced

Basic Level

1. **What is Git?**

- Answer: Git is a distributed version control system designed to handle everything from small to very large projects with speed and efficiency. It allows multiple developers to work on a project simultaneously without overwriting each other’s changes.

2. What is a repository in Git?

- Answer: A repository (repo) in Git is a storage location where your project's files and their revision history are stored. It can be local to your computer or hosted on a remote server.

Basic Level

3. What are the differences between Git and GitHub?

- Answer: Git is a version control system that lets you manage and keep track of your source code history. GitHub is a cloud-based hosting service that lets you manage Git repositories.

4. How do you create a new Git repository?

- Answer: You can create a new Git repository by using the command ‘git init’ for an existing directory or ‘git clone <url>’ to copy an existing repository from a remote source.

Basic Level

5. **What is a commit in Git?**

- Answer: A commit is a snapshot of your repository at a specific point in time. It is like saving a version of your project that you can revert to or compare against later.

Intermediate Level

1. How do you stage files for a commit in Git?

- Answer: You stage files for a commit using the ‘git add’ command. For example, ‘git add <filename>’ stages a single file, while ‘git add’ stages all changes in the current directory.

2. What is the difference between git pull and git fetch?

- Answer: ‘git fetch’ downloads objects and refs from another repository, while ‘git pull’ fetches from the remote repository and merges it with the local branch.

Intermediate Level

3. How do you resolve a merge conflict in Git?

- Answer: To resolve a merge conflict, you need to manually edit the conflicting files to reconcile the differences between the branches. After editing, you stage the resolved files using ‘git add’ and then commit the changes.

4. What is a branch in Git?

- Answer: A branch in Git is a separate line of development. It allows you to work on different features or bug fixes independently from the main project.

Intermediate Level

5. **How do you delete a branch in Git?**

- Answer: You can delete a branch using ‘git branch -d <branch_name>’ if it has been merged, or ‘git branch -D <branch_name>’ to force delete it if it hasn't been merged.

Advanced Level

1. **What is Git rebase and how does it differ from Git merge?**

- Answer: 'git rebase' is a command that lets you integrate changes from one branch into another. It moves or combines a sequence of commits to a new base commit. Unlike 'git merge' which preserves the history of commits, 'git rebase' rewrites the commit history.

2. **What is a Git stash and how do you use it?**

- Answer: Git stash is a way to temporarily save changes that you don't want to commit yet but need to switch branches. You can use 'git stash' to save your work and 'git stash pop' to reapply the stashed changes.

Advanced Level

3. How do you cherry-pick a commit in Git?

- Answer: Cherry-picking in Git means to apply the changes introduced by some existing commit. You can do this using ‘git cherry-pick <commit_hash>’.

4. What is a submodule in Git?

- Answer: A submodule is a way to keep a Git repository as a subdirectory of another Git repository. This can be useful for including and tracking dependencies.

Advanced Level

5. How do you perform a bisect in Git?

- Answer: ‘git bisect’ helps you find the commit that introduced a bug by performing a binary search. You start with ‘git bisect start’, mark a good commit with ‘git bisect good <commit>’ and a bad commit with ‘git bisect bad <commit>’, and Git will check out a commit halfway between them. You test that commit and mark it as good or bad, and Git repeats the process until it finds the offending commit.