https://www.automationtestinginsider.com/2019/10/selenium-questions-part9-git-and-github.html

Selenium Questions Part9: Git and Github

1. What is Git? What is the difference 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. If you have open-source projects that use Git, then GitHub is designed to help you better manage them.

2. What is the advantage of using GitHub for Selenium?

Answer: GitHub is a cloud-based hosting service that lets you manage Git repositories; it helps to have a backup code in case of physical failures.

- github supports branching, so we can have multiple versions of code.
- github supports project cloning, so it helps in easily distribution of project across multiple teams and multiple locations
- github supports code pull so anyone with access rights can pull code in local machine. This can also be integrated with jenkins.
- github supports code push so anyone with access rights can checkin code in github central repository.

3. How to handle git conflicts?

Answer: Git can handle on its own most merges by using its automatic merging features. There arises a conflict when two separate branches have made edits to the same line in a file, or when a file has been deleted in one branch but edited in the other. Conflicts are most likely to happen when working in a team environment.

- Identify the files that have caused the conflict.
- Make the necessary changes in the files so that conflict does not arise again.
- Add these files by the command git add.
- Finally to commit the changed file using the command git commit

Please refer below YouTube video link

https://www.youtube.com/watch?v=CQqdTDBlopg&feature=youtu.be

4. Explain different Git commands?

Answer: Below are the most git commands

1. Initialize a repo

Create an empty git repo or re-initialize an existing one

```
$ git init [repository path]
```

2. git remote add [variable name] [Remote Server Link]

This command is used to connect your local repository to the remote server.

Example:

```
git remote add origin https://github.com/hverma22/Test5
```

3. git clone

This command is used to obtain a repository from an existing URL.

```
git clone [url]
```

Example: navigate to your repo path where you want to clone and write below command in cmd

```
git clone https://github.com/hverma22/Test2
```

4. How to Create a New Branch in Git

To create a new branch use:

```
$ git checkout -b <new_branch_name>
```

5. How to List Branches in Git

```
$ git branch
```

Example output:

```
develop

my_feature

master
```

6. git log

```
git log
```

This command is used to list the version history for the current branch.

Example:

```
git log --oneline
```

7. git merge

```
git merge [branch name]
```

This command merges the specified branch's history into the current branch.

8. How to Switch Branches in Git

When you create a new branch then Git automatically switches to the new branch.

If you have multiple branches, then you can easily switch between branches with git checkout:

```
$ git checkout master

$ git checkout develop
$ git checkout my_feature
```

You can get the specific previous version as well.

Command: git checkout <ChangeID> <filePath with extenion>

Example:

```
git checkout 6475fgh5 pom.xml
```

9. How to Delete Branches in Git

To delete a local branch:

```
$ git branch -d <local_branch>
```

To delete a remote branch on origin:

```
$ git push origin :<remote_branch>
```

10. Git Stage Files

To stage or simply add files, you need to use git add command. You can stage individual files:

```
$ git add foo.js
```

or all files at once:

```
$ git add .
```

11. git diff

This command shows the file differences which are not yet staged.

Example:

```
git diff –-staged
git diff [first branch] [second branch]
```

12.Git Unstage Changes

If you want to remove a certain file from the stage:

```
$ git reset HEAD foo.js
```

Or remove all staged files:

```
$ git reset HEAD .
```

13. Git Status

If you want to see what files have been created, modified or deleted, Git status will show you a report.

```
$ git status
```

14. git rm

This command deletes the file from your working directory and stages the deletion.

git rm [file]

15.git commit

git commit -m “[commit message]”

This command records or snapshots the file permanently in the version history.

Example:

git commit -m “First Commit”

16. git show

git show [commit]

This command shows the metadata and content changes of the specified commit.

Command: git show <ChangeID>:<FilePath>

Example:

git show 45dhfg56:/src/test/newtest.xml

17. Undoing Commits

The following command will undo your most recent commit and put those changes back into staging, so you don’t lose any work:

$ git reset --soft HEAD~1

To completely delete the commit and throw away any changes use:

$ git reset --hard HEAD~1

18. git push - After you have committed your changes, next is to push to a remote repository.

git push [variable name] master

This command sends the committed changes of master branch to your remote repository.

Example:

Push a local branch for the first time:

git push origin master

git push origin master --force

After that, then you can just use

$ git push

19. To push a local branch to a different remote branch, you can use:

$ git push origin <local_branch>:<remote_branch>

20. Undo Last Push

If you have to undo your last push, you can use:

```
$ git reset --hard HEAD~1 && git push -f origin master
```

21.git config

This command sets the author name and email address respectively to be used with your commits.

```
git config –global user.name "[name]"
```

```
git config –global user.email "[email address]"
```

Example:

```
git config user.name "Hitendra Kuamar Verma"
```

```
git config user.email "Hitendra@Hitendra-PC"
```

22. git pull

```
git pull [Repository Link]
```

This command fetches and merges changes on the remote server to your working directory.

Example:

```
git pull https://github.com/hverma22/Test2.git
```

5. What is the version control tool you are using and tell me the steps what you follow and how will you resolve conflicts?

Answer: Please refer below YouTube video link

https://www.youtube.com/watch?v=CQqdTDBlopg&feature=youtu.be

6. What is the difference between SVN & GIT?

Answer: Below are the differences -

- Git is a distributed VCS; SVN is a non-distributed VCS.
- Git uses multiple repositories including a centralized repository and server, as well as some local repositories; SVN is a centralized version control system.
- The content in Git is stored as metadata; SVN stores files of content.
- Git branches are easier to work with than SVN branches.
- Git does not have the global revision number feature like SVN has.
- Git has better content protection than SVN.
- Git was developed for Linux kernel by Linus Torvalds; SVN was developed by CollabNet, Inc.
- Git belongs to the 3rd generation of Version Control tools; SVN belongs to the 2nd generation of Version Control tools

7. How do you maintain source code in GIT?

Answer: Please refer below link –

https://www.automationtestinginsider.com/2020/06/selenium-integration-with-git-and-github.html

8. Suppose there are 10 classes & I want to push only 5 classes, how do you do that?

Answer: Normally we commit to git, all files are going to git but in your scenario push only single file git. For this, you have to run specific command to push the only single file to git.

```
$ git commit -m "Message goes here" filename
```

Example to push to single file to git

```
$ git commit -m "Pushing Only Single file to git" config/file1.txt
```

Let’s take look how to push one or two or three files to git in a single commit.

```
$ git commit -m "Message goes here" file1 file2 file3
```

Example to push to three files to git

```
$ git commit -m "Pushing Only three files to git" config/file1.txt  config/file2.txt config/file3.txt
```

9. Mention the various Git repository hosting functions.

Answer:

- Github
- Gitlab
- Bitbucket
- SourceForge
- GitEnterprise

10. In Git how do you revert a commit that has already been pushed and made public?

Answer: There can be two approaches to tackle this question and make sure that you include both because any of the below options can be used depending on the situation:

Remove or fix the bad file in a new commit and then push it to the remote repository. This is the most obvious way to fix an error. Once you have made necessary changes to the file, then commit it to the remote repository using the command: git commit -m “commit message”

Also, you can create a new commit that undoes all changes that were made in the bad commit. To do this use the command

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

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

Answer: Git pull command pulls new changes or commits from a particular branch from your central repository and updates your target branch in your local repository.
Git fetch is also used for the same purpose but it works in a slightly different way. When you perform a git fetch, it pulls all new commits from the desired branch and stores it in a new branch in your local repository. If you want to reflect these changes in your target branch, git fetch must be followed with a git merge. Your target branch will only be updated after merging the target branch and fetched branch. Just to make it easy for you, remember the

equation below:

Git pull = git fetch + git merge

12. What is git stash?

Answer: Often, when you’ve been working on part of your project, things are in a messy state and you want to switch branches for some time to work on something else. The problem is, you don’t want to do a commit of half-done work just so you can get back to this point later. The answer to this issue is Git stash.

Stashing takes your working directory that is, your modified tracked files and staged changes and saves it on a stack of unfinished changes that you can reapply at any time.

```
$ git status

  modified:   index.php

  modified:   css/styles.css
```

Apply Git Stash

```
$ git stash

Saved working directory and index state WIP on master:

   2dfe283 Implement the new login box

HEAD is now at 2dfe283 Implement the new login box
```

Git's Stash is meant as a temporary storage. When you're ready to continue where you left off, you can restore the saved state easily:

```
$ git stash pop
```

13. What is the function of ‘git stash apply’?

Answer: If you want to continue working where you had left your work then ‘git stash apply‘command is used to bring back the saved changes onto your current working directory.

```
git stash apply n
```

To get list of stashes:

```
git stash list
```

14. What is the function of ‘git config’?

Answer: Git uses your username to associate commits with an identity. The git config command can be used to change your Git configuration, including your username.

Now explain with an example.

Suppose you want to give a username and email id to associate a commit with an identity so that you can know who has made a particular commit. For that I will use:

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

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

15. How will you know in Git if a branch has already been merged into master?

Answer: To know if a branch has been merged into master or not you can use the below commands:

git branch --merged – It lists the branches that have been merged into the current branch.

git branch --no-merged – It lists the branches that have not been merged.

16. Can you explain the Gitflow workflow?

Answer: To record the history of the project, Gitflow workflow employs two parallel long-running branches – master and develop:

Master – this branch is always ready to be released on LIVE, with everything fully tested and approved (production-ready).

Hotfix – these branches are used to quickly patch production releases. These branches are a lot like release branches and feature branches except they’re based on master instead of develop.

Develop – this is the branch to which all feature branches are merged and where all tests are performed. Only when everything’s been thoroughly checked and fixed it can be merged to the master.

Feature – each new feature should reside in its own branch, which can be pushed to the develop branch as their parent one.

Please refer below YouTube video to understand the explanation of above Q/A

Please refer below Git and Github Playlist here:

Posted by Automation Testing Insider at October 16, 2019
Email ThisBlogThis!Share to TwitterShare to FacebookShare to Pinterest
Labels: Most frequently asked questions in Selenium, Selenium Automation Testing Questions and Answers, Selenium Interview Questions and Answers

No comments:

Post a Comment

https://www.simplilearn.com/tutorials/git-tutorial/git-interview-questions

Top 40 Git Interview Questions and Answers [Updated 2022]

Lesson 11 of 11By Simplilearn

Last updated on Feb 16, 2022163249

Previous

Table of Contents

Git is the most popular source code management tool. Whether you're a programmer or a non-technical person, Git will help you with its collaboration features like bug tracking, task management, and wikis.

Git plays a significant role in an organization to implement DevOps methodologies. Looking at the popularity of Git in DevOps, it's quite evident that there will be many job opportunities in the near future. According to a report from Grand View Research, the global DevOps market is expected to reach 12.85 Bn by 2025, growing at a CAGR of 18.60%. Here are some of the top Git interview questions that will help you crack an interview.

Basic Git Interview Questions

1. What is Git?

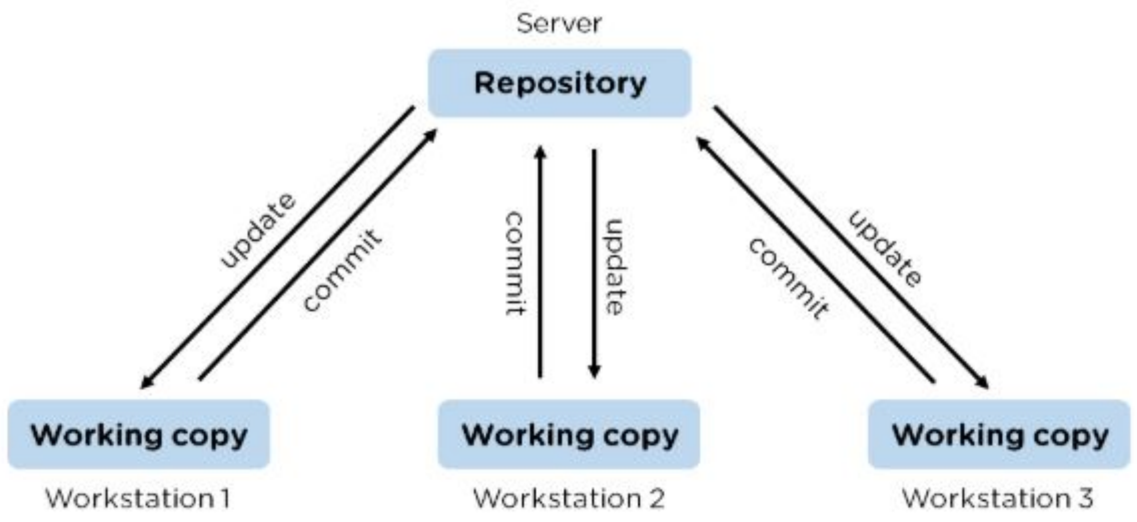
Git is a version control system for tracking changes in computer files and is used to help coordinate work among several people on a project while tracking progress over time. In other words, it's a tool that facilitates source code management in software development.

Git favors both programmers and non-technical users by keeping track of their project files. It enables multiple users to work together and handles large projects efficiently.

2. What do you understand by the term 'Version Control System'?

A version control system (VCS) records all the changes made to a file or set of data, so a specific version may be called later if needed.

This helps ensure that all team members are working on the latest version of the file



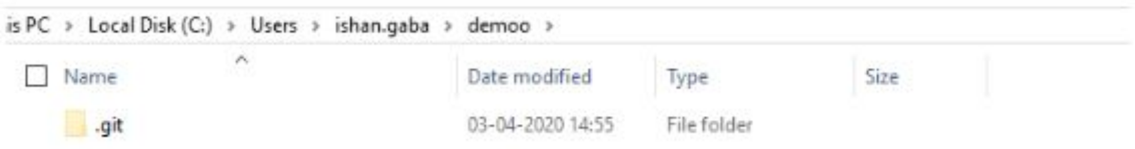
3. What's the difference between Git and GitHub?

Git	GitHub
Git is a software	GitHub is a service
Git can be installed locally on the system	GitHub is hosted on the web
Provides a desktop interface called git GUI	

	Provides a desktop interface called GitHub Desktop.
It does not support user management features	Provides built-in user management

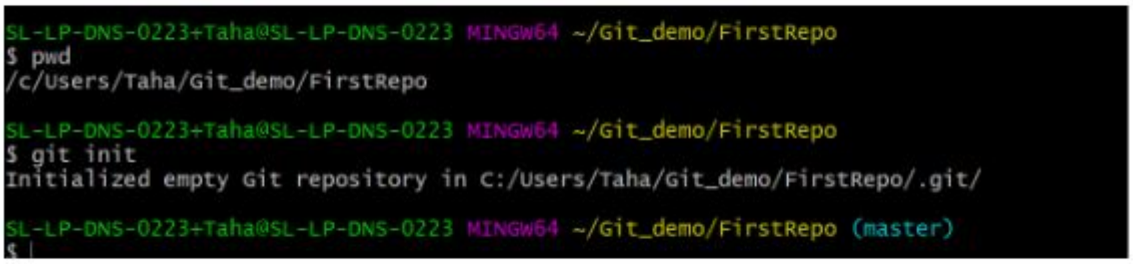
4. What is a Git repository?

Git repository refers to a place where all the Git files are stored. These files can either be stored on the local repository or on the remote repository.



5. How can you initialize a repository in Git?

If you want to initialize an empty repository to a directory in Git, you need to enter the git init command. After this command, a hidden .git folder will appear.



```
SL-LP-DNS-0223+Taha@SL-LP-DNS-0223 MINGW64 ~/Git_demo/FirstRepo
$ pwd
/c/Users/Taha/Git_demo/FirstRepo

SL-LP-DNS-0223+Taha@SL-LP-DNS-0223 MINGW64 ~/Git_demo/FirstRepo
$ git init
Initialized empty Git repository in C:/Users/Taha/Git_demo/FirstRepo/.git/

SL-LP-DNS-0223+Taha@SL-LP-DNS-0223 MINGW64 ~/Git_demo/FirstRepo (master)
$ |
```

6. How is Git different from Subversion (SVN)?

GIT	SVN
Git is a distributed decentralized version control system	SVN is a centralized version control system.
Git stores content in the form of metadata.	SVN stored data in the form of files.
The master contains the latest stable release.	In SVN, the trunk directory has the latest stable release
The contents of Git are hashed using the SHA-1 hash algorithm.	SVN doesn't support hashed contents.

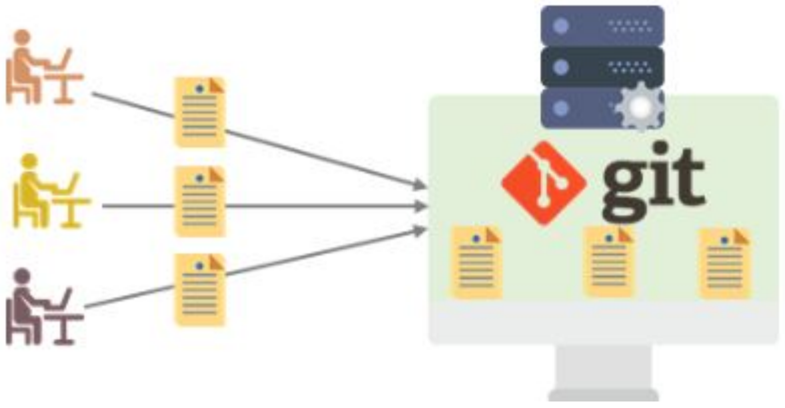
7. Name a few Git commands with their function.

- Git config - Configure the username and email address
- Git add - Add one or more files to the staging area
- Git diff - View the changes made to the file
- Git init - Initialize an empty Git repository

- Git commit - Commit changes to head but not to the remote repository

8. What are the advantages of using Git?

- Faster release cycles
- Easy team collaboration
- Widespread acceptance
- Maintains the integrity of source code
- Pull requests



9. What language is used in Git?

Git is a fast and reliable version control system, and the language that makes this possible is 'C.'

Using C language reduces the overhead of run times, which are common in high-level languages.

10. What is the correct syntax to add a message to a commit?

git commit -m "x files created"

11. Which command is used to create an empty Git repository?

git init - This command helps to create an empty repository while working on a project.

12. What does git pull origin master do?

The git pull origin master fetches all the changes from the master branch onto the origin and integrates them into the local branch.

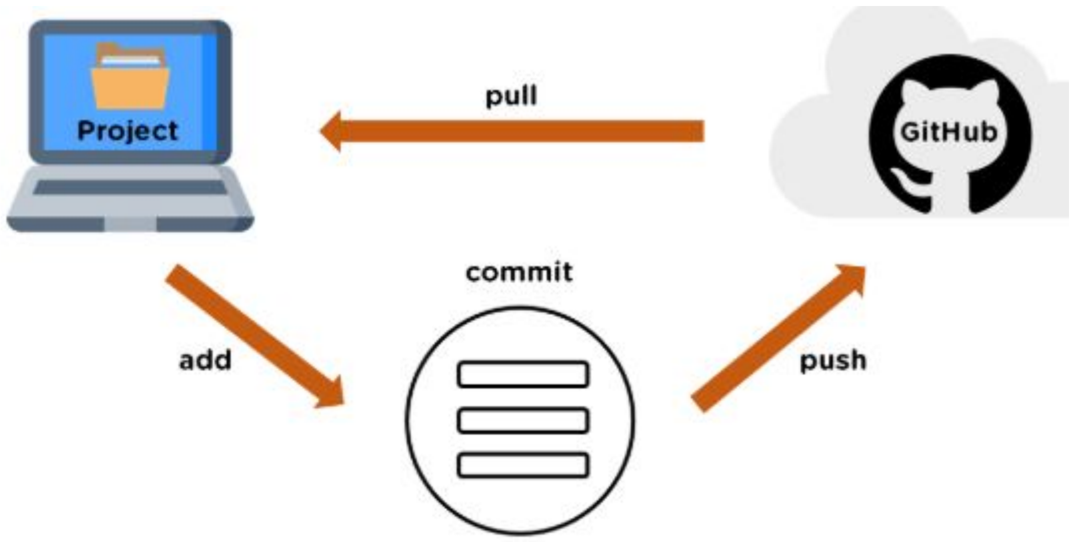
git pull = git fetch + git merge origin/ master

After having gone through the beginner level Git interview questions, let us now look at intermediate GIT interview questions and answers.

Intermediate Git Interview Questions

13. What does the git push command do?

The Git push command is used to push the content in a local repository to a remote repository. After a local repository has been modified, a push is executed to share the modifications with remote team members.



14. Difference between git fetch and git pull.

Git Fetch	Git Pull
The Git fetch command only downloads new data from a remote repository.	Git pull updates the current HEAD branch with the latest changes from the remote server.
It does not integrate any of these new data into your working files.	Downloads new data and integrate it with the current working files.
Command - git fetch origin git fetch --all	Tries to merge remote changes with your local ones.

	Command - git pull origin master

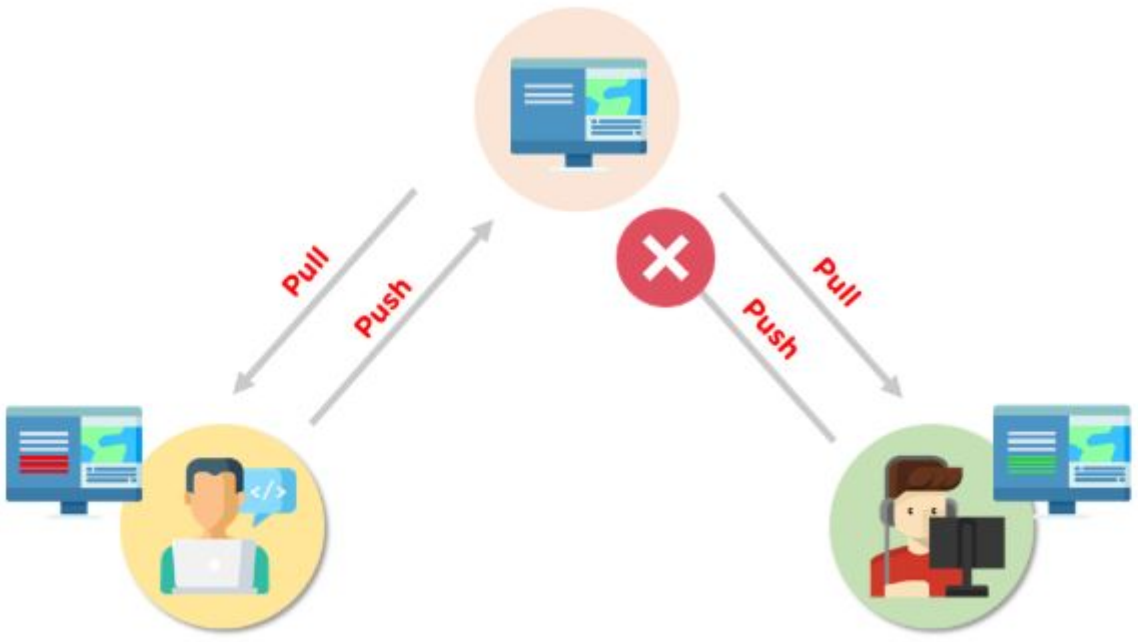
15. GitHub, GitLab and Bitbucket are examples of git repository _______ function?

hosting. All the three are services for hosting Git repositories

16. What do you understand about the Git merge conflict?

A Git merge conflict is an event that occurs when Git is unable to resolve the differences in code between the two commits automatically.

Git is capable of automatically merging the changes only if the commits are on different lines or branches.



17. How do you resolve conflicts in Git?

Here are the steps that will help you resolve conflicts in Git:

- Identify the files responsible for the conflicts.
- Implement the desired changes to the files
- Add the files using the git add command.
- The last step is to commit the changes in the file with the help of the git commit command.

18. What is the functionality of git ls-tree?

The git ls-tree command is used to list the contents of a tree object.

19. What is the process to revert a commit that has already been pushed and made public?

There are two processes through which you can revert a commit:

1. Remove or fix the bad file in a new commit and push it to the remote repository. Then commit it to the remote repository using:

git commit –m “commit message”

2. Create a new commit to undo all the changes that were made in the bad commit. Use the following command:

git revert <commit id>

20. How is a bare repository different from the standard way of initializing a Git repository?

Standard way	Bare way
You create a working directory with the git init command.	Does not contain any working or checked out copy of source files.
A .git subfolder is created with all the git-related change history.	Bare repositories store git revision history in the root folder of your repository instead of the .git subfolder.

21. What does git clone do?

Git clone allows you to create a local copy of the remote GitHub repository. Once you clone a repo, you can make edits locally in your system rather than directly in the source files of the remote repo

22. What is Git stash?

Let’s say you're a developer and you want to switch branches to work on something else. The issue is you don’t want to make commits in uncompleted work, so you just want to get back to this point later. The solution here is the Git stash.

Git stash takes your modified tracked files and saves it on a stack of unfinished changes that you can reapply at any time. To go back to the work you can use the stash pop.

23. What does the git reset --mixed and git merge --abort commands do?

git reset --mixed is used to undo changes made in the working directory and staging area.

git merge --abort helps stop the merge process and return back to the state before the merging began.

24. What do you understand about the Staging area in Git?

The Staging Area in Git is when it starts to track and save the changes that occur in files. These saved changes reflect in the .git directory. Staging is an intermediate area that helps to format and review commits before their completion.

25. What is Git Bisect and how do you use it?

The Git Bisect command performs a binary search to detect the commit which introduced a bug or regression in the project's history.

Syntax: git bisect <subcommand> <options>

26. How do you find a list of files that has been changed in a particular commit?

The command to get a list of files that has been changed in a particular commit is:

git diff-tree –r {commit hash}

- -r flag allows the command to list individual files
- commit hash lists all the files that were changed or added in the commit.

27. What is the use of the git config command?

The git config command is used to set git configuration values on a global or local level. It alters the configuration options in your git installation. It is generally used to set your Git email, editor, and any aliases you want to use with the git command.

28. What is the functionality of git clean command?

The git clean command removes the untracked files from the working directory.

29. What is SubGit and why is it used?

SubGit is a tool that is used to migrate SVN to Git. It transforms the SVN repositories to Git and allows you to work on both systems concurrently. It auto-syncs the SVN with Git.

30. If you recover a deleted branch, what work is restored?

The files that were stashed and saved in the stashed index can be recovered. The files that were untracked will be lost. Hence, it's always a good idea to stage and commit your work or stash them.

Now let’s raise the level of difficulty with advanced Git interview questions and answers.

Interested to learn more about Git? Check out the DevOps Engineer Master's Program and get certified today.

Advanced Git Interview Questions

31. Explain the different points when a merge can enter a conflicted stage.

There are two stages when a merge can enter a conflicted stage.

1. Starting the merge process

If there are changes in the working directory of the stage area in the current project, the merge will fail to start. In this case, conflicts happen due to pending changes that need to be stabilized using different Git commands.

2. During the merge process

The failure during the merge process indicates that there’s a conflict between the local branch and the branch being merged. In this case, Git resolves as much as possible, but some things have to be fixed manually in the conflicted files.

32. What has to be run to squash the last N commits into a single commit?

In Git, squashing commits means combining two or more commits into one.

Use the below command to write a new commit message from the beginning.

```
git reset -soft HEAD~N &&git commit
```

But, if you want to edit a new commit message and add the existing commit messages, then you must extract the messages and pass them to Git commit.

The below command will help you achieve this:

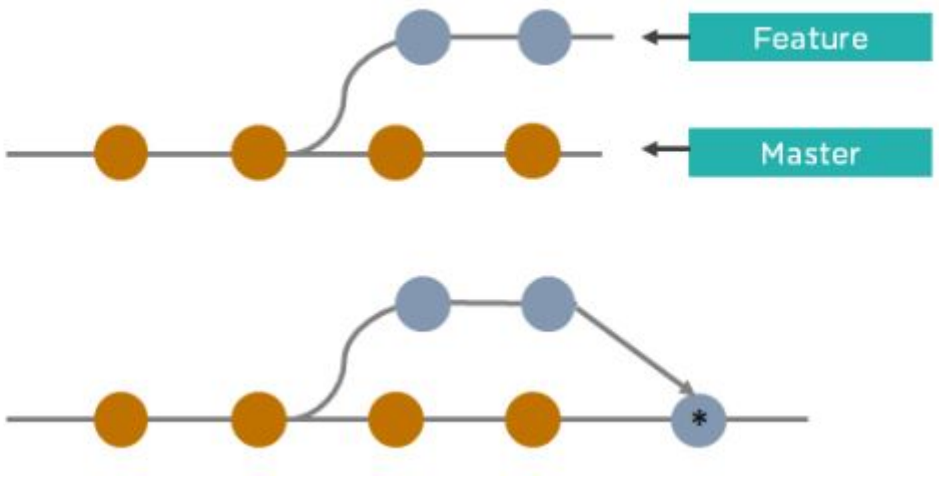
git reset -soft HEAD~N &&git commit -edit -m“$(git log -format=%B -reverse .HEAD@{N})”

33. What is the difference between fork, branch, and clone?

Fork	Branch	Clone
The fork is the process when a copy of the repository is made. It's usually experimentation in the project without affecting the original project. They’re used to advise changes or take inspiration from someone else’s project.	Git branches refer to individual projects within a git repository. If there are several branches in a repository, then each branch can have entirely different files and folders.	Git clone refers to creating a clone or a copy of an existing git repository in a new directory. Cloning automatically creates a connection that points back to the original repository, which makes it very easy to interact with the central repository.

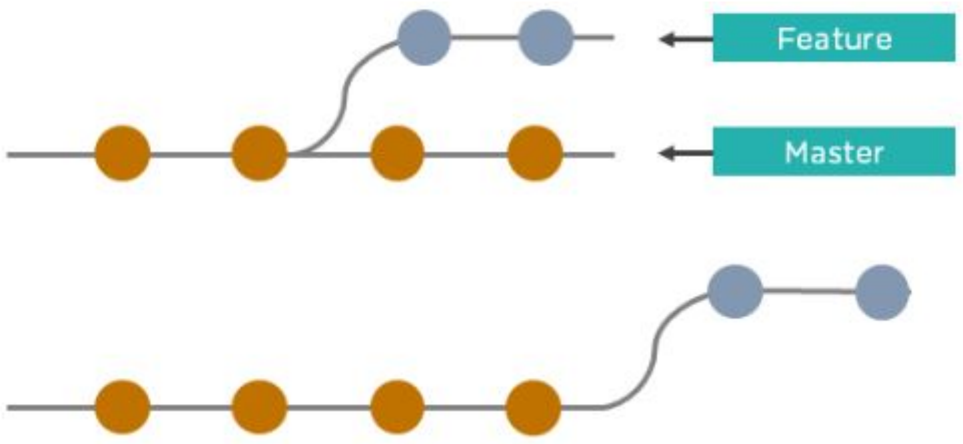
34. How is Git merge different from Git rebase?

Git merge is used to incorporate new commits into your feature branch.



- Git merge creates an extra merge commit every time you need to incorporate changes.
- It pollutes your feature branch history.

As an alternative to merging, you can rebase the feature branch into master.



- Git rebase Incorporates all the new commits in the master branch.
- It rewrites the project history by creating brand new commits for each commit in the original branch

35. What is the command used to fix a broken commit?

To fix a broken commit in Git, you may use the “git commit --amend” command, which helps you combine the staged changes with the previous commits instead of creating an entirely new commit.

36. How do you recover a deleted branch that was not merged?

To recover a deleted branch, first, you can use the git reflog command. It will list the local recorded logs for all the references. Then, you can identify the history stamp and recover it using the git checkout command.

37. What is git stash drop?

The Git stash drop command is used to remove a particular stash. If there’s a stash you're no longer using or you want to remove a specific item of stash from the list, you can use the stash commands.

Let’s say you want to delete an item named stash@{abc}; you can use the command:

git stash drop stash@{abc}.

38. What’s the difference between reverting and resetting?

Reverting	Resetting
The revert command in Git is used to create a new commit that undoes the changes made in the previous commit. When you use this command, a new history is	Git reset is a command that is used to undo the local changes that have been made to a Git

added to the project; the existing history is not modified.	repository. Git reset operates on the following: commit history, the staging index, and the working directory.

39. How can you discover if a branch has already been merged or not?

There are two commands to determine these two different things.

git branch --merged - Returns the list of branches that have been merged into the current branch.

git branch --no-merged - Returns the list of branches that have not been merged.

40. What is “git cherry-pick”?

The command git cherry-pick enables you to pick up commits from a branch within a repository and apply it to another branch. This command is useful to undo changes when any commit is accidentally made to the wrong branch. Then, you can switch to the correct branch and use this command to cherry-pick the commit.