Git

- **git config:**
  Usage: git config –global user.name “[name]” :
  Usage: git config –global user.email “[email address]”
  This command sets the author name and email address respectively to be used with your commits.
- **git init :** git init [repository name] : This command is used to start a new repository.
- **git clone:** git clone [url] : This command is used to obtain a repository from an existing URL.
- **git add :**
  git add [file] : This command adds a file to the staging area. ,
  git add * : This command adds one or more to the staging area.
- **git commit :**
  git commit -m “[Type in the commit message]” : This command records or snapshots the file permanently in the version history. ex: git commit -m "first line" ,
  git commit -a : This command commits any files you’ve added with the git add command and also commits any files you’ve changed since then.
- **git diff :**
  git diff : This command shows the file differences which are not yet staged.,
  git diff –staged : This command shows the differences between the files in the staging area and the latest version present.,
  git diff [first branch] [second branch] : This command shows the differences between the two branches mentioned.
- **git reset :**
  git reset [file] : This command unstages the file, but it preserves the file contents.,
  git reset [commit] :This command undoes all the commits after the specified commit and preserves the changes locally.
- **git status :** git status : This command lists all the files that have to be committed.
- **git rm :** git rm [file] : This command deletes the file from your working directory and stages the deletion.
- **git log :**
  git log : This command is used to list the version history for the current branch. ,
  git log –follow[file] : This command lists version history for a file, including the renaming of files also
- **git show :** git show [commitID] : This command shows the metadata and content changes of the specified commit.
- **git tag :** git tag [commitID] : This command is used to give tags to the specified commit.
- **git branch :**
  git branch : This command lists all the local branches in the current repository.
  git branch [branch name] : This command creates a new branch.
  git branch -d [branch name] This command deletes the feature branch.
- **git checkout**
  git checkout [branch name] This command is used to switch from one branch to another.
  git checkout -b [branch name] This command creates a new branch and also switches to it.
-

git merge : git merge [branch name] This command merges the specified branch's history into the current branch.

- **git remote :** git remote add [variable name] [Remote Server Link] This command is used to connect your local repository to the remote server.
- **git push :**

git push [variable name] master : This command sends the committed changes of master branch to your remote repository. ex: git push origin master

git push [variable name] [branch] : This command sends the branch commits to your remote repository. ex: git push origin master

git push –all [variable name] : This command pushes all branches to your remote repository. ex: git push -- all origin

git push [variable name] :[branch name] : This command deletes a branch on your remote repository. ex: git push origin : branch_2

- **git pull :** git pull [Repository Link] **:** This command fetches and merges changes on the remote server to your working directory.
- **git stash :** git stash save **:** This command temporarily stores all the modified tracked files.

git stash pop : This command restores the most recently stashed files.

git stash list : This command lists all stashed changesets.

git stash drop : This command discards the most recently stashed changeset.

1. What is Git?

Git is a Distributed Version Control system(DVCS)

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.

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.

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.

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 releas
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

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.

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.