Q U EST IO N S

1. What are the advantages of using Git?

Complete version history: Github preserves all changes/files in the source code which allows reproducibility of older versions

Branching & Merging: With the branch setting different users can work simultaneously on the same project and/or a user can test different code/algorithms in a project simultaneously

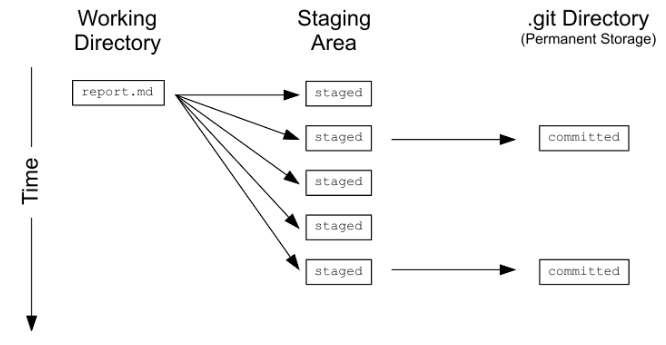
Merging allows to bring the work from different branches and users together.

1. What language is used in Git?

C is the programming language that is used for creating Git which ensures that the overheads are reduced.

1. What is the meaning of “Index” or “Staging Area” in Git?

The **Git index** is used as a staging area between your working directory and your repository. You can use the **index** to build up a set of changes that you want to commit together. When you create a commit, what is committed is what is currently in the **index**, not what is in your working directory.



1. What is the process for creating a repository in Git?

A git repository can be created either by the git web UI by selecting create repository option

Or we can use linux commands

1. What is ‘head’ in Git and how many heads can be created in a repository?

Head is the reference to a commit object. Master is the default head in a repository and each repository can have multiple heads

1. Why do we need branching in Git?

Branching in Git allows us to jump between different versions (branches) of a program allowing us to test different functionalities of a program and/or divide a project in parallel tasks.

1. Write a way to create a new branch in Git?

Git merge or git pull creates a new feature into the main branch

1. How do you define a ‘conflict’ in Git?

If you want to merge a commit there is a change in one place and same change already exists then while merging the Git will not be able to predict which is the change that needs to be taken precedence.

1. How to resolve a conflict in Git?

If you want to resolve a conflict in Git then you need to edit the files for fixing the conflicting changes and then you can run “git add” to add the resolved files and after that you can run the ‘git commit’ for committing the repaired merge.

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

Git config is used to specify git configuration settings. For example for setting up uer name and email

$ git config --global user.name "John Doe"

$ git config --global user.email johndoe@example.com

1. What is Git fork?

Git Fork means you just create a copy of the main repository of a project source code to your own GitHub profile. Here you can experiment whatever you like without affecting the main source of that project. The fork is mostly used to indicate or propose any changes to the source project or create our new idea using that project source as a starting point. Then make your changes and create a Pull Request to the main repository branch. If the Main Repository owners like your changes they will merge it to the main repository.

Why we need Fork?

You may don’t have the write permission to work directly on the main repository. (Mostly this model is used for open source projects).

If everyone clone and directly work on that main project repository/ branch then it’ll be very hard to manage.

1. Difference between fork, branch and clone?

When you fork a repository, you create a copy of the original repository (upstream repository) but the repository remains on your GitHub account. Whereas, when you clone a repository, the repository is copied on to your local machine with the help of Git.

Changes made to the forked repository can be merged with the original repository via a pull request. Pull request knocks the repository owner and tell that “I have made some changes, please merge these changes to your repository if you like it”. On the other hand, changes made on the local machine (cloned repository) can be pushed to the upstream repository directly. For this, the user must have the write access for the repository otherwise this is not possible. If the user does not have the write access, the only way to go is through the forked request. So in that case, the changes made in the cloned repository are first pushed to the forked repository and then a pull request is created. It is a better option to fork before clone if the user is not declared as a contributor and it is a third party repository (not of the organization).

Forking is a concept while cloning is a process. Forking is just containing a separate copy of the repository and there is no command involved. Cloning is done through the command ‘git clone‘ and it is a process of receiving all the code files to the local machine.

<https://www.toolsqa.com/git/difference-between-git-clone-and-git-fork/#:~:text=When%20you%20fork%20a%20repository,with%20the%20help%20of%20Git.>

1. What's the difference between a "pull request" and a "branch"?

A branch is just a separate version of the code.

A pull request is when someone take the repo, makes their own branch, does some changes, then tries to merge that branch in (put their changes in the other person's code repository)

1. What is the difference between "git pull" and "git fetch"?

Git pull: fetches and copies the changes from remote repository to local repository

Git fetch: tells local repo to retrieve meta data from remote repo, but doesn’t copy any changes

1. How to revert previous commit in Git?

Git show HEAD retrieves the most recent commit, Git show Head~1 retrieves the commit before it

1. Explain the advantages of Forking Workflow

A crucial advantage of the Forking Workflow is that contributions can be integrated without even needing everybody to push to a single central repository that leads to clean project history. Developers can push to their own server-side repositories, but only the project maintainer can push to the official repository.  
  
If developers are ready to publish a local commit, then they push the commit to their own public repository and not the official one. After this, they go for a pull request with the main repository that lets the project maintainer know an update is ready to be integrated

1. Difference between HEAD, working tree and index, in Git?

**Working trees:**They are nothing but the files that you are currently working on.

**HEAD:** HEAD is a pointer to the branch or commit that you last checked out, and which will be the parent of a new commit if you make it. For example, if you're on the master branch, then HEAD will be pointing to master, and when you commit, that new commit will be a descendant of the revision that your master pointed to, and master will be updated to point to the new commit.

**Index:** The git "index" is where you place files you want commit to the git repository.The index is a staging area where the new commit is prepared. Essentially, the contents of the index are what will go into the new commit (though if you do git commit -a, this will automatically add all changes to files that [Git](https://intellipaat.com/blog/what-is-git/) knows about to the index before committing, so it will commit the current contents of your working tree). The command git add will add or update files from the working tree into your index.

1. How to identify if a certain branch has been merged into master?

git branch --merged master lists branches merged into *master*

git branch --merged lists branches merged into *HEAD* (i.e. tip of current branch)

git branch --no-merged lists branches that have not been merged

1. What is the use of a Git clone?

To **clone** a **repository** means to duplicate and download everything in the **repository**. It will duplicate and download the entire **repository**.

1. What is Git stash?

git-stash - Stash the changes in a dirty working directory away

<https://git-scm.com/docs/git-stash>

1. When should I use "git stash"?

Use git stash when you want to record the current state of the working directory and the index, but want to go back to a clean working directory. The command saves your local modifications away and reverts the working directory to match the HEAD commit.

1. What is Git stash drop?

Git stash drop: Remove a single stash entry from the list of stash entries.

1. What is Git stash save?

This option is deprecated in favour of **git stash push**. It differs from "stash push" in that it cannot take pathspec. Instead, all non-option arguments are concatenated to form the stash message.

1. What README.MD ? What is its purpose? What does MD stands for?

README.MS is an html file in a repository and/or project that provides the summary of the peoject and lists any instructions on how to operate the written program.

MD stands for markdown

1. How to create repository from command prompt?

touch README.md

git init

git add README.md

git commit -m "first commit"

git remote add origin git@github.com:alexpchin/<reponame>.git

git push -u origin master

1. What is the function of ‘git checkout’ in Git?

Updates files in the working tree to match the version in the index or the specified tree. If no pathspec was given, **git checkout** will also update HEAD to set the specified branch as the current branch.

1. How can you bring a new feature in the main branch?
2. What is the function of ‘git rm’?

**git rm** is used to remove a file from a **Git** repository. It is a convenience method that combines the effect of the default shell **rm** command with **git** add .

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

In order to **apply** your **Git stash** to your current working directory, use the “**git stash apply**” command and specify the **stash** you want to **apply**. If you don't specify any arguments to the **apply** command, the top of the stack will be applied.

30. What is the use of ‘git log’?

The command git log is used to view the **log** of the project's history. Log entries are shown most recent first. When you run git log, Git automatically uses a pager to show one screen of output at a time. Press the space bar to go down a page or the 'q' key to quit.

git log path, where path is the path to a specific file or directory. The log for a file shows changes made to that file; the log for a directory shows when files were added or deleted in that directory, rather than when the contents of the directory's files were changed.

31. What is ‘git add’ is used for?

To add a file to the staging area, use git add filename

32. What is 'git diff' is used for?

In order to compare the file as it currently is to what you last saved, you can use git diff filename. git diff without any filenames will show you all the changes in your repository, while git diff directory will show you the changes to the files in some directory.

33. What is ‘git status’ is used for?

When you are using Git, you will frequently want to check the **status** of your repository. To do this, run the command git status, which displays a list of the files that have been modified since the last time changes were saved.

34. Can we create multiple branch with one command?

for branch in alpha{1..3}; do git checkout -b $branch; done; git push origin --all

35. what is the command that is used to delete a branch?

Local: Delete a branch with git branch -d <branch>.

For example: git branch -d fix/authentication

The -d option will delete the branch only if it has already been pushed and merged with the remote branch. Use -D instead if you want to force the branch to be deleted, even if it hasn't been pushed or merged yet.

Remote: git push <remote> --delete <branch>.

For example: git push origin --delete fix/authentication

The branch is now deleted remotely.

You can also use this shorter command to delete a branch remotely: git push <remote> :<branch>

For example: git push origin :fix/authentication

If you get the error below, it may mean that someone else has already deleted the branch.

36. What is another option for merging in git?

Rebase as an **Alternative** to **Merge**. While **merging** is definitely the easiest and most common way to integrate changes, it's not the only one: "Rebase" is an **alternative** means of integration.37. How to remove a file from git without removing it from your file system?

38. Use of "git rebase" instead of "git merge"?

As an alternative to merging, you can rebase the feature branch onto master branch using the following commands:

git checkout feature

git rebase master

This moves the entire feature branch to begin on the tip of the master branch, effectively incorporating all of the new commits in master. But, instead of using a merge commit, rebasing re-writes the project history by creating brand new commits for each commit in the original branch.

The major benefit of rebasing is that you get a much cleaner project history. First, it eliminates the unnecessary merge commits required by git merge. Second, as you can see in the above diagram, rebasing also results in a perfectly linear project history—you can follow the tip of feature all the way to the beginning of the project without any forks. This makes it easier to navigate your project with commands like git log, git bisect, and gitk.

But, there are two trade-offs for this pristine commit history: safety and traceability. If you don’t follow the [Golden Rule of Rebasing](https://www.atlassian.com/git/tutorials/merging-vs-rebasing#the-golden-rule-of-rebasing), re-writing project history can be potentially catastrophic for your collaboration workflow. And, less importantly, rebasing loses the context provided by a merge commit—you can’t see when upstream changes were incorporated into the feature.

Source: <https://www.atlassian.com/git/tutorials/merging-vs-rebasing#:~:text=The%20Rebase%20Option&text=But%2C%20instead%20of%20using%20a,commits%20required%20by%20git%20merge%20.>

39. What is a repository in Git?

A Git repository is the .git/ folder inside a project. This repository tracks all changes made to files in your project, building a history over time. Meaning, if you delete the .git/ folder, then you delete your project’s history.

40. Command used to write a commit message?

Commit -m

41. What does commit object contain?

The commit object contains the directory tree object hash, parent commit hash, author, committer, date and message.

42. Write one use-case of Github?

Collaborating on a project working from different machines

43. Name some alternative of Git?

Bitbucket, GitLab, Subversion (SVN), Perforce

44. What is a gist in Git?

A **gist** can be a string of code, a bash script or some other small piece of data. These bits of information are hosted by GitHub as a repository. Using **gist** has all the benefits of utilizing a GitHub repository, and also provides a more lightweight way of utilizing GitHub's versioning features

45. What is a gist programming?

46. Name any two Git repository hosting services which are common?

Gitlab

SourceForge

Misc.

