**Git and GitHub**

**(1) What is the difference between Git and SVN ?**

|  |  |
| --- | --- |
| **Git** | **SVN** |
| Git is a Decentralized Version Control tool | SVN is a  Centralized Version Control tool |
| It belongs to the 3rd generation of Version Control tools | It belongs to the 2nd generation of Version Control tools |
| Clients can clone entire repositories on their local systems | Version history is stored on a server-side repository |
| Commits are possible even if offline | Only online commits are allowed |
| Push/pull operations are faster | Push/pull operations are slower |
| Works are shared automatically by commit | Nothing is shared automatically |

### ****(2) What is a distributed VCS?****

* These are the systems that don’t rely on a central server to store a project file and all its versions.
* In Distributed VCS, every contributor can get a local copy or “clone” of the main repository.

### ****(3) What is the difference between Git and GitHub?****

Git is a version control system of distributed nature that is used to track changes in source code during software development.

GitHub is a Git repository hosting service, plus it adds many of its own features. GitHub provides a Web-based graphical interface.

### ****(4) What are the benefits of using Version Control System?****

With the Version Control System(VCS), all the team members are allowed to work freely on any file at any time. VCS gives the flexibility to merge all the changes into a common version.

### ****(5) What language is used in Git?****

Git uses ‘C’ language. GIT is fast, and ‘C’ language makes this possible by reducing the overhead of run times associated with high-level languages.

**(6) What is commit massage ?**

Git commit -m “[descriptive message]” *commit your staged content as a new commit snapshot*

Commit message is a component of git which shows up when you submit a change. Git gives you a content tool where you can enter the adjustments made to a commit

**(7) Mention the various Git repository hosting functions ?**

* GitHub
* Git lab
* Bit bucket
* Source Forge
* Git Enterprise

### ****(8) What is a repository in Git ?****

Repository in Git is a place where Git stores all the files. Git can store the files either on the local repository or on the remote repository.

### ****(9) How can you create a repository in Git ?****

To create a repository, create a directory for the project if it does not exist, then run the command “***git init***”. By running this command .git directory will be created in the project directory.

### ****(10) What is a ‘conflict’ in git ?****

A 'conflict' appears when the commit that has to be combined has some change in one place, and the current act also has a change at the same place. Git will not be easy to predict which change should take precedence.

**(11) How is git instaweb used ?**

‘git instaweb’ is used to automatically direct a web browser and run a webserver with an interface into your local repository.

**(12) What is git is-tree ?**

‘git is-tree’ represents a tree object including the mode and the name of each item and the SHA-1 value of the blob or the tree.

**(13) What is the difference between git pull and git fetch?**

Git pull command pulls innovation or commits from a specific branch from your central repository and updates your object branch in your local repository.

Git fetch is also used for the same objective, but it works in a slightly different method. When you behave a git fetch, it pulls all new commits from the desired branch and saves it in a new branch in your local repository

**Git pull = git fetch + git merge**

**15) What is the purpose of the git clone?**

The git clone command generates a copy of a current Git repository. To get the copy of a central repository, 'cloning' is the simplest way used by programmers.

**(16) What is git pull origin?**

pull is a get and a consolidation. 'git pull origin master' brings submits from the master branch of the source remote (into the local origin/master branch), and then it combines origin/master into the branch you currently have looked out.

**(17) What does git commit a ?**

Git commits "records changes to the storehouse" while git push " updates remote refs along with contained objects" So the first one is used in a network with your local repository, while the latter one is used to communicate with a remote repository.

**(18) Why GIT better than Subversion ?**

GIT is an open source version control framework; it will enable you to run 'adaptations' of a task, which demonstrate the changes that were made to the code over time also it allows you keep the backtrack if vital and fix those changes. Multiple developers can check out, and transfer changes, and each change can then be attributed to a particular developer.

**(19) What does 'hooks' comprise of in Git?**

This index comprises of Shell contents which are enacted after running the relating git commands. For instance, Git will attempt to execute the post-commit content after you run a commit.

**(20) Why is it desirable to create an additional commit rather than amending an existing commit ?**

There are couples of reason:-

1. The correct activity will devastate the express that was recently saved in a commit. If only the commit message gets changed, that's not a problem. But if the contents are being modified, chances of excluding something important remains more.
2. Abusing "git commit- amends" can cause a small commit to increase and acquire inappropriate changes.

**(21) In Git, how would you return a commit that has just been pushed and made open ?**

There can be two answers to this question and ensure that you incorporate both because any of the below choices can be utilized relying upon the circumstance:

Remove or fix the bad document in another commit and push it to the remote repository. This is a unique approach to correct a mistake. Once you have necessary changes to the record, commit it to the remote repository for that I will utilize

**git submit - m "commit message."**

Make another commit that fixes all changes that were made in the terrible commit. to do this, I will utilize a command

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

**(22) What does the committed item contain ?**

Commit item contains the following parts; you should specify all the three present below:

A set of records, representing to the condition of a task at a given purpose of time

References to parent commit objects

An SHAI name, a 40 character string that uniquely distinguishes the commit object.

**(24) How might you fix a messed up submit ?**

To fix any messed up commit, you will utilize the order "git commit?correct." By running this direction, you can set the wrecked commit message in the editor.

**(25) Describing branching systems you have utilized ?**

This question is a challenge to test your branching knowledge with Git along these lines, inform them regarding how you have utilized branching in your past activity and what reason does it serves, you can refer the below mention points:

**Feature Branching:**

A component branch model keeps the majority of the changes for a specific element within a branch. At the point when the item is throughout tested and approved by automated tests, the branch is then converged into master.

**Task Branching:**

In this model, each assignment is actualized on its branch with the undertaking key included in the branch name. It is anything but difficult to see which code actualizes which task, search for the task key in the branch name.

**Release Branching:**

Once the create branch has procured enough features for a discharge, you can clone that branch to frame a Release branch. Making this branch begins the following discharge cycle so that no new features can be included after this point, just bug fixes, documentation age, and other release oriented assignments ought to go in this branch. When it is prepared to deliver, the release gets converged into master and labeled with a form number. Likewise, it should be converged once again into creating a branch, which may have advanced since the release was started.

At last, disclose to them that branching methodologies fluctuate starting with one association then onto the next, so I realize essential branching activities like delete, merge, checking out a branch, etc.

**(24) By what method will you know in Git if a branch has just been combined into master ?**

The appropriate response is immediate.

To know whether a branch has been merged into master or not you can utilize the below commands:

**git branch - merged** It records the branches that have been merged into the present branch.

**git branch - no merged** It records the branches that have not been merged.

**(26) Mention some of the best graphical GIT customers for LINUX ?**

Some of the best GIT customer for LINUX is

1. Git Cola
2. Smart git
3. Git-g
4. Git GUI
5. Giggle
6. qGit

**(27) What is Subgit ? Why use it ?**

'Subgit' is a tool that migrates SVN to Git. It is a stable and stress-free migration. Subgit is one of the solutions for a company-wide migration from SVN to Git that is:

1. It is much superior to git-svn
2. No need to change the infrastructure that is already placed.
3. It allows using all git and all sub-version features.
4. It provides stress ? free migration experience.

**(28) What is the function of 'GIT PUSH' in GIT ?**

'GIT PUSH' updates remote refs along with related objects.

### (29) What is 'bare repository' in Git ?

A "bare" repository in Git includes the version control information and no working files (no tree), and it doesn’t include the special. git sub-directory. Instead, it consists of all the contents of the .git sub-directory directly in the main directory itself, whereas working list comprises of:

1. A .git subdirectory with all the Git associated revision history of your repo.
2. A working tree, or find out copies of your project files.

### (30) What is the purpose of GIT stash ?

GIT stash takes the present state of the working file and index and puts in on the stack for next and gives you back a clean working file. So in case if you are in the middle of object and require to jump over to the other task, and at the same time you don't want to lose your current edits, you can use GIT stash.

**(31) Name a few Git commands and explain their usage ?**

Below are some basic Git commands:

|  |  |
| --- | --- |
| Command | Function |
| git rm [file] | deletes the file from your working directory and stages the deletion. |
| git log | list the version history for the current branch. |
| git show [commit] | shows the metadata and content changes of the specified commit. |
| git tag [commitID] | used to give tags to the specified commit. |
| git checkout [branch name]  git checkout -b [branch name] | used to switch from one branch to another.  creates a new branch and also switches to it. |

### (32) What is GIT stash drop ?

When you are done with the stashed element or want to delete it from the directory, run the git 'stash drop' command. It will delete the last added stash item by default, and it can also remove a specific topic if you include as an argument.

**(33) What are the advantages of using GIT ?**

Here are some of the essential advantages of Git :

* Data repetition and data replication is possible
* It is a much applicable service
* For one depository you can have only one directory of Git
* The network performance and disk application are excellent
* It is effortless to collaborate on any project
* You can work on any plan within the Git

**(34) Why do we require branching in GIT ?**

With the help of branching, you can keep your branch, and you can also jump between the different branches. You can go to your past work while at the same time keeping your recent work intact.

**(35) What is the purpose of 'git config' ?**

The 'Git config' is a great method to configure your choice for the Git installation. Using this command, you can describe the repository behavior, preferences, and user information.

**(36) What is the definition of "Index" or "Staging Area" in GIT ?**

When you are making the commits, you can make innovation to it, format it and review it in the common area known as 'Staging Area' or 'Index'.