What Is Git and Why Is It Used?

Git is an open-distributed version control system that allows developers to track changes to their codebase and collaborate on projects with other developers. Git is typically used for software development, but it can be used for any type of file.

When a developer makes a change to a code file, they can commit that change to their local Git repository. Then, they can push their changes to a remote Git repository, such as a server hosted by a company or an open-source project. Other developers can then pull the changes down from the remote repository and incorporate them into their code.

And an important thing about Git is that it is a free source and open to anyone.

### **1. What is Git?**

[Git](https://www.simplilearn.com/tutorials/git-tutorial/what-is-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 is GitHub?**

   To provide Internet hosting for version control and software development, GitHub makes use of Git.

### **4. Mention some popular Git hosting services.**

* GitHub
* SourceForge
* GitLab
* Bitbucket

### **5. Different types of version control systems**

* Local version control systems have a database that stores all file changes under revision control on disc in a special format.
* Centralized version control systems have a single repository, from which each user receives their working copy.
* Distributed version control systems contain multiple repositories, and different users can access each one with their working copy.

### **6. What benefits come with using GIT?**

* Data replication and redundancy are both possible.
* It is a service with high availability.
* There can only be one Git directory per repository.
* Excellent network and disc performance are achieved.
* On any project, collaboration is very simple

### **7. What’s the difference between**[**Git and GitHub**](https://www.simplilearn.com/tutorials/git-tutorial/git-vs-github)**?**

|  |  |
| --- | --- |
| Git | GitHub |
| Git is a software | GitHub is a service |
| [Git can be installed](https://www.simplilearn.com/tutorials/git-tutorial/git-installation-on-windows) 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 |

### **8. 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.



### **9. 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.



### **10. How is Git different from Subversion (SVN)?**

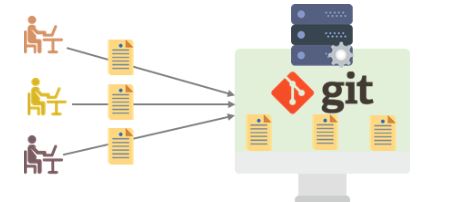
|  |  |
| --- | --- |
| GIT | SVN |
| Git is a distributed decentralized version control system | SVN is a centralized version control system. |
| Git stores content in the form of metadata. | SVN stored data in the form of files. |
| The master contains the latest stable release. | In SVN, the trunk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

### **11. 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

### **12. What are the advantages of using Git?**

* Faster release cycles
* Easy team collaboration
* Widespread acceptance
* Maintains the integrity of source code
* [Pull requests](https://www.simplilearn.com/tutorials/git-tutorial/git-pull-request)



### **13. 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](https://www.simplilearn.com/c-programming-article) reduces the overhead of run times, which are common in high-level languages.

### **14. What is the correct syntax to add a message to a commit?**

 git commit -m "x files created"

### **15. Which command is used to create an empty Git repository?**

git init - This [command](https://www.simplilearn.com/tutorials/git-tutorial/git-commands) helps to create an empty repository while working on a project.

### **16. 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.

### **17.  What does the git push command do?**

The [Git push command](https://www.simplilearn.com/tutorials/git-tutorial/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.



### **18. 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 |

### **19. GitHub, GitLab and Bitbucket are examples of git repository \_\_\_\_\_\_\_ function?**

hosting. All the three are services for hosting Git repositories

### **20. What do you understand about the Git merge conflict?**

A [Git merge conflict](https://www.simplilearn.com/tutorials/git-tutorial/merge-conflicts-in-git) 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.



### **21. 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.

### **23. 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>

### **24. 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

### **25. Explain these commands one by one– git status, git log, git diff, git revert <commit>,  git reset <file>.**

* Git status - It shows the current status of the working directory and the staging area.
* Git revert<commit> -  It is used for undoing changes to a repository's commit history.
* Git log- It is a key tool for reviewing and reading the history of everything that happens to a repository.
* Git diff- It is a multi-purpose Git command that performs a diff function on Git data sources when executed.
* Git reset<file>- it is used to unstage a file

### **26. What exactly is forking in Git?**

It is a repository duplicate and forking allows one to experiment with changes without being concerned about the original project

### **27. How does Git work?**

Git works by tracking changes to files in a project and allowing developers to easily revert to previous versions if necessary. Git also makes it easy to collaborate on projects, as it allows multiple developers to work on the same codebase simultaneously.

### **28. What are some of the most popular Git commands?**

Some of the most popular Git commands are,

* "git init" (which initializes a Git repository)
* "git add" (which adds files to a Git repository)
* And "git commit" (which saves changes to a Git repository)

### **29. What is the difference between Git and other revision control systems?**

* Git is a distributed revision control system, which means that it can be used without a central server. This allows for a great deal of flexibility in how projects are managed.
* On the other side, revision control systems are often centralized, which can limit the flexibility of how projects are managed

### **30. How do I install Git?**

Installing Git is simple. Just download the latest version from the Git website (<https://git-scm.com/>)

### **31. What are some of the most important features of Git?**

Some of the most important features of Git are its distributed nature, its ability to track changes, and its support for branches.

* The distributed nature of Git allows developers to work independently and offline.
* The ability to track changes helps developers to keep track of their work and revert to previous versions if necessary.
* The support for branches allows developers to experiment with new features without affecting the main codebase

### **32. What differentiates between the commands git remote and git clone?**

The main difference between the git remote and git clone commands is that the git remote adds a remote repository as a shortcut to your current repository, while the git clone creates an entirely new copy of a remote repository.

### **33. Tell me the difference between git pull and git fetch?**

Both of these commands will fetch any new commits from the remote repository, but they differ in how they handle these commits.

Git pull will merge the remote commits into the current branch, while git fetch will simply retrieve the commits and store them in the local repository. This means that if you have any uncommitted changes, git pull may result in merge conflicts, while git fetch will not

### **34. Is Git and GitHub the same thing?**

No, Git and GitHub are two different things.

* Git is a version control system that lets you track changes to your code.
* GitHub is a hosting service for Git repositories. You can use GitHub to store your code remotely, or you can use it to collaborate with other developers on a project

### **35. 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. |