1 . What are the advantages of using Git ?

Answer :

1. Performance :

Git performs very strongly and reliably when compared to other version control systems. New code changes can be easily commited, version branches can be effortlessly compared and merged, and code can also be optimized to perform better.

1. Security:

Git is designed specially to maintain the integrity of source code. File contents as well as the relationship between file and directories, tags, commits, versions etc. are secured cryptographically using an algorithm called SHA1 which protects the code and change history against accidental as well as malicious damage. You can be sure to have an authentic content history for your soruce code with Git.

1. Flexibility:

A key design objective of Git is the kind of flexibility it offers to support several kinds of nonlinear development workflows and its efficiency in handling both small scale and large scale projects as well as protocols. It is uniquely designed to support tagging and branching operations and store each and every activity carried out by the user as an integral part of "change" history. Not all VCSs support this feature.

1. Wide acceptance:

Git offers the type of performance, functionality, security and flexibility that most developers and teams need to develop their projects. When compared to other VCS Git is most widely accepted system owing to its universally accepted usability and performance standards.

1. Quality Open Source Project:

Git is a widely supported open source project with over ten years of operational history. People maintaining the project are very well matured and possess a long term vision to meet the long term needs of users by releasing staged upgrades at regular intervals of time to improve functionality as well as usability.

2. What language is used in Git ?

Answer:

Written in C, TCL, PERL, PYTHON

3. What is the meaning of index or staging Area in Git ?

Answer:

While working on a project, if we make any changes then we are dealing with our project’s working directory. This project directory is on our computer’s file system. All the changes we make remain in the working directory until we add them to the staging area via git add command. The staging area is best described as a preview of our next commit. Meaning when we do a git commit, git will take the changes that are in staging area and make a new commit out of those changes.

One practical use of the staging area is that it allows us to fine-tune our commits. We can add and remove changes from staging area until we are satisfied with how our next commit will look like, at which point we can do git commit. And after we commit our changes they go into .git/objects directory where they are saved as commit, blob and tree objects.

4. What is the process of creating a repository in Git ?

Answer:

1. In the upper-right corner of any page, click , and then click **New repository**.
2. Type a short, memorable name for your repository. For example, "Parth Repo".
3. Optionally, add a description of your repository. For example, "My repository on GitHub."
4. Choose to make the repository either public or private.
5. Select **Initialize this repository with a README**.
6. Click **Create repository**.

5. What is head in github and how many heads can be created in github?

Answer:

Head is the current branch. It is a symbolic reference to a branch. HEAD is always there and it’s pointing to one of those these other pointers, to one of the branches that we are currently working on. It is basically the parent of our next commit and it is what should be what was last checked-out into our working directory.

6. Why do we need branching in Git ?

Answer:

Using branches help you organize the workflow more efficiently and rather effortlessly. Let’s say you are building a software for a company in team. It would be a good idea to have a branch for each developer because usually they work on different things. And because of the way Git works, you can keep working on your branch regardless of the work that is happening in other branches.

7. Write a way to create a new branch in Git.

Answer:

$ git checkout -b mybranch

$ git push origin mybranch

To see all the branches we can use :

$ git branch -a

8. How do you define a conflict in Git ?

Answer:

A merge conflict usually occurs when your current branch and the branch you want to merge into the current branch have diverged. That is, you have commits in your current branch which are not in the other branch, and vice versa. Now, when Git merges the other branch into your current branch, it looks at the differences between the base commit and the current revision, and at the differences between the base commit and the other branch's latest commit. When there are unambiguous differences (i.e. only one side changed a certain piece of code), the changes are applied.

The merge conflicts occur when there are disagreeing changes. In that case, your conflicted file will have so-called conflict markers

9. How to resolve a conflict in Git ?

Answer:

We can use git mergetool, which opens a GUI that steps us through each conflict.

10. What is the function of git Config ?

Answer:

The git config command is a convenience function that is used to set Git configuration values on a global or local project level. These configuration levels correspond to .gitconfig text files. Executing git config will modify a configurationtext file.

11. What is git fork ?

Answer:

A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. Most commonly, forks are used to either propose changes to someone else's project or to use someone else's project as a starting point for your own idea.

12. Difference between fork, branch and clone?

Answer:

Fork creates a copy of the project hosted on your own GitHub account.

Clone" uses git software on your computer to download the source code and it's entire version history unto that computer.

13. What’s the difference between a pull request and branch ?

Answer:

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).

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

Answer:

**git fetch**really only downloads new data from a remote repository - but it doesn't integrate any of this new data into your working files. Fetch is great for getting a fresh view on all the things that happened in a remote repository.  
Due to it's "harmless" nature, you can rest assured: fetch will never manipulate, destroy, or screw up anything. This means you can never fetch often enough.

**git pull,** in contrast, is used with a different goal in mind - to update your current HEAD branch with the latest changes from the remote server. This means that pull not only downloads new data; it also directly**integrates**it into your current working copy files.

15. How to revert previous commit in Git?

Answer:

If you want to revert the last commit just do git revert <unwanted commit hash> , then you can push this new commit, which undid your previous commit.

16. Explain the advantage of forking workflow

Answer:

The main **advantage** of the **Forking Workflow** is that contributions can be integrated without the need for everybody to push to a single central repository. Developers push to their own server-side repositories, and only the project maintainer can push to the official repository.

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

Answer:

A single git repository can track an arbitrary number of branches, but your working tree is associated with just one of them (the "current" or "checked out" branch), and HEAD points to that branch."

My understanding is that a working tree is the dir and sub-dirs within it that contain the source files. It can be anywhere, but normally it is the same dir in which the hidden .git dir is located.

HEAD is the commit at the tip of the current branch. If you've just checked out the branch, i.e. have no modified files, then its content matches the working tree. As soon as you modify anything, it no longer matches.

18. How to identify if certain branch is merged into master?

Answer:

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

19. What is the use of git clone?

Answer:

git clone is a Git command line utility which is used to target an existing repository and create a clone, or copy of the target repository into our local machine.

20. What is git stash?

Answer:

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.

21. When should I use git stash?

Answer:

Same as above answer.

22. What is git stash drop?

Answer:

In case we do not need a specific stash, we use git stash drop command to remove it from the list of stashes. By default, this command removes to latest added stash.

23. What is git stash save ?

Answer:

Stashing takes the dirty state of 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.

24. what readme.md is? What is the purpose of it? What does .md stands for ?

Answer:

.md is markdown and readme.md is used to generate the html summary we see at the bottom of projects.

25. How to create repository from command line?

Answer:

* Git init
* Git add .
* Git commit

26. What is the function of git checkout ?

Answer:

git checkout, when used on branches, alters the target of the HEAD reference. It can be used to create branches, switch branches, and checkoutremote branches. The git checkout command is an essential tool for standard Git operation. It is a counterpart to git merge.

27. How can you bring a new feature in the main branch ?

Answer:

Merge your (now updated) master branch into your feature branch to update it with the latest changes .

28. what is the function of git rm?

Answer:

The git rm command can be used to remove individual files or a collection of files. The primary function of git rm is to remove tracked files from the Git index. Additionally, git rm can be used to remove files from both the staging index and the working directory.

29. what is the function of git stash apply?

Answer:

 You can reapply the one you just stashed by using the command.

30. What is the use of git log?

Answer:

Git logs allow you to review and read a history of everything that happens to a repository. The history is built using git-log , a simple tool with a ton of options for displaying commit history.

31. what is git add used for ?

Answer:

git add will promote pending changes from the working directory to the staging area.

32. What is git diff used for ?

Answer:

Diffing is a function that takes two input data sets and outputs the changes between them. git diff is a multi-use Git command that when executed runs a diff function on Git data sources. These data sources can be commits, branches, files and more.

33. What is git status used for ?

Answer:

The git status command displays the state of the working directory and the staging area. It lets you see which changes have been staged, which haven't, and which files aren't being tracked by Git.

34. Can we create multiple branch with one command?

Answer:

Yes we can.

35. What is the command used to delete a branch?

Answer:

Git branch -d branch\_name

36. What is the another option for merging in git ?

Answer:

We can use git rebase command for a different way of integration.

37. How to remove a file from git without removing it from the file system?

Answer:

Git rm -rf –cached $FILES

Then commit, push and then pull on remote repository.

38. Use of git rebase instead of git merge?

Answer:

Alternative of merging in git.

39. What is arepository in git ?

Answer:

The purpose of Git is to manage a project, or a set of files, as they change over time. Git stores this information in a data structure called a repository.

40. Command used to write a commit message?

Answer:

Git commit -m “commit message”

41. What does commit object contain ?

Answer:

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

42. Write one use- case of github?

Answer:

Used in my project of auction bidding, to work as a team.

43. Name some alternative of git.

Answer:

Bitbucket, sourceforge, allura, aws code commit etc..

44. what is a gist in Git?

Answer:

Gists are a great way to share your work. You can share single files, parts of files, or full applications. You can access gists at https://gist.github.com. Every gist is a Git repository, which means that it can be forked and cloned.

45. What is gist programming ?

Answer:

Gists are generally used as scratchpad / notepad, to write down small (or not so small) snippets of code or other information, written directly in the web interface, without the need to start a new project on your computer.

46. Name any two git repository hosting service which are common?

Answer:

I guess bit bitbucket and gitlab.