**GitHub Advantages:**

It makes easy to share open source Projects with your team

It provides great platform to share your work which you can showcase to your employers and recruiter

It has features

you can have a version history of your code so that previous versions are not lost with every iteration.

It supports 200 different types of Programming languages

you can have a version history of your code so that previous versions are not lost with every iteration.

**Languages used**

One can use any language since Git isn’t a programming environment

**Meaning of “Index” or “Staging area”**

Staging area is files that are going to be a part of the next commit, which lets git know what changes in the file are going to occur for the next commit

The staging area, that's what files are going to be a part of your next commit. It's how Git knows what is going to change between the current commit and the next one.

**What is the process for creating a repository in Git?**

In the upper-right corner of any page, use the drop-down menu, and select New repository.

Type a short, memorable name for your repository. For example, "hello-world".

Optionally, add a description of your repository. For example, "My first repository on GitHub."

Choose a repository visbility. For more information, see "[About repository visibility](https://docs.github.com/en/github/creating-cloning-and-archiving-repositories/about-repository-visibility)."

Select Initialize this repository with a README.

Click Create repository.

Thus, your repository is being created

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

The HEAD in Git is the **pointer** to the current branch reference, which is in turn a **pointer** to the last commit you made or the last commit that was checked out into your working directory.

Each head has a name (branch name or tag name, etc). By default, there is a head in every repository called master. A repository can contain any number of heads

**Why do we need branching in Git?**

Branches serve as an abstraction for the edit/stage/commit process.

Using branches help you organize the workflow more efficiently and rather effortlessly.

**Write a way to create a new branch in Git?**

$ git branch <new-branch>

**How do you define a ‘conflict’ in Git?**

A conflict arises when two separate branches have made edits to the same line in a file, or when a file has been deleted in one branch but edited in the other.

**How to resolve a conflict in Git?**

The most direct way to resolve a merge conflict is to edit the conflicted file.

Also there are different types of commands in git that can be used to resolve various conflicts

**What is the function of ‘git config’?**

he git config command is a convenience function that is used to set Git configuration values on different levels like, local, global and system.

**What is Git fork?**

 A fork is a copy of a repository

**Difference between fork, branch and clone?**

Difference between forking and cloning

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.

Difference between forking and branching

When you fork a repository, you’re creating your own copy of the entire project including the repo. A branch is simply a separate set of commits within a repo that already exists.

Forking is something you’ll typically do once when you start working on the project. Creating a branch is something you’ll do regularly if you are collaborating with anyone or if you are maintaining multiple releases simultaneously. It’s a way of keeping your work separate from the main branch until it is complete and tested

**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).

**What is the difference between "git pull" and "git fetch"?**

git fetch is the command that tells your local git to retrieve the latest meta-data info from the original (yet doesn’t do any file transferring. It’s more like just checking to see if there are any changes available).

git pull on the other hand does that AND brings (copy) those changes from the remote repository.

**How to revert previous commit in Git?**

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.

**Explain the advantages of Forking Workflow**

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.  This means that complete feature branches will be purposed for merge into the original project maintainer's repository.

**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

**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

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

We can use following commands for this purpose:

I. git branch --merged master: This prints the branches merged into master

II. git branch --merged lists: This prints the branches merged into HEAD (i.e. tip of current branch)

III. git branch --no-merged: This prints the branches that have not been merged

By default, this applies only to local branches.

We can use -a flag to show both local and remote branches.

Or we can use -r flag to show only the remote branches.

**What is the use of a Git clone?**

git clone is primarily used to point to an existing repo and make a clone or copy of that repo at in a new directory, at another location.

**What is Git stash? And when to use it?**

This command 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 (even on a different branch).

Often, when you’ve been working on part of your project, things are in a messy state and you want to switch branches for a bit to work on something else. The problem is, you don’t want to do a commit of half-done work just so you can get back to this point later. And this is when you use git stash.

**What is Git stash drop? And What is Git stash save?**

Git stash is a temporary storage. When you're ready to continue where you left off, you can restore the saved state easily.

The git stash command takes your uncommitted changes (both staged and upstaged), saves them away for later use, and then reverts them from your working copy.

**What README.MD? What is its purpose? What does MD stand for?**

README.MD denotes that the file is markdown formatted. Markdown is a markup language. With it you can easily display headers or have italic words, or bold or almost anything that can be done to text

**README**.**md is** used to generate the html summary you see at the bottom of projects. MD means “Markdown”

**How to create repository from command prompt?**

**•** Create a new branch and switch

• Switch between branches

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

The **git checkout** command lets you navigate between the branches created by **git** branch

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

Create branch with your feature:

git checkout -b $feature\_name

Write code. Commit changes:

git commit -am "My feature is ready"

Push your branch to GitLab:

git push origin $feature\_name

Review your code on commits page.

Create a merge request.

**What is the function of ‘git rm’?**

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.

**What is the function of ‘git stash apply’?**

git stash apply applies the changes and leaves a copy in the stash

**What is the use of ‘git log’?**

git log lists the commits made in that repository in reverse chronological order; that is, the most recent commits show up first.

**What is ‘git add’ is used for?**

The git add command adds a change in the working directory to the staging area. It tells Git that you want to include updates to a particular file in the next commit. However, git add doesn't really affect the repository in any significant way—changes are not actually recorded until you run git commit.

**What is 'git diff' is used for?**

Diff command is used in git to track the difference between the changes made on a file.

**What is ‘git status’ is used for?**

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

**Can we create multiple branch with one command?**

By using one command called as **a worktree**, you can create multiple branches running at the same time

**what is the command that is used to delete a branch?**

git branch -D my-branch

**What is another option for merging in git?**

Apart from using git merge command one can also use "git rebase" that offers another way to merge in git

**Use of "git rebase" instead of "git merge"?**

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.

**What is a repository in Git?**

Git repository is just a file location where you are storing all the files related to your project

**Command used to write a commit message?**

A shortcut command that immediately creates a commit with a passed commit message. By default, git commit will open up the locally configured text editor, and prompt for a commit message to be entered. Passing the -m option will forgo the text editor prompt in-favor of an inline message.

**What does commit object contain?**

A commit object contains the reference to another tree object and some other information (author, committer etc.)

**Write one use-case of Github?**

Github is a web-based platform used for version control. Git simplifies the process of working with other people and makes it easy to collaborate on projects. Team members can work on files and easily merge their changes in with the master branch of the project.

**Name some alternative of Git?**

* Fossil
* Bazaar
* GitCenter
* darcs.

**What is a gist in Git?**

Gist is an easy method to share snippets or excerpts of data with others. 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

**Name any two Git repository hosting services which are common?**

* Launchpad
* GitLab