* **Advantages of GitHub**

GitHub is the cloud platform where we can work simultaneously on multiple projects with inbuilt programming support. You can easily track your progress with project analysis dashboard. Altogether, it is compatible, easily accessible platform to maintain your projects.

* **Languages used in Git**

Git is the command platform where you can handle GitHub with commands. We can use Linux command.

* **“Index” or “Staging area”**

The index, or staging area, is where commits are prepared. The index compares the files in the working tree to the files in the repo. When you make a change in the working tree, the index marks the file as modified before it is committed.

* **Creating Repository -**

We need to select create new repository tab.

* **Head –**

HEAD is a reference to the last commit in the currently check-out branch. You can think of the HEAD as the "current branch". When you switch branches with git checkout, the HEAD revision changes to point to the tip of the new branch.

* **Branching –**

To organize workflow. You can create with the help of this command: $ git branch <new-branch>

* **Conflict/ Resolve Conflict -**

It happens when you merge branches that have competing commits, and Git needs your help to decide which changes to incorporate in the final merge. You can use conflict command to resolve merge conflict.

* **‘git config’ –**

The git config command is a convenience function that is used to set Git configuration values on a global or local project level.

* **Git fork –** It copy of a repository
* **"pull request" and a "branch" –**

A branch is separate subproject. 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 pull request" and a "branch" code repository).

* **"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 does not do any file transferring. It is 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.

* **Commit Revert -**

Use command *git revert <unwanted commit hash>* and push the new 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.

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

The -a flag will show both local and remote branches, and the -r flag shows only the remote branches. You can use the git merge-base command to find the latest common commit between the two branches. If that commit is the same as your branch head, then the branch has been completely merged

* **Git clone**

We can clone the repository.

* **Git Stash -**

When you Git stash or Git stash save, Git will create a Git commit object with some name and then save it in your repo. So, it means that you can view the list of stashes you made at any time. You can see the list of stashes made. And the most recent stash made is in the top.

* **README.MD**

README.md is used to generate the html summary you see at the bottom of projects. GitHub has their own flavor of Markdown.

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

1. Go into the directory containing the project.
2. Type git init.
3. Type git add to add all the relevant files.
4. You will probably want to **create** a. gitignore file right away, to indicate all the files you do not want to track. Use git add. Gitignore.
5. Type git commit.

* **‘git checkout’**

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?**

Command:

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.

* **‘git stash apply’**

It helps in changes and leaves a copy in the stash

* **‘git log’**

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

* **‘git add’ –**

To add a change in the working directory to the staging area.

* **'git diff' –**

To track the changes made on a file.

* **‘git status’ -** The state of the working directory and the staging area.
* **Multiple Branch Creation -**

We can use *Worktree*

* **Delete Branch –**

*git branch -D my-branch*

* **Merging Alternative -**

We can use *git rebase*

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

Rebase vs Merge. Rebasing and merging are both designed to integrate changes from one branch into another branch but in different ways. When you do rebase a feature branch onto master, you move the base of the feature branch to master branch's ending point. Merging adds a new commit to your history.

* **Repository –**

It is file in GitHub where you can name and store different subfiles over there.

* **Commit –**

Thecommit" command is used to save your changes to the local repository. It is reference to another tree object and some other information.

* **GitHub Usecase –**

Github can integrate with common platforms such as google cloud. You can engage the people with your technology with the help of giving access to read/write the code/project.

* **Alternative of Git -**

Gitbash, BitBucket, GitCenter

* **Gist -**

Every gist is a Git repository, which means that it can be forked and cloned. If you are signed in to GitHub when you create a gist, the gist will be associated with your account and you will see it in your list of gists when you navigate to your [gist home page](https://gist.github.com/).

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

BitBucket, GitLab