**Git and github part – 2 ( 20.09.23)**

**What is prune in fetch ?**

In Git, the fetch command is used to update your local repository with changes from a remote repository. When you run git fetch, Git will retrieve new branches, tags, and commits from the remote repository and store them locally so that you can review and merge them into your working branch if needed.

The prune option in git fetch is used to remove references to remote branches that no longer exist on the remote repository. When you fetch changes from a remote repository, Git fetches information about all branches on that remote, even if some of those branches have been deleted on the remote since your last fetch. By default, Git keeps these references locally, which can clutter your repository over time.

**What is squash and interactive interface of squash ?**

In the context of Git, "squash" refers to a technique used during the process of merging or rebasing commits to combine multiple commits into a single commit. Squashing is particularly useful when you have a series of smaller, incremental commits that you want to consolidate into a larger, more meaningful commit before pushing your changes to a remote repository. This can help maintain a cleaner and more organized commit history.

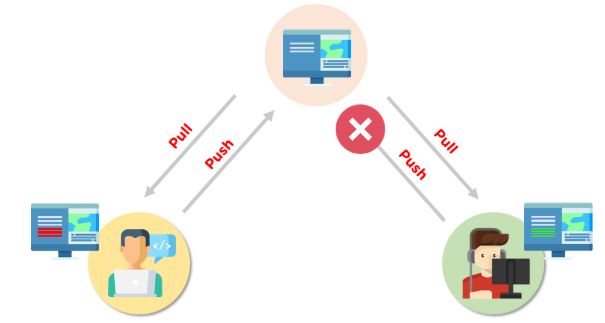
**What is merge conflicts and how to resolve it ?**

A merge conflict occurs in Git when you attempt to merge or pull changes from one branch into another, but Git is unable to automatically combine the changes due to conflicting edits in the same part of a file by different contributors. Merge conflicts often happen in collaborative development environments when multiple developers are working on the same codebase concurrently.

**What is a Git Merge Conflict?**

A git merge conflict is an event that takes place when Git is unable to automatically resolve differences in code between two commits. Git can merge the changes automatically only if the commits are on different lines or branches.

The following is an example of how a Git merge conflict works:



Let’s assume there are two developers: Developer A and Developer B. Both of them pull the same code file from the remote repository and try to make various amendments in that file. After making the changes, Developer A pushes the file back to the remote repository from his local repository. Now, when Developer B tries to push that file after making the changes from his end, he is unable to do so, as the file has already been changed in the remote repository.

To prevent such conflicts, developers work in separate isolated branches. The Git merge command combines separate branches and resolves any conflicting edits.

## How to Resolve Merge Conflicts in Git?

There are a few steps that could reduce the steps needed to resolve merge conflicts in Git.

Step 1: The easiest way to resolve a conflicted file is to open it and make any necessary changes.

Step 2: After editing the file, we can use the git add a command to stage the new merged content.

Step 3: The final step is to create a new commit with the help of the git commit command.

Step 4: Git will create a new merge commit to finalize the merge.

**What are pull request in github ?**

A pull request (PR) is a fundamental concept in version control systems, particularly in the context of collaborative software development using platforms like Git and GitHub. It is also known as a merge request in some other version control systems like GitLab.

**Creation:** When multiple developers are working on a software project, they often work in separate branches to develop new features, fix bugs, or make changes to the codebase. A pull request is created when a developer wants to propose changes made in their branch to be merged into another branch, usually the main or master branch.

**Request for Review:** The developer who creates the pull request requests that other team members or collaborators review their code changes. This review can include examining the code for errors, ensuring it meets coding standards, and assessing whether it aligns with the project's goals.

**Discussion:** Collaborators can leave comments, ask questions, or provide feedback on the changes made in the pull request. This discussion allows for a thorough review of the code, helping to catch and address any issues or improvements that need to be made.

**Approval:** After addressing feedback and ensuring the code meets the project's standards and requirements, collaborators can approve the pull request. This approval indicates that the proposed changes are ready to be merged into the target branch.

**Merge:** Once the pull request is approved, the changes are merged into the target branch (usually the main branch). This incorporates the new code into the project, and it becomes part of the official codebase.

**Closing:** After the changes are merged, the pull request is typically closed. Some platforms automatically close pull requests upon successful merging.

**What is forking ?**

Forking is a concept commonly associated with distributed version control systems like Git and is often used in collaborative software development. Forking refers to the process of creating a copy of a repository (usually an open-source project) in order to work on it independently. This copy, known as a fork, allows the forker to make changes, experiment, and develop new features without directly affecting the original repository.

**What is cloning ?**

Cloning is a term commonly associated with version control systems, particularly Git, and it refers to the process of creating a local copy of a remote repository. This local copy contains all the files, commit history, and branches from the remote repository, allowing you to work with the code locally on your computer.

**What is upstream url ?**

The term "upstream URL" typically refers to the URL of the original or primary source of a project or software package. This concept is often used in the context of package management systems, version control, and software distribution.

**git reset:** The git reset command is used to reset the current branch to a specific commit, which effectively moves the branch pointer to a different commit, making it the new commit where the branch starts.

**Git stash -** The git stash command is used to temporarily save changes in your working directory that you don't want to commit yet, allowing you to switch branches or perform other operations. Stashing is useful when you're in the middle of working on something, and you need to switch to a different branch or resolve an issue without committing your work.

**git remote add [alias] [url] -** add a git URL as an alias

**git fetch [alias] -** fetch down all the branches from that Git remote

**git merge [alias]/[branch] -** merge a remote branch into your current branch to bring it up to date

**git push [alias] [branch] -** Transmit local branch commits to the remote repository branch

**git pull -** fetch and merge any commits from the tracking remote branch