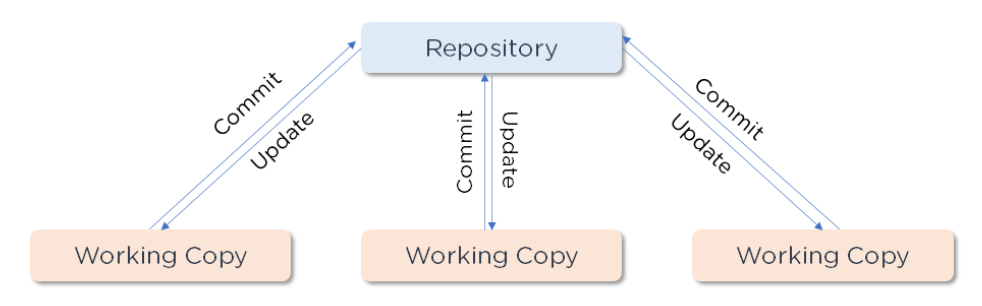
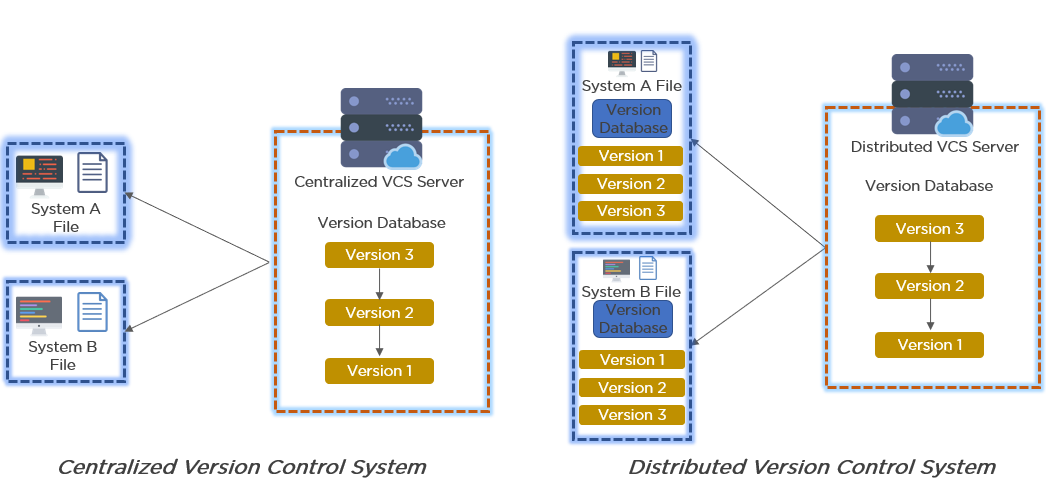
1. What is ‘Version Control System’?



A [version control system](https://www.simplilearn.com/tutorials/devops-tutorial/version-control) (VCS) is a program that records any changes to a file or set of data so that it is possible to restore it to a previous version if necessary. This guarantees that everyone on the team is working on the most up-to-date version of the file.

2. Differentiate Between Centralized and Distributed Version Control System



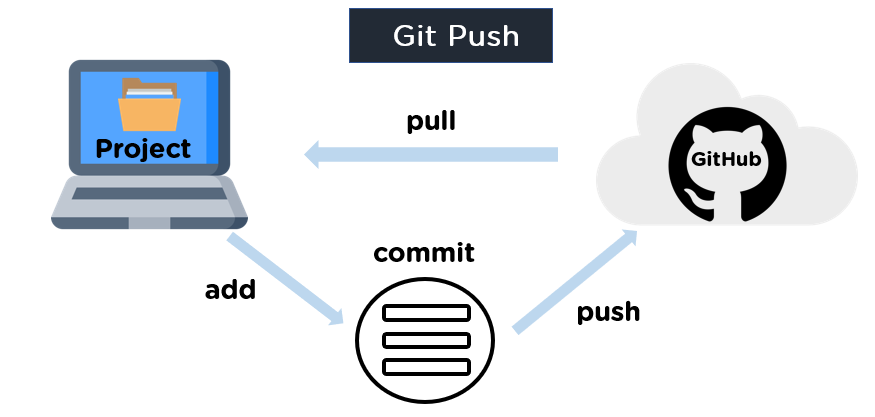
In a Centralized Version Control System:

* It stores all file versions on a central server.
* No developer has a complete copy of the local system's files.
* If the project's central server fails, you will lose all the project's data.

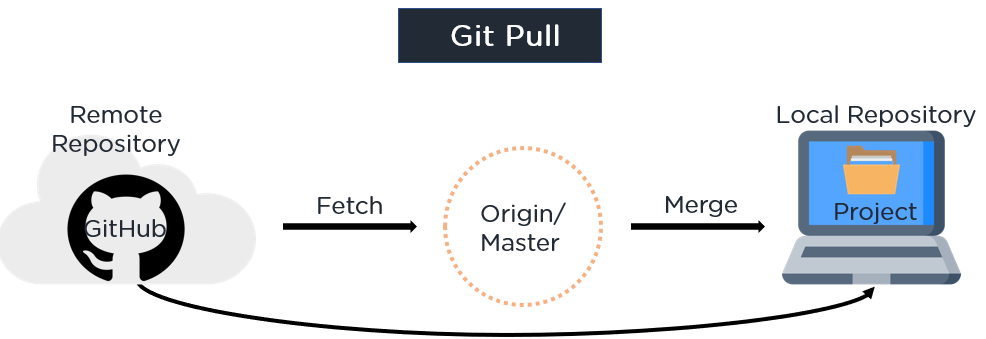
In a Distributed Version Control System:

* Every developer has a copy of all the code versions on their computer.
* Improves the ability to work offline and eliminates the need for a single backup location.
* Even if the server crashes, there is no danger.

### 3. Explain Git Push and Git Pull



[Git push](https://www.simplilearn.com/tutorials/git-tutorial/git-push-command) is a command that pushes the contents of a local repository to a remote repository. It runs a push after it has changed a local repository to share the changes with remote team members.



[Git pull](https://www.simplilearn.com/tutorials/git-tutorial/git-pull-request) is a command that pulls changes from a remote repository and merges them into the local repository. It's made up of two commands: git fetch followed by git merge.

4. Name a few Git Commands and function

* Git Config - Configure the username and email address
* Git init - Initialize a local Git repository
* Git Add - Add one or more files to the staging area
* Git Diff - View the changes made to the file
* Git Commit - Commit changes to the head but not to the remote repository
* Git reset - Undo local changes to the state of a Git repo
* Git Status - Displays the state of the working directory and staging area
* Git Merge - Merge a branch into an active branch
* Git Push - Upload content from the local repository to a remote repository
* Git Pull - Fetch and download content from a remote repository

### 5. Explain the Difference Between Git Pull and Git Fetch

#### Git Fetch

* It downloads only new data from a remote repository using Git fetch
* It does not include any of this new information in your working files
* To update the remote-tracking branches, run Git fetch at any time
* Command - git fetch origin

              git fetch –-all

#### Git Pull

* Git pulls new data and integrates it with the current working files, updating the current HEAD branch with the latest modifications from the remote server
* It attempts to combine remote modifications with those made locally
* Command - git pull origin master

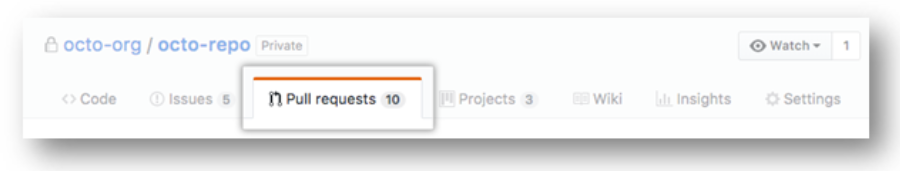
### 6. What is a Merge Conflict in Git and how can it be resolved?

When you have merging branches with opposing commits, a [merge conflict](https://www.simplilearn.com/tutorials/git-tutorial/merge-conflicts-in-git) occurs, and Git needs your help to select which changes to include in the final merge.

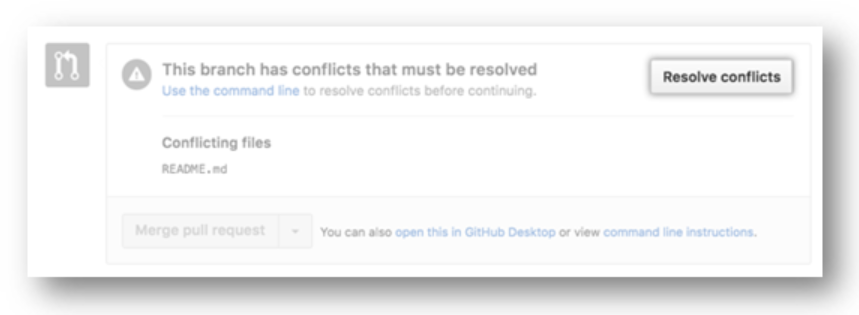
#### Resolve using GitHub conflict editor

When competing for line changes, i.e. when users make different modifications to the same line of the same file on different branches in your Git repository, this is done to avoid merging conflicts.

Step 1: Under your repository name, click Pull requests.



Step 2: Click the pull request with the merge conflict you'd like to resolve in the "Pull Requests" list. Click Resolve conflicts near the bottom of your pull request.

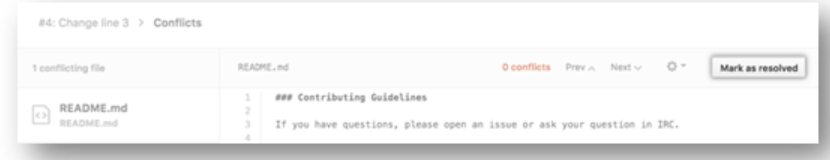


Step 3: Decide whether you want to maintain just your branch's changes, only the other branch's changes, or make a completely new modification that includes both branches' changes.

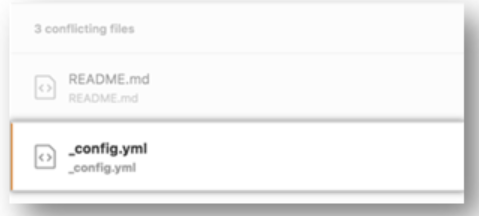
Step 4: Delete the conflict markers and make the changes you want in the final merge.



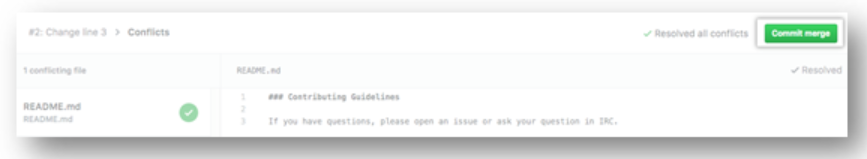
Step 5: If your file has over one merge conflict, scroll down to the next set of conflict markers and repeat steps four and five to resolve the issue. Mark the file as resolved once you've resolved all the conflicts.



Step 6: If you have more than one file with a conflict, go to the left side of the screen and select the next file you wish to edit under "conflicting files". Then repeat the above steps until you've resolved all the merge conflicts in your pull request.



Step 7: Click Commit merge once you've resolved all of your merge conflicts. It merges the entire base branch into your head branch as a result of this. Click Merge pull request to combine your pull requests.



Resolve the issue with a local clone of the repository and commit the update to your GitHub branch.

It resolves all other forms of merge conflicts using this method. To push the update, you can use the command line or a program like GitHub desktop.

Step1: Open Git Bash. Go to the local Git repository where the merge conflict exists.



Step2: Make a list of the files that have been affected by the merge dispute. In this case, there is a merge conflict in the file styleguide.md.



Step 3: Navigate to the file with merge conflicts in any text editor, such as Sublime Text or Atom. Look for the conflict marker "<<<<<<" if you want to see where the merging conflict started in your file.

After the line "<<<<<<HEAD", you'll see the changes from the base branch.

Step 4: Next you’ll see =======, which divides your changes from the changes in the other branch, followed by >>>>>>> BRANCH-NAME

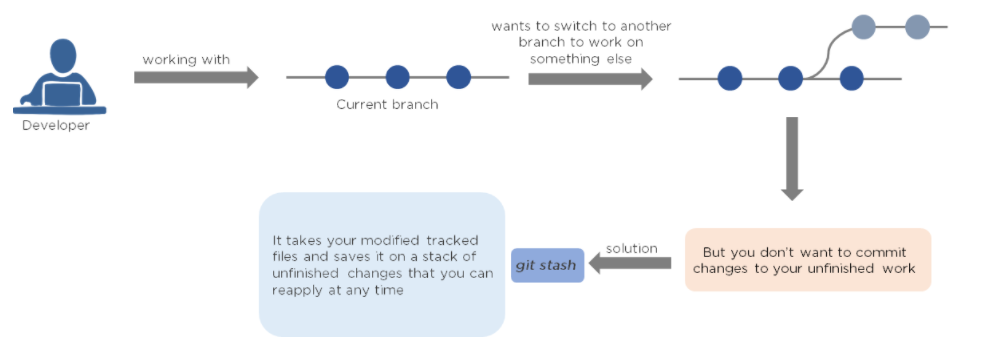
Step 5: Decide whether you want to simplt maintain your branch's changes, only the other branch's changes, or make a completely new modification that includes both branches' changes.

Step 6: Delete the conflict markers, <<<<<, =====, >>>>> and make changes you want in the final merge.

Step 7: Add or stage your changes. Commit your changes with a comment.

You may now combine the branches using the command line, or you can upload your changes to your [GitHub](https://www.simplilearn.com/tutorials/git-tutorial/what-is-github) remote repository and merge them in a pull request.

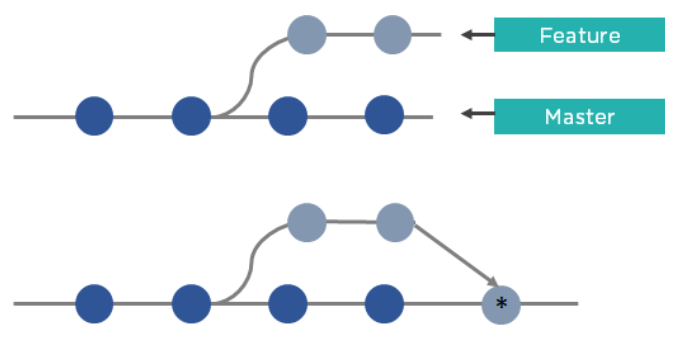
### 7. What is Git Stash?



### 8. Differentiate Between Git Merge and Git Rebase

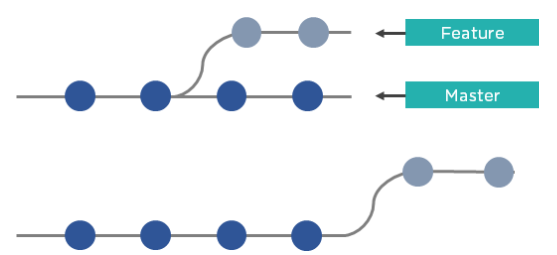
Assume you're working on a new feature in a dedicated branch, and another team member pushes new commits to the master branch.

#### Merge



Merge is used to incorporate new commits into your feature branch. Every time you need to include modifications, this requires an extra merging commit. It taints the history of your feature branch.

#### Rebase



You can rebase the feature branch into master instead of merging it. This incorporates all the new commits in the master branch. It also re-writes the project history by creating brand new commits for each commit in the original branch.

This brings us to the end of Git Questions. These are some of the most important Front End Interview Questions related to Version Control.