**Git Branch**

1. git branch – lists all the branches
2. git branch <branch-name> - to create a new branch
3. git checkout <branch-name> - switches the workspace to the specified branch
4. git checkout -b <branch-name> - creates a new branch and switches the workspace to the branch
5. git branch -d <branch-name> - safe delete (you should be outside the branch to be deleted)
6. git branch -D <branch-name> - force delete (use with caution and cannot be undone)
7. git status – to check the changes in the branch
8. git log – to go through past commits (dates, name of developer who committed, etc.)

Whenever we do development, we create a new branch from the master branch – called the feature branch, and after the changes have been made to the feature branch, we merge the feature branch back to the master branch.

Steps:

1. switch to the master branch: git checkout master
2. merge the child branch into the master branch: git merge feature-branch

To merge two remote Git branches, you can follow these steps:

1. Clone the remote repository: Begin by cloning the remote repository that contains the two branches you want to merge.

$ git clone <url-to-remote-repo>

1. Fetch the two branches: Use git fetch to fetch the two branches from the remote repository.

$ git fetch origin branch1

$ git fetch origin branch2

1. Checkout the branch you want to merge into: Use git checkout to switch to the branch that you want to merge the other branch into.

$ git checkout branch1

1. Merge the other branch into your current branch: Use git merge to merge the other branch into your current branch.

$ git merge branch2

1. Resolve any merge conflicts: If there are any conflicts between the two branches, you will need to resolve them manually.
2. Push the merged changes to the remote repository: Once you have resolved any conflicts, you can push the merged changes to the remote repository.

$ git push origin branch1

By following these steps, you should be able to merge two remote Git repositories into one. Keep in mind that merging two repositories with unrelated histories can be a complex process and can result in conflicts, so it's important to be careful and take your time.