

## git Lab program - 5.

### collaboration and Remote Repositories

Fetch the latest changes from a remote repository and rebase your local branch onto the updated remote branch.

Remote repository - Repository stored on github  
ex: origin (default remote name in git)

Local Repository - The copy on your computer.

Fetch - `git fetch` - Download the latest changes commits from the github

Rebase - take your changes and place them on top those new updates.

If someone changes something on Github, you can fetch to see those changes and rebase your own changes on top of them to keep your version up.

Step 1: Create a new repository on Github  
ex: gitlabs-rebase-demo

Step 2: Add one file in Github  
click add file → Create new file → commit

Hello from  
Github version 1

→ click commit

Step 3: clone it your computer

cd Desktop

mkdir gitlabs

cd gitlabs

git clone https://github.com/gayathishn/gitlabs-rebase-demo.git

cd gitlabs-rebase-demo

Step 4: open that, edit the file locally

Hello from  
local version 2

→ git add.

git commit -m "add version 2"

Step 5: simulate a change on Github

→ open demo.txt

edit the file add one more line

Hello from  
github version 3

→ Then commit

now the remote repo has new changes that your local repo does not know about yet.

Step 6: Fetch the latest changes from github  
In gitbash run

git fetch origin

This downloads the new github commit but doesn't apply it yet.

If you want to see what changes

git log origin/main --oneline

step 7: Rebase your local branch  
git rebase origin/main

this tells git → Apply my local commits after  
the latest github commit  
if no conflicts occurs, rebase finishes successfully

→ If there are conflicts occurs

Fix the conflicts in the file manually

12345 Head

lines (s) from remote (origin/main)

=====

lines (e) from your local commit

>>>> (commit - how)

remove the

1111, 22222,

>>>> marker

save file

→ add the file git add demo.txt

⇒ continue the rebase git rebase --continue

step 8: Verify

cat demo.txt

Hello from ~~your~~ version 1  
Hello from github version 3  
Hello from github version 2

this means both your local and remote changes  
are now together

step 9: push back to Github

Finally push your updated work

git push origin main --force

needed bcz we changed history and rebase