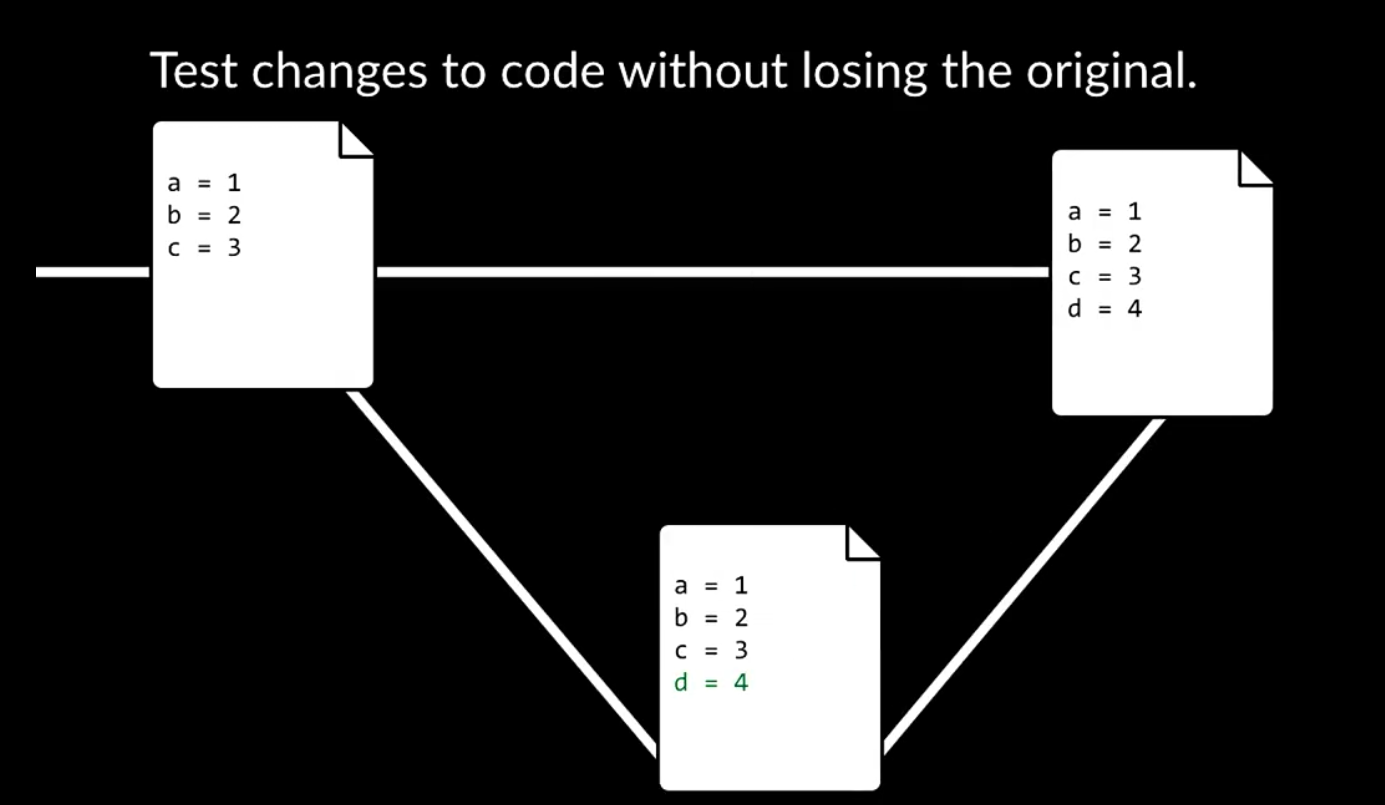
Test changes to code without losing the original.

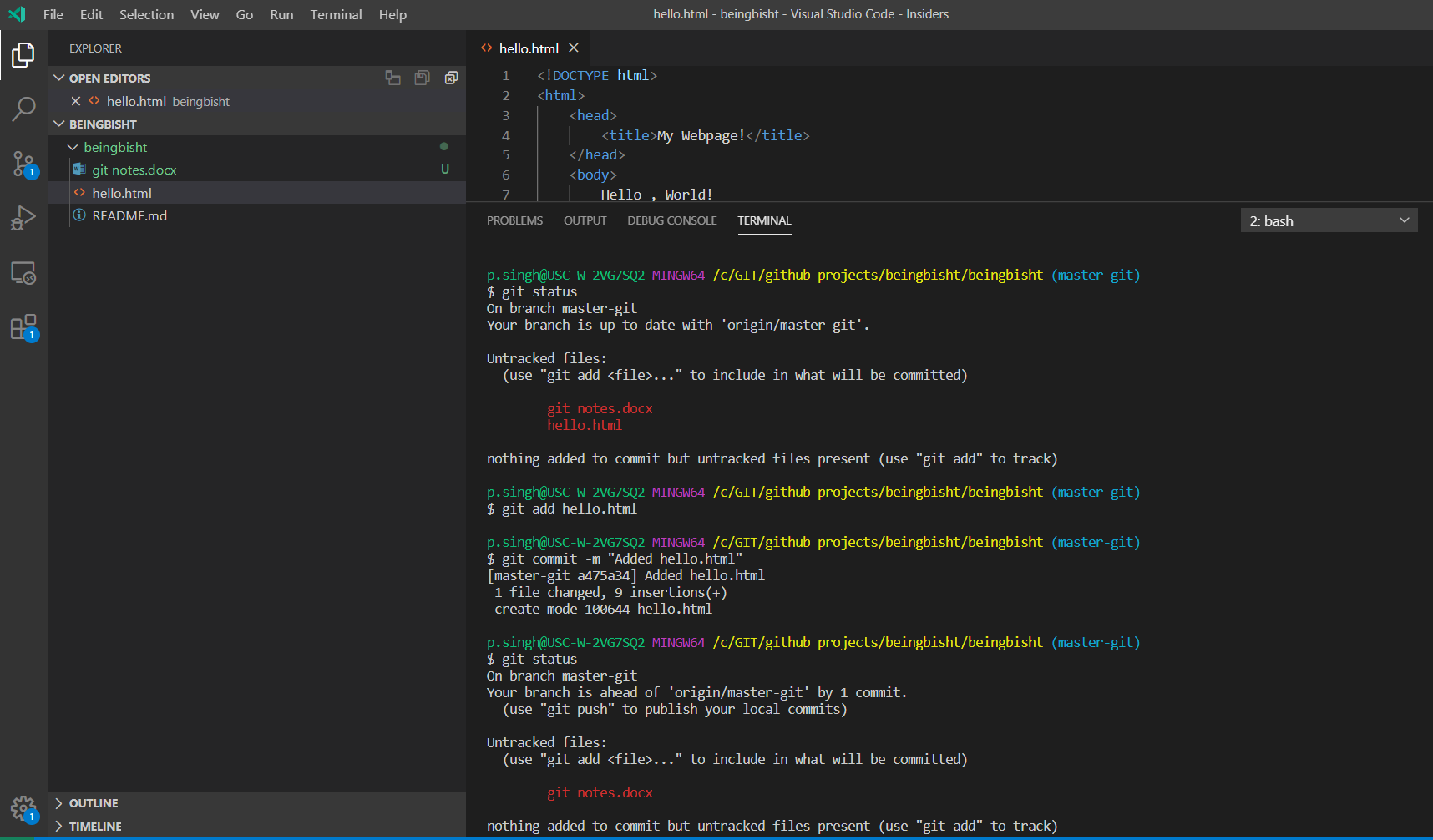


Revert back to old versions of code.

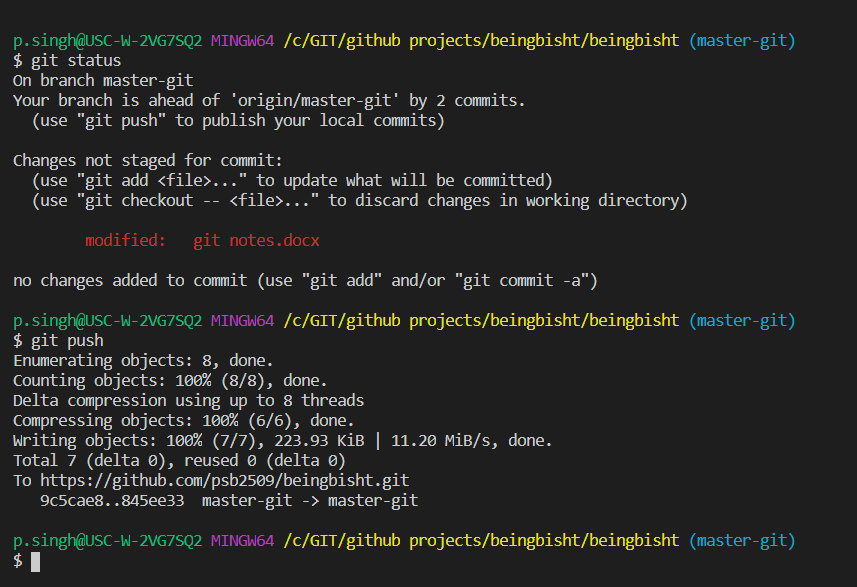
git clone <url> - Clone a copy of code-base from remote repo in local

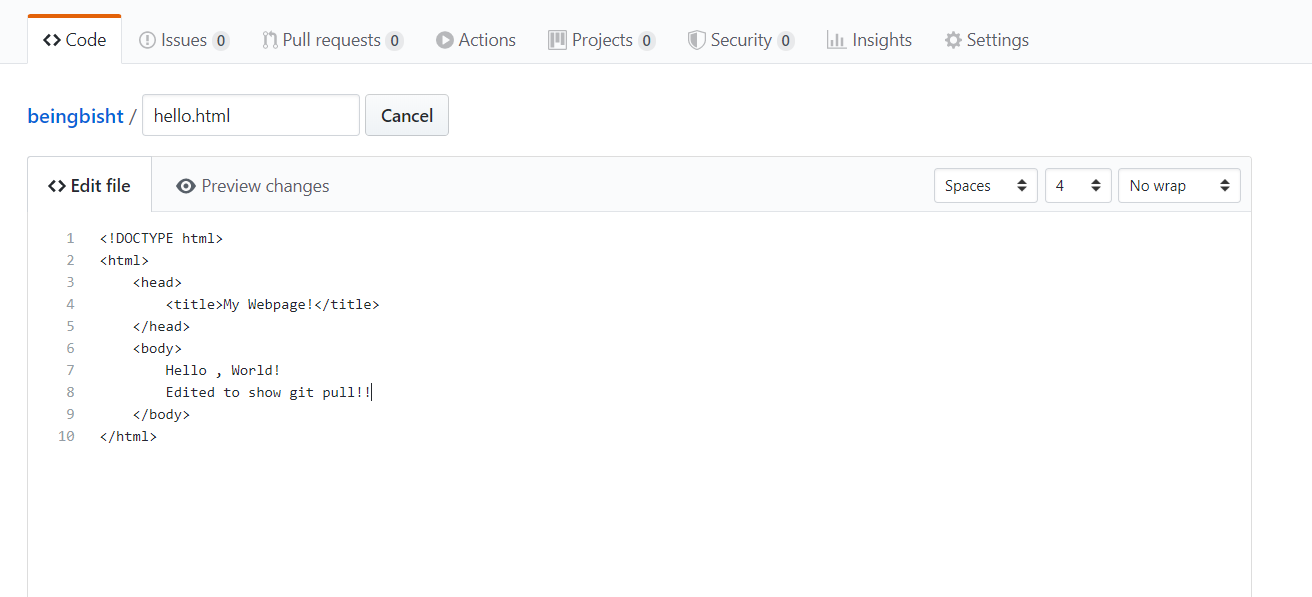
git add <filename> – Tell git that these are the files we want to commit to git next time we commit anything to repo

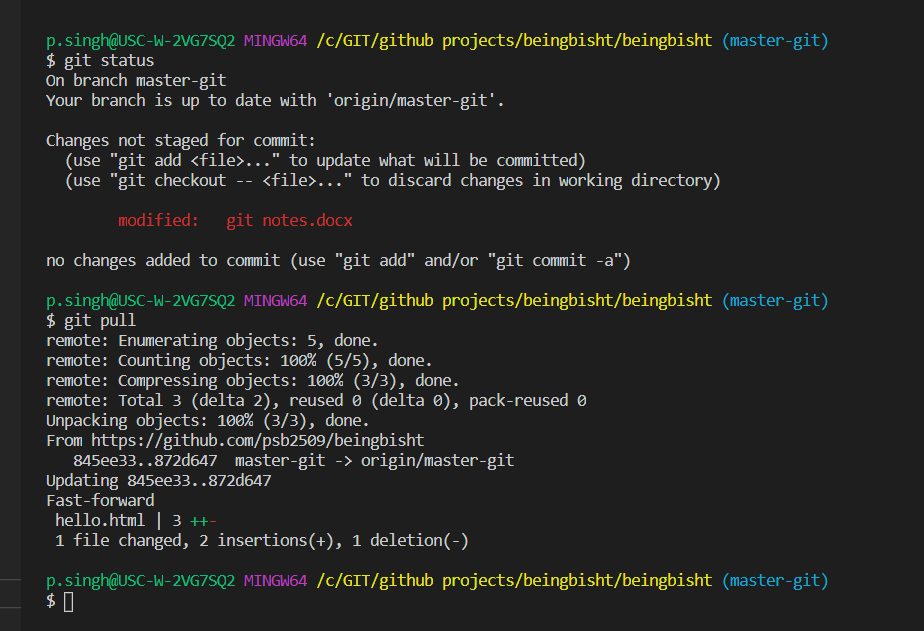
git status – Tells you what’s currently going on in your repository.



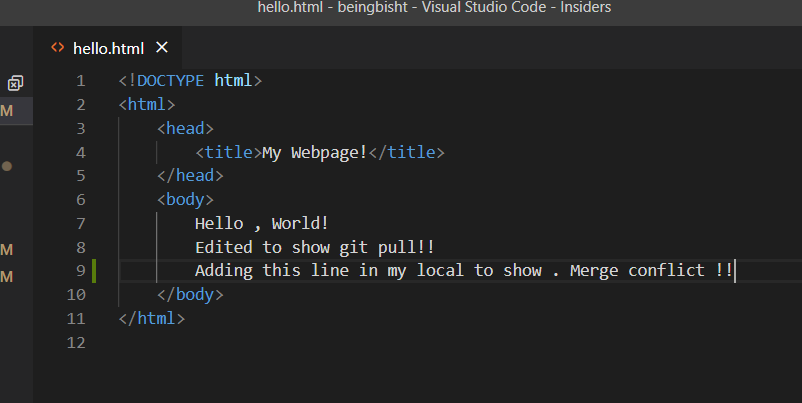
git push : Push the changes committed on your local repo to remote repo stored on the internet .

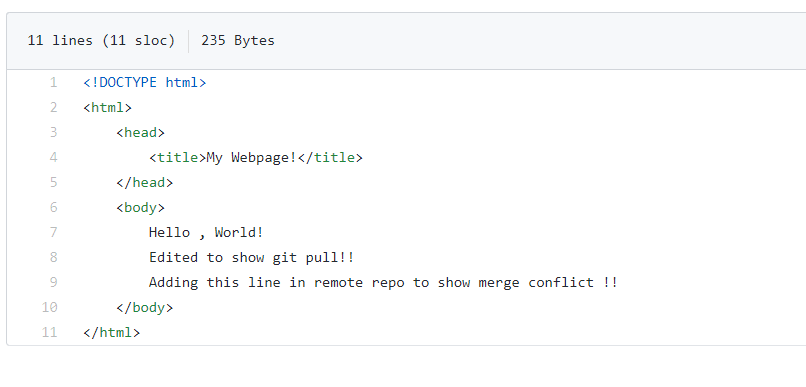


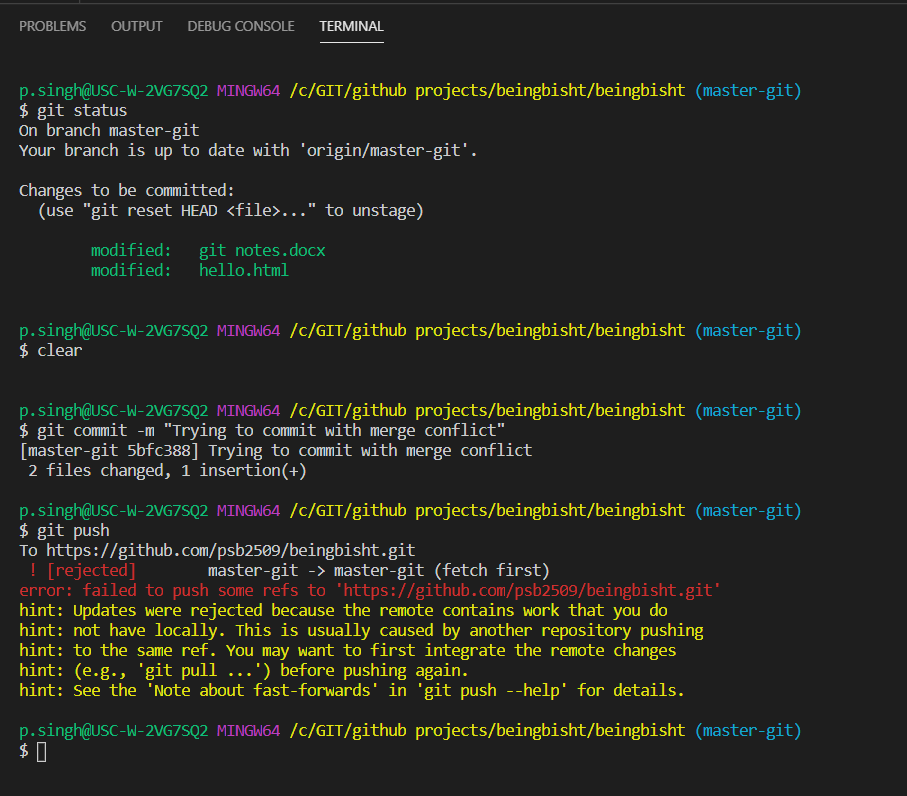




Merge Conflicts:

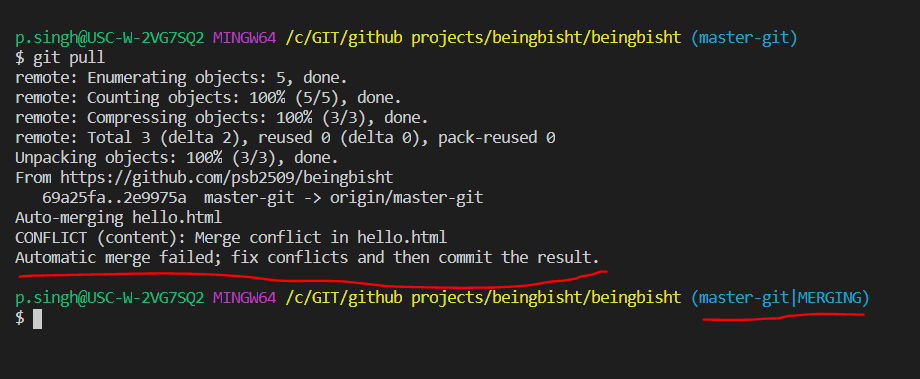




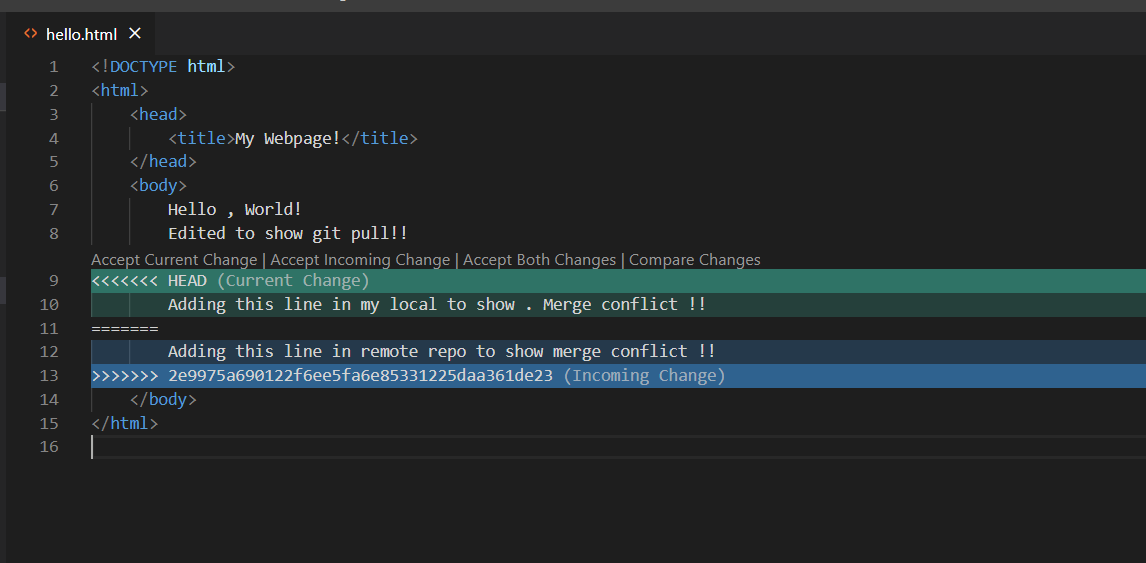


I was able to add the change in my local and was able to commit as well. But when I push this change to remote repo it throws error.

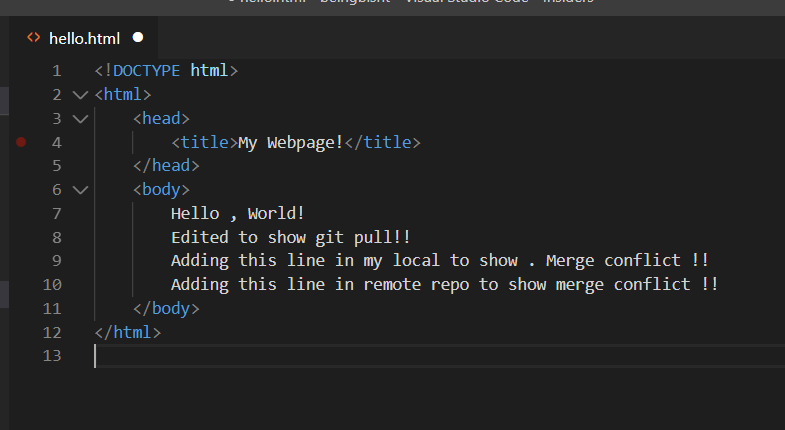
I need to now git pull to bring in changes done in remote.



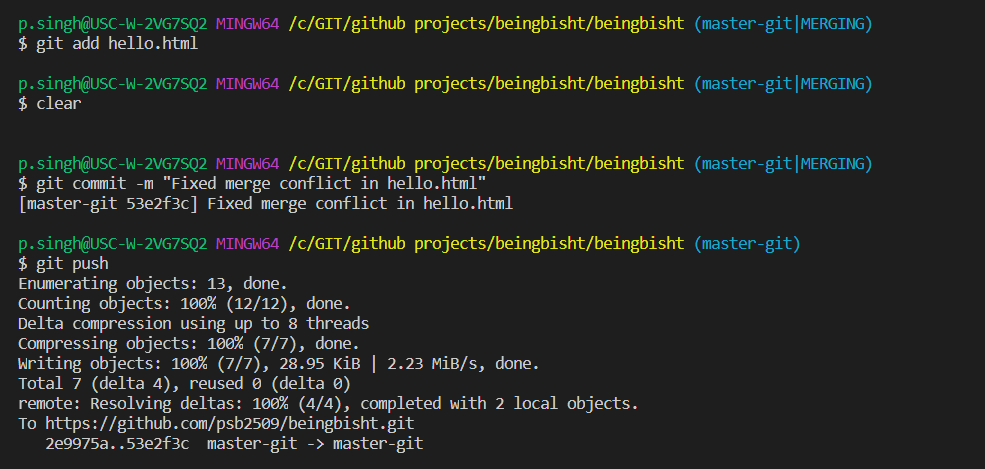
Now if I open the file with conflict. I see this .



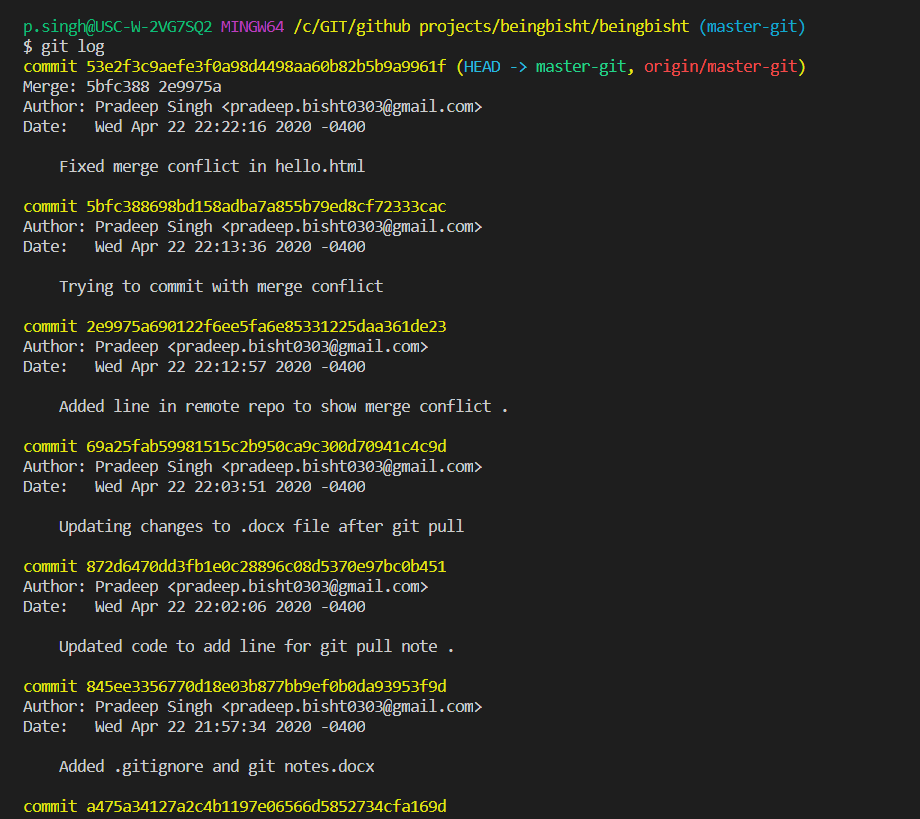
After accepting both changes.



I need to add and commit this file and then I can push the changes to git



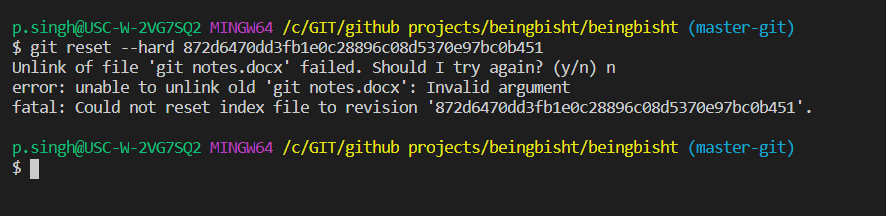
git log – gives you a history of all the commits you have made starting with latest commit first.



git reset <commit hash>

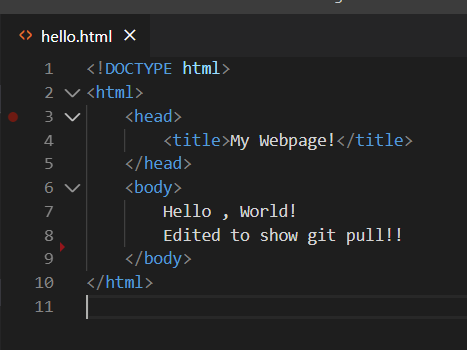
This will reset your repo code to the version that you provide in the command.

* git reset --hard < commit hash> : Will reset the repo version to a commit version .
* git reset --hard origin/master : Will reset the repo version to a remote master origin.

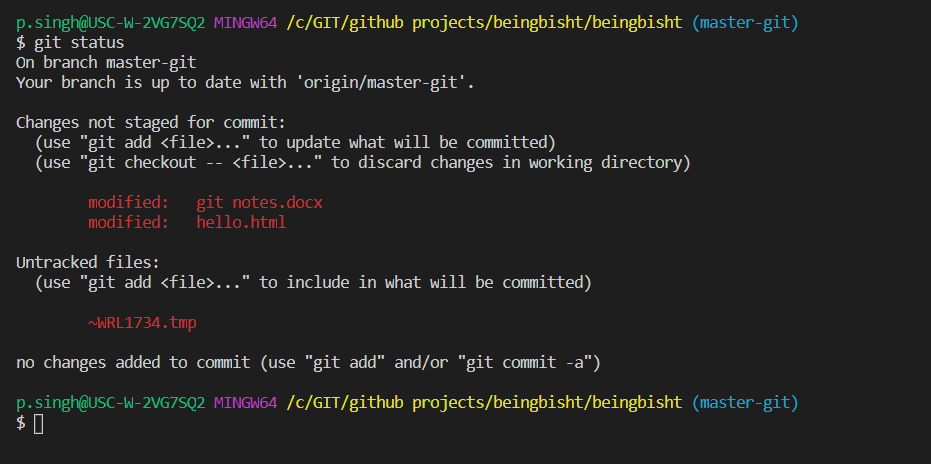


I tried resetting my code to version before I added changes for merge conflict demo. Since I had also commit .docx changes and it was open git tried confirming for reset. When I chose ‘n’ is reset only those files which it could “hello.html” which was what I wanted but did not change .docx because it could not unlink it. It also did a reset of .gitignore to version before I added \*tmp to it .

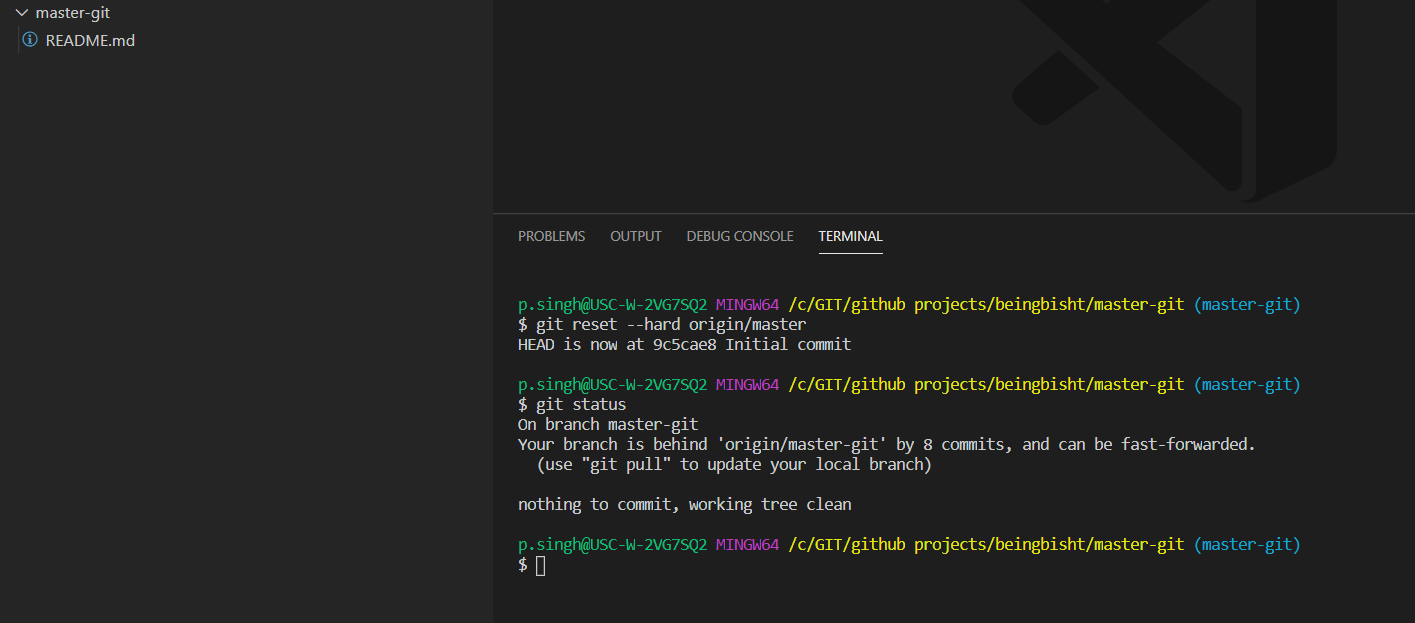
My repo has now following code.



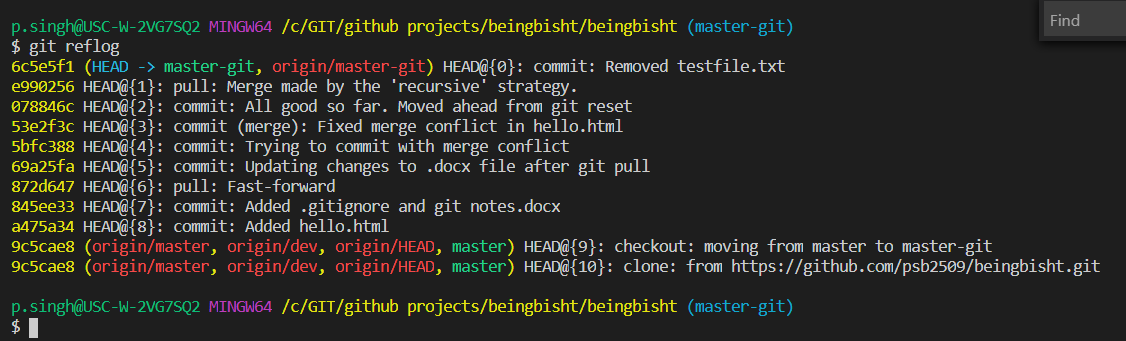
Git status gave me this .



git reset --hard origin/master . Take the repo back to master branch code .Whatever is there in master branch will appear back and whatever extra code was there. That will get deleted.

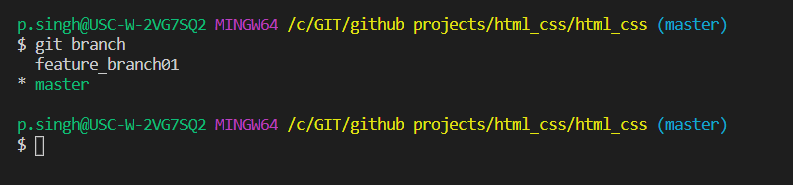


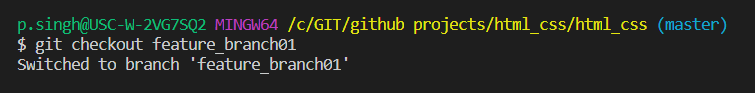
git reflog



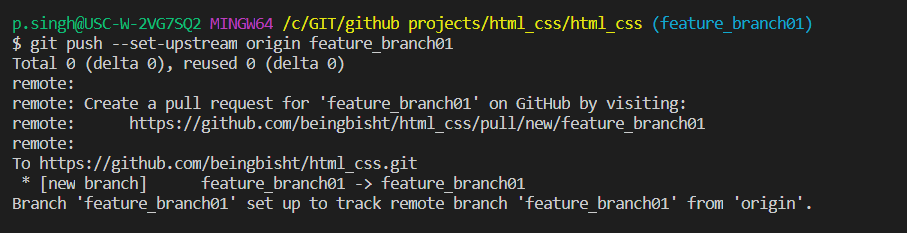
Gives a list of references to git changes .Much cleaner way of looking at commits

Branching:









Remotes:

Remote is the version of repo existing in GitHub

git fetch – will pull the changes from remote to your local.

git merge origin-master will merge the changes brought from remote to your local master branch

These two steps are combine in one command called git pull

