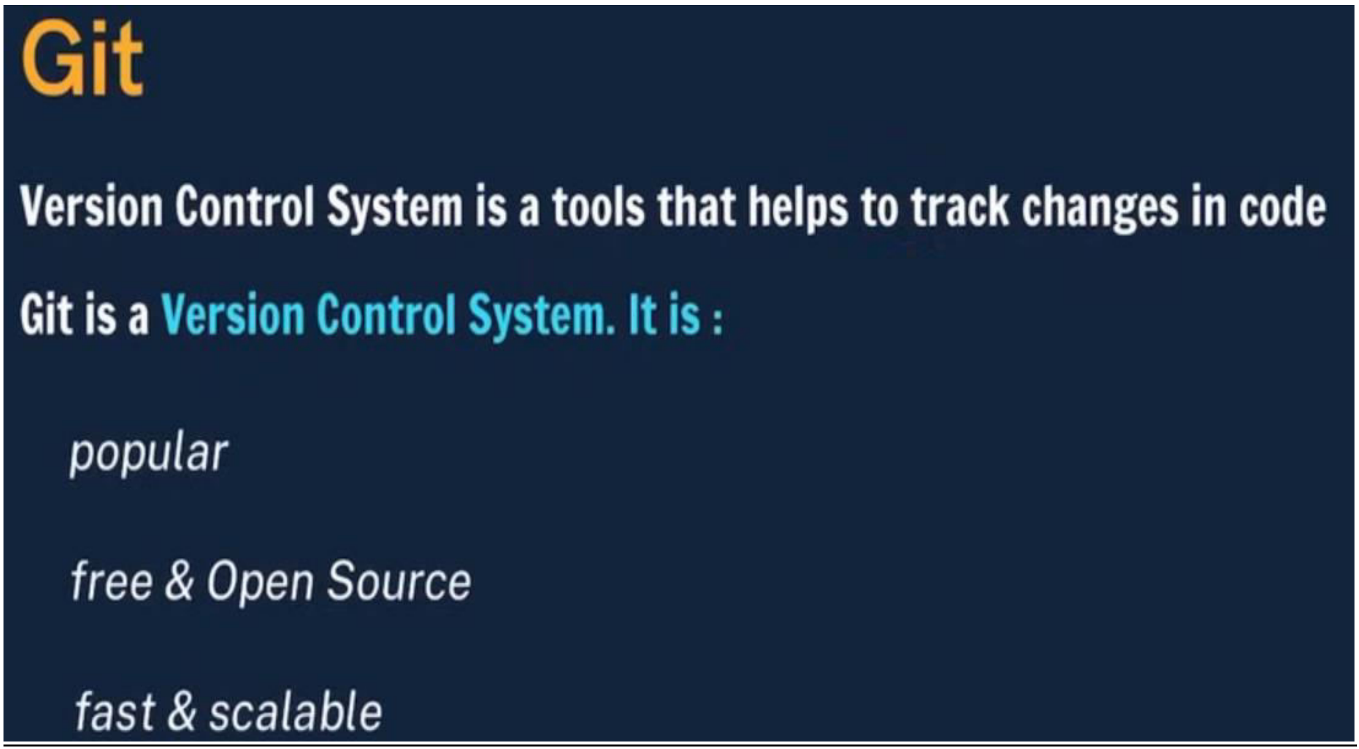
GIT AND GITHUB BY DIDI

GIT



Git

Version Control System is a tools that helps to track changes in code

Git is a Version Control System. It is :

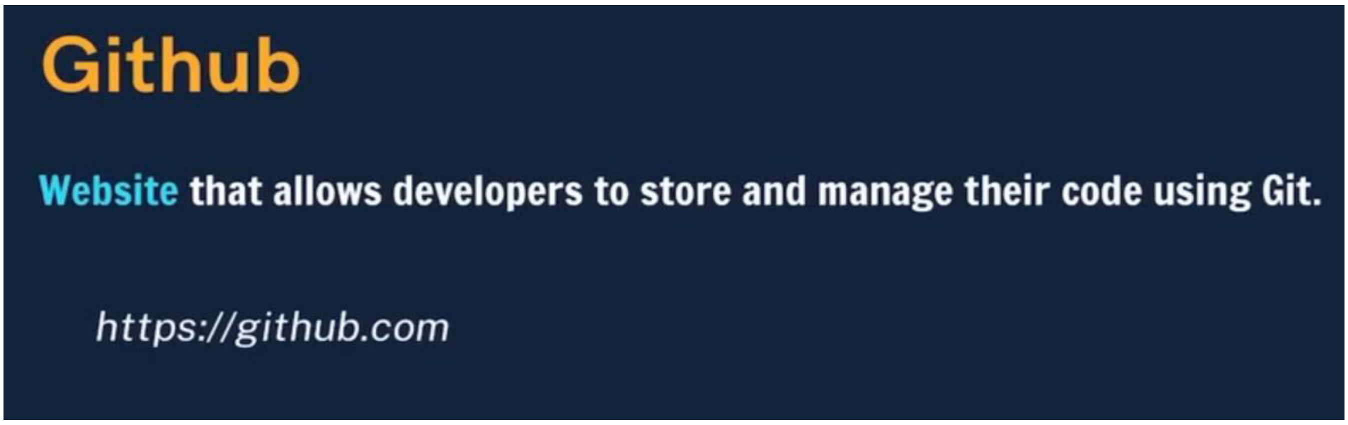
popular

free & Open Source

fast & scalable

Ab git hamare sare codes ko track krta h, matlb agr humne koi change kia ho pehle, hame uss point pe wapas jana h to wo bhi hum krskte h kyuki git sb kich track krke rakhta h. git collaborate krne me bhi madat krta h, jaise ki agr ek project pe 10 log kaam krr rhe h, to 20 sab koi na koi change krenge file me ,to kiska change accept hoga kiska nhi hoga aise bht sari problems git theek krdeta h.

GITHUB

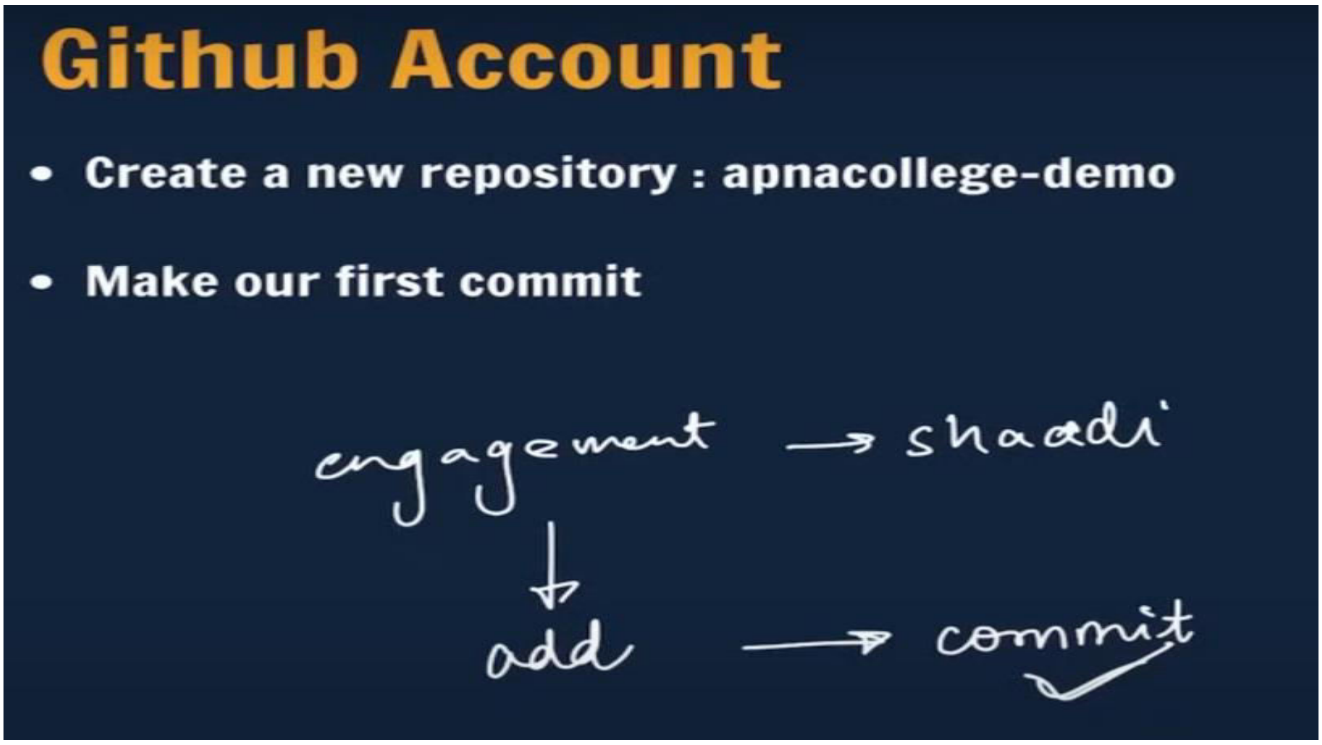


Github

Website that allows developers to store and manage their code using Git.

https://github.com

Git ke tracking records, code ke changes jo git se update krenge wo sab github website me rahega.

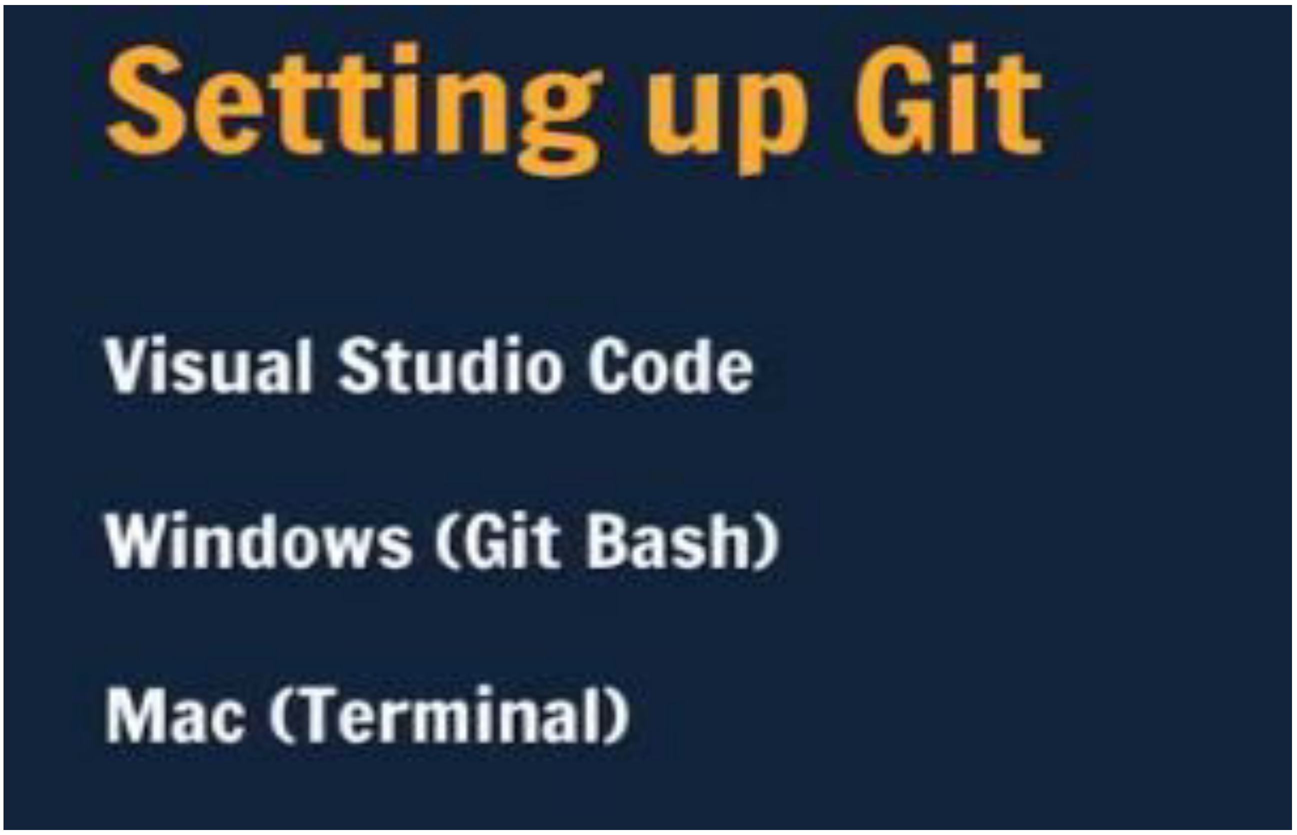


Github Account

- Create a new repository : apnacollege-demo
- Make our first commit

$$\begin{tikzcd}\text{engagement} \arrow[r] \arrow[d] & \text{shaadi} \\ \text{add} \arrow[r] & \text{commit} \checkmark\end{tikzcd}$$

We have created a repository in GitHub, the repository is like a big folder that stores our files and folders and also tracks the changes that occurred in each file. To make changes in the git repository, there is a path like when we did some changes so we have to write git add and git commit in order to make changes in the remote(GitHub) repository. The local(vs code where we work originally) repository changes are done manually so when we are completed with our changes then we save our changes in the remote repository too, by doing git add and git commit.



Setting up Git

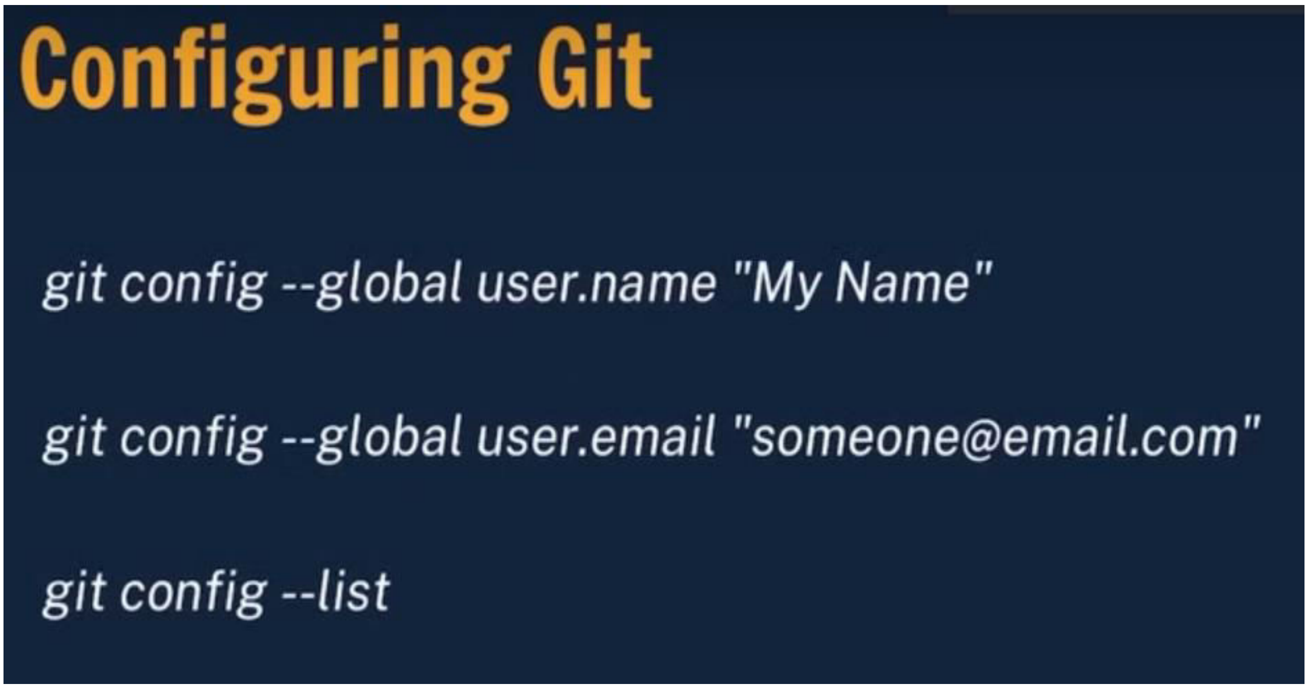
Visual Studio Code

Windows (Git Bash)

Mac (Terminal)

Git codes can be written on git bash and vs code terminals.

E

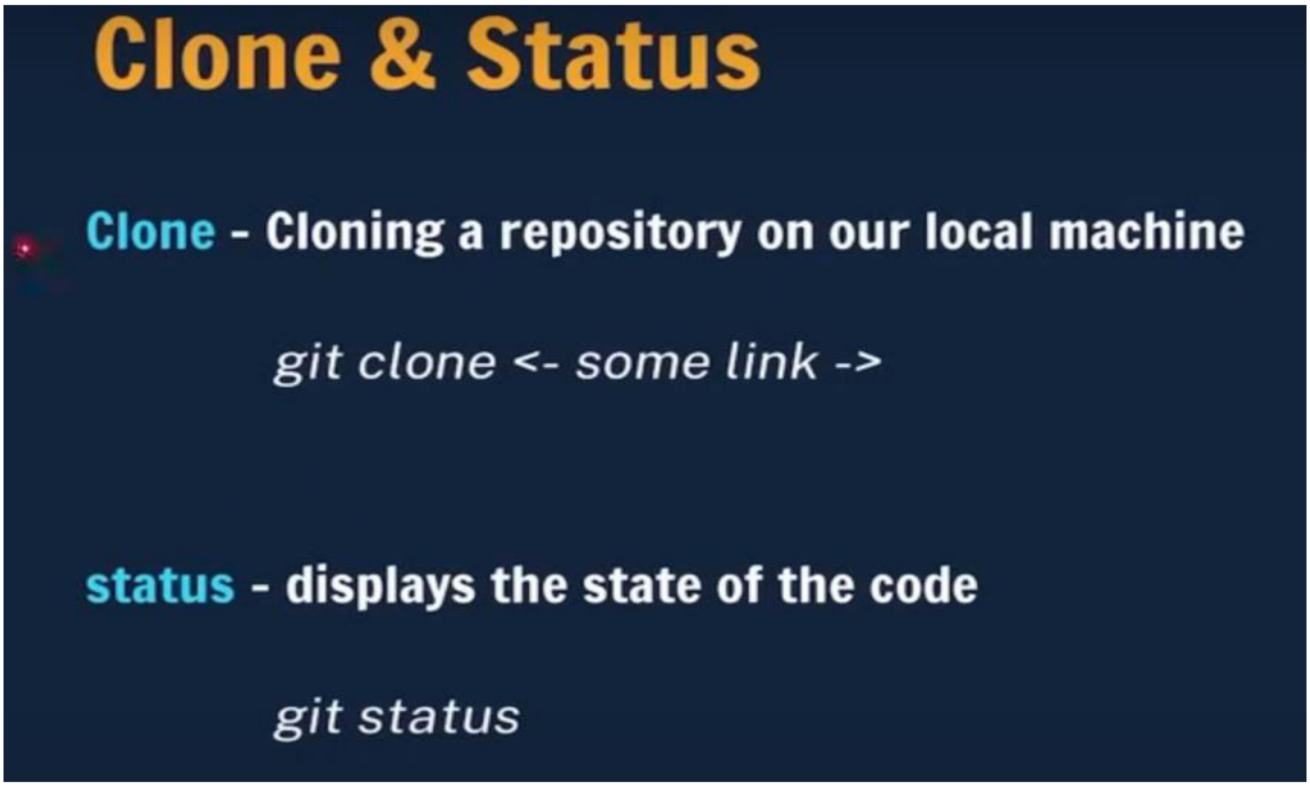


Configuring Git

```
git config --global user.name "My Name"

git config --global user.email "someone@email.com"

git config --list
```

First user should configure it as it should have permission to change/ files on GitHub repositories.

Now we will see how we are fetching entire data-set from remote repository to local repository.



Clone & Status

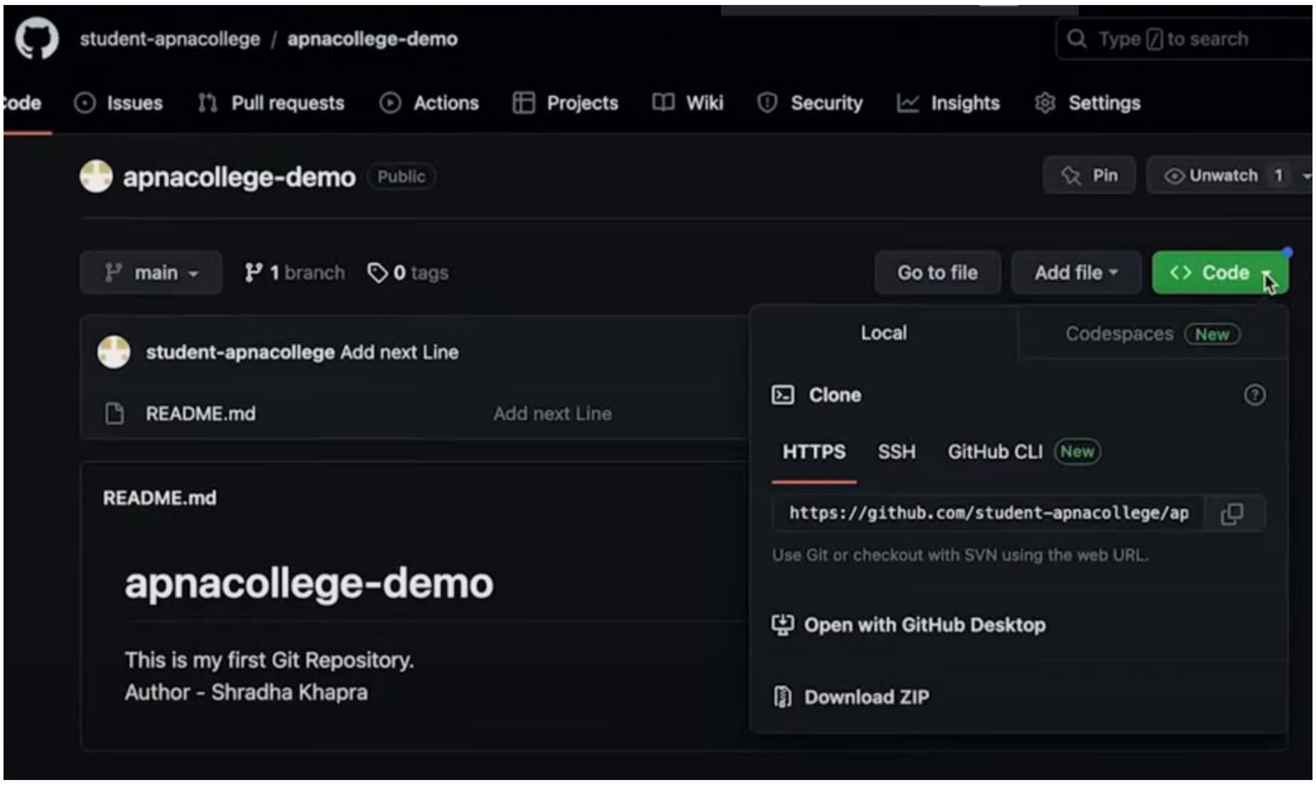
Clone - Cloning a repository on our local machine

git clone <- some link ->

status - displays the state of the code

git status

By using clone commands we can fetch/copy the entire remote repository in the local repository.



student-apnacollege Add next Line

README.md Add next Line

README.md

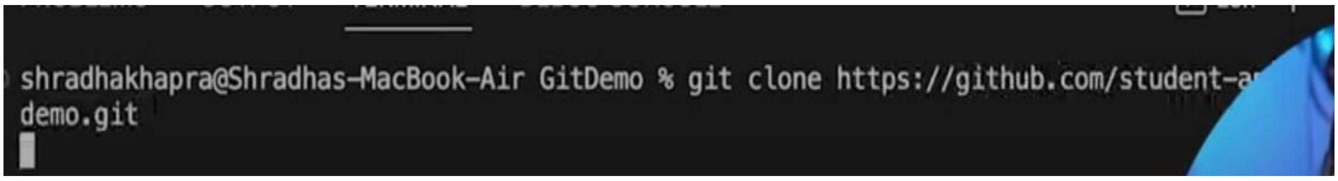
apnacollege-demo

This is my first Git Repository.
Author - Shradha Khapra

```
https://github.com/student-apnacollege/ap
```

Use Git or checkout with SVN using the web URL.

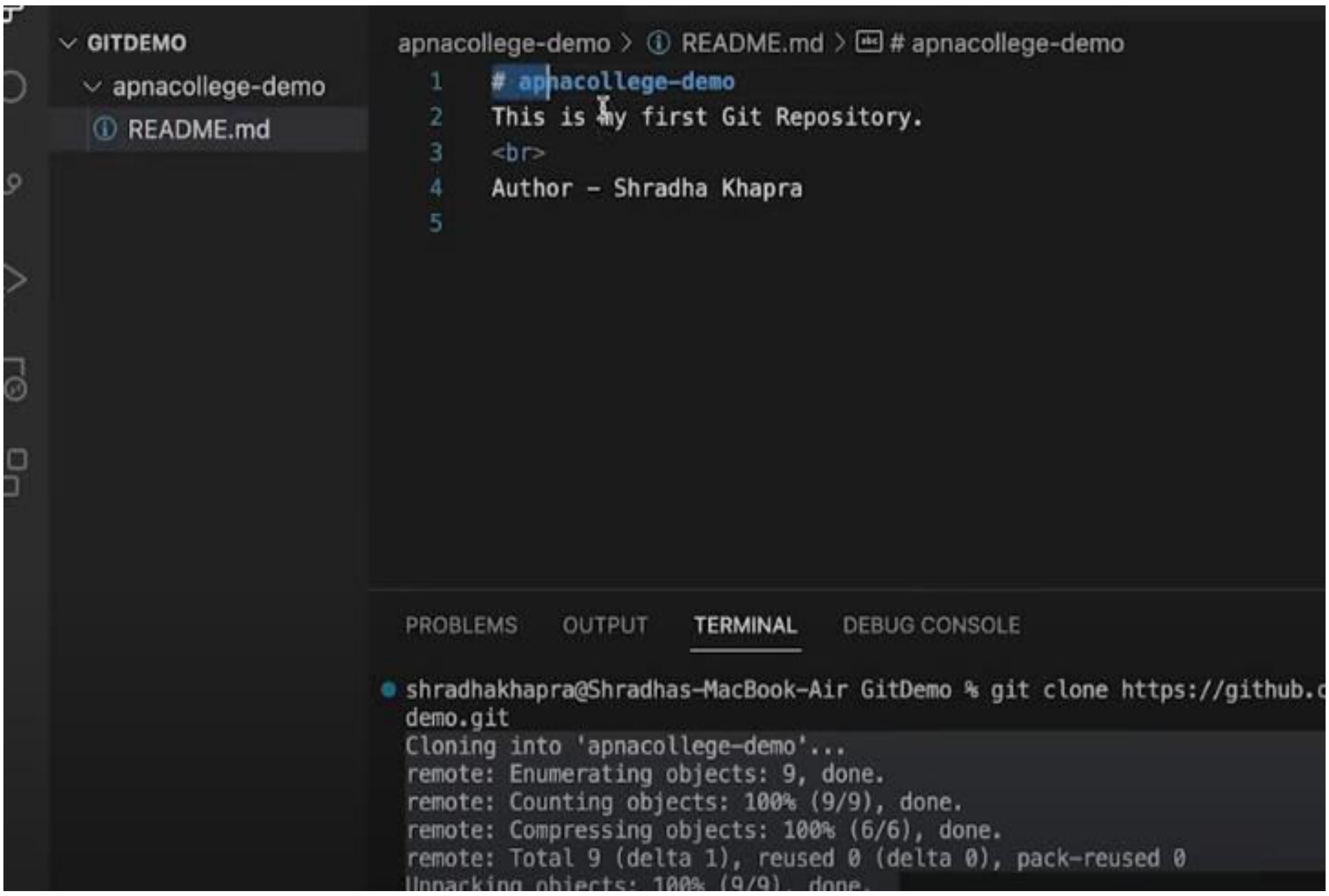
First of all, we have to copy the repository link, by clicking on the green button where the Code is written, then copy the HTTPS link.



```
shradhakhapra@Shradhas-MacBook-Air GitDemo % git clone https://github.com/student-a
demo.git
```

In vs code terminal write the git clone ←link(remote repository link you copied)→.

Then entire folder will get copied onto the local repository like this:-



```
# apnacollege-demo
This is my first Git Repository.
<br>
Author - Shradha Khapra
```

```
shradhakhapra@Shradhas-MacBook-Air GitDemo % git clone https://github.c
demo.git
Cloning into 'apnacollege-demo'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 9 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (9/9), done.
```

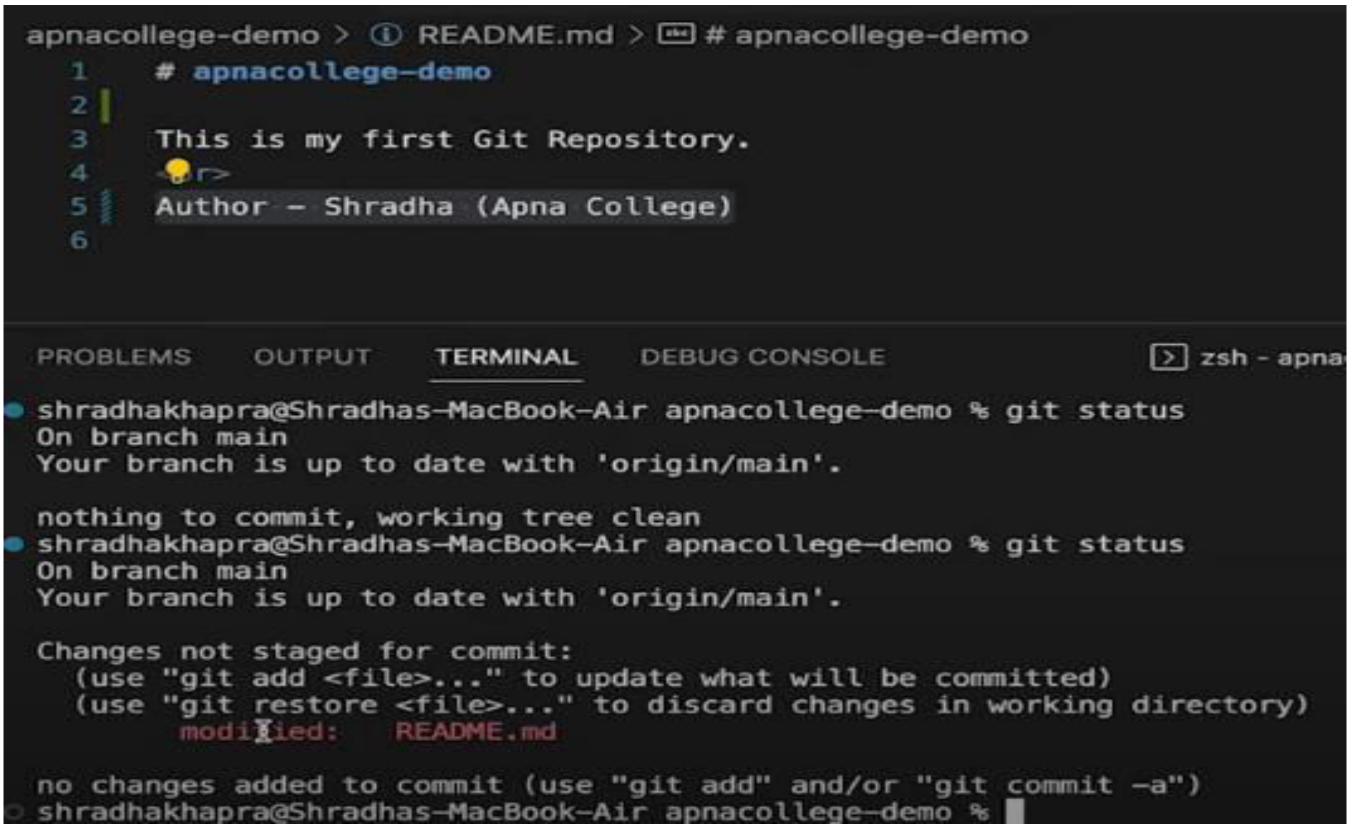
To see all the files write (ls) in the terminal and to see hidden files write (ls -a).

To create a file write (echo . > file_name) in vs code.

To create a new folder write (mkdir file_name) in the terminal.

If after doing ls -a, we are able to see .git then it means git is tracking our local repository.

GIT STATUS will provide the status of all the changes that we made like this:-

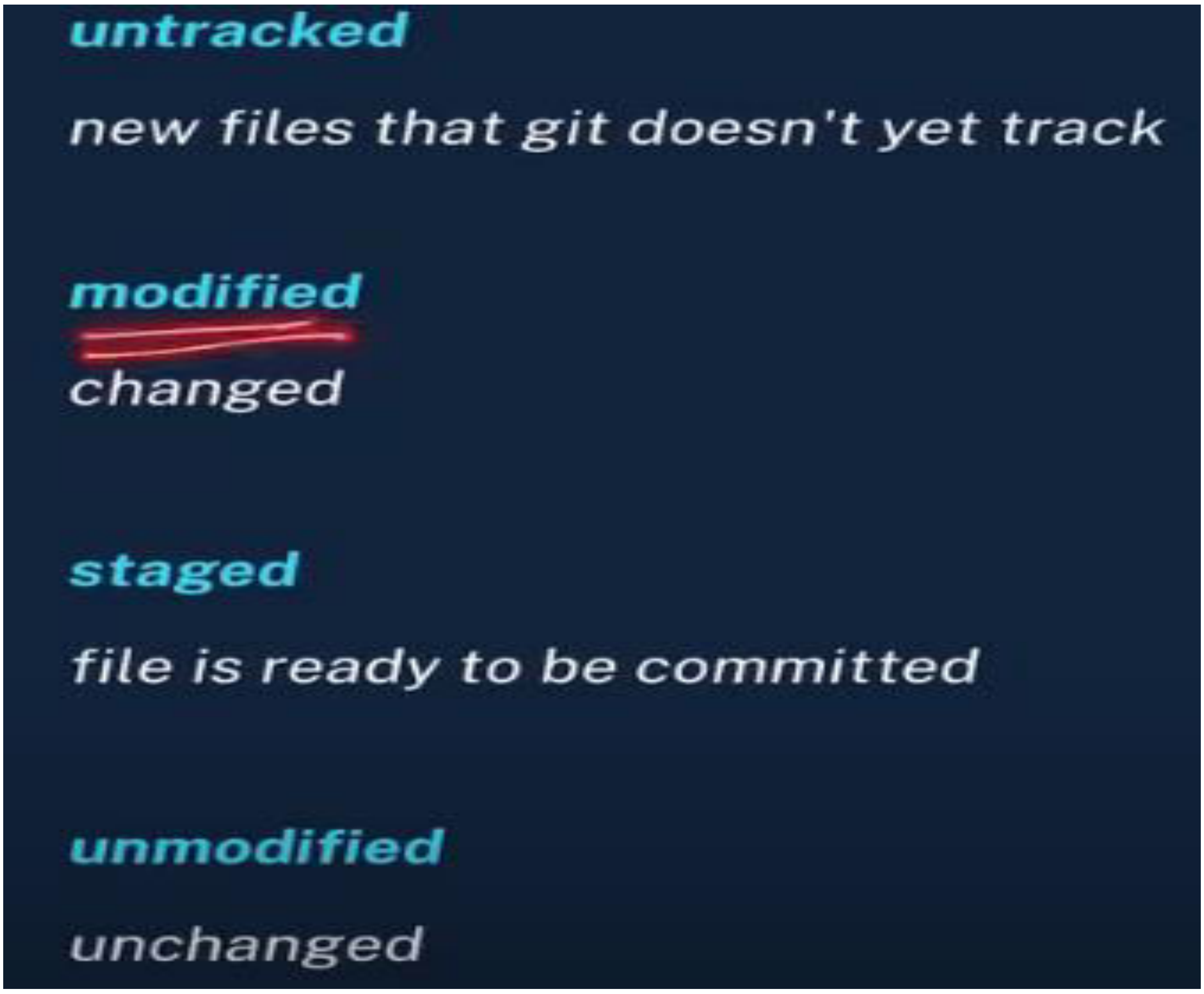


```
# apnacollege-demo

This is my first Git Repository.

Author - Shradha (Apna College)
```

```
shradhakhapra@Shradhas-MacBook-Air apnacollege-demo % git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
shradhakhapra@Shradhas-MacBook-Air apnacollege-demo % git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
shradhakhapra@Shradhas-MacBook-Air apnacollege-demo %
```

Here we modify/make changes in the readme file, so git status is telling us that readme.md is modified and not stages means after writing some codes on readme.md file you have not written the code (git add readme.md), after this we have to write (git commit -m “first commit”) this will make clear that we have agreed to save our changes.

There are 4 types of status categories:-



untracked

new files that git doesn't yet track

modified

changed

staged

file is ready to be committed

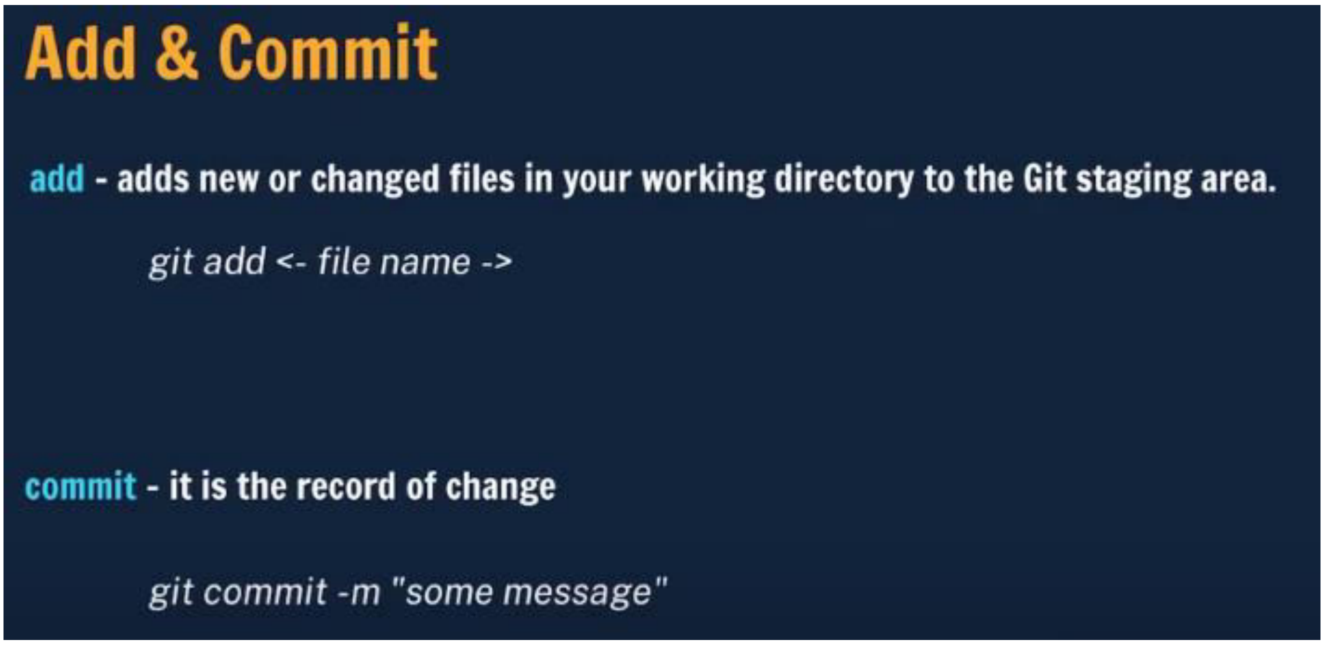
unmodified

unchanged

1) Untracked means you have created a new file that is not tracked by git, which means you have not done the code (git add new_file) and git commit.
2) Modified means git is tracking the file but the changes have not been staged means you have not coded yet (git add this_file)

3) Stages means you have done the code of add and commit, and now the file is ready to get updated on the remote repository.
4) Unmodified means no changes have been made to this file.

These are the git tracking commands.


Add & Commit

add - adds new or changed files in your working directory to the Git staging area.

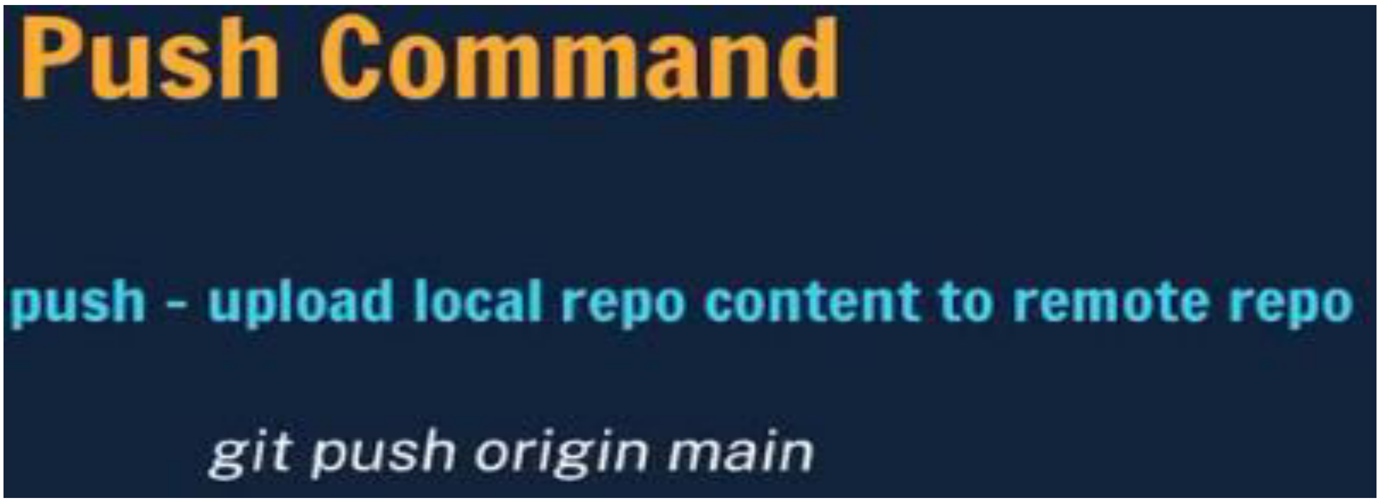
```
git add <- file name ->
```

commit - it is the record of change

```
git commit -m "some message"
```

Commit will save only those changes that have been put into the staging area by the add command.

Now after committing, we have to save these changes into our remote repository, here push command will play the role.

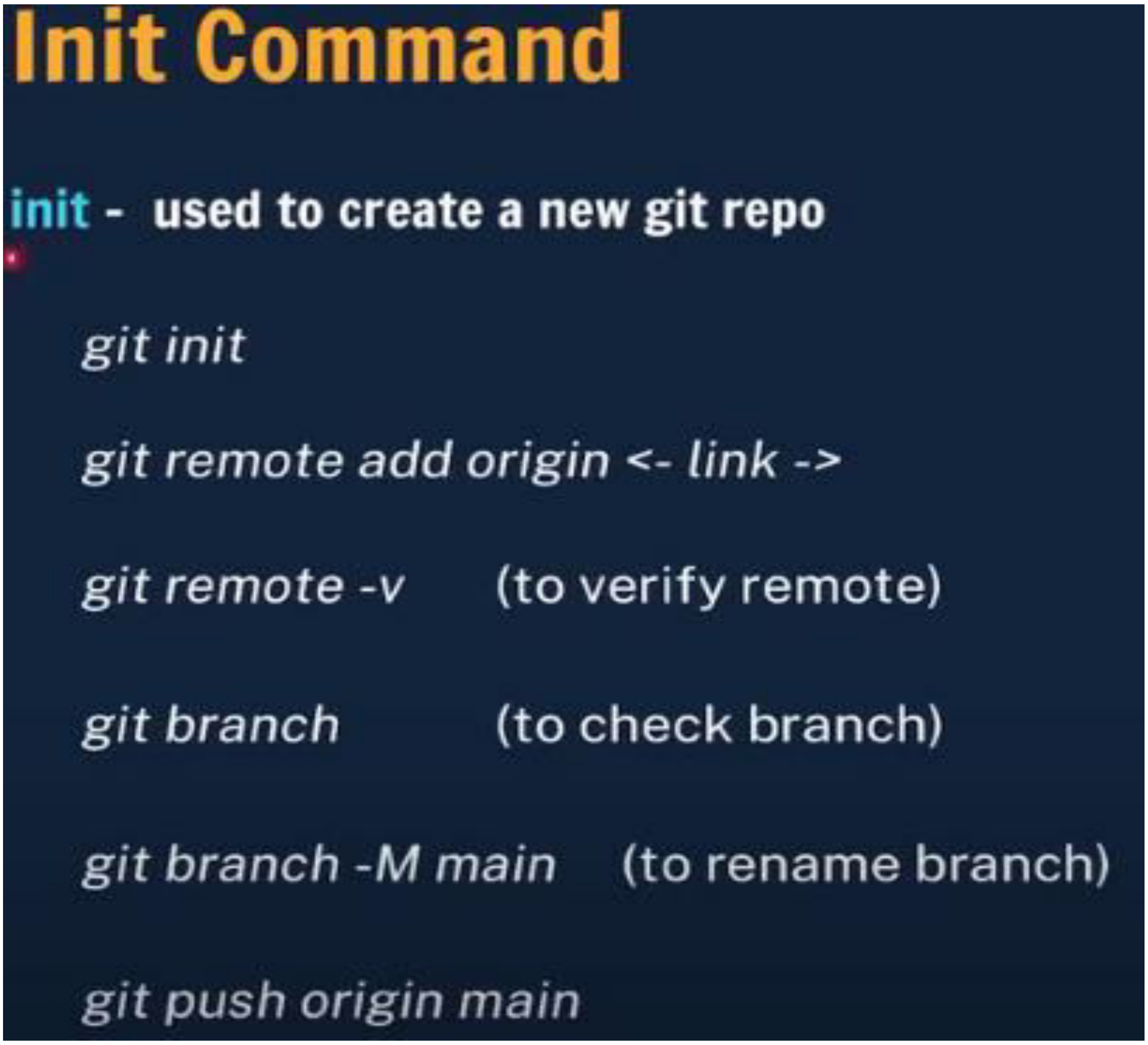

Push Command

push - upload local repo content to remote repo

```
git push origin main
```

Here git push is the basic command, but the origin is the default repository(it is the specific repository from where we have copied the HTTPS link) of the remote repository(a user may have many repositories) where we have to store our changes, and main the branch name. All our changes will be stored on that branch.

Now if we have not cloned the remote repository and directly started coding in vs then how we will connect with github?
ans:-



Init Command

init - used to create a new git repo

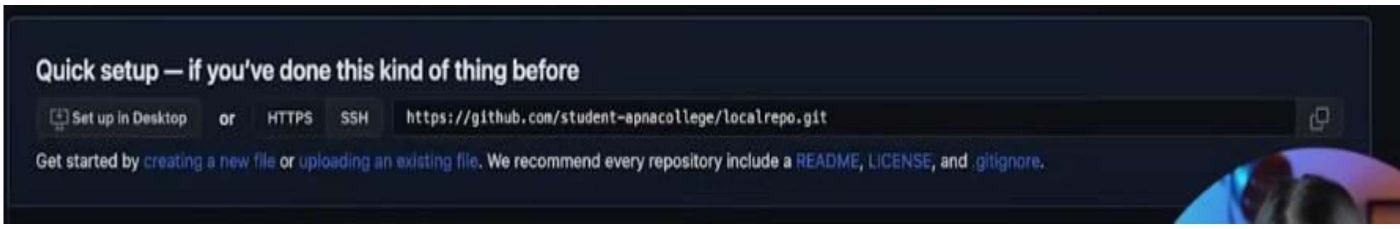
```
git init

git remote add origin <- link ->

git remote -v        (to verify remote)

git branch           (to check branch)

git branch -M main   (to rename branch)

git push origin main
```

This set of codes is used to make a connection with GitHub website.

Now (git init) is used to initialize the git on the current project/folder.

Now we have initialized git on the folder/project but we have to save the local repository in the remote repository.

So create a repository on github website manually(by going on the website and creating the repository).



Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH https://github.com/student-apnacollege/localrepo.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and gitignore.

Now copy this link and write this code in terminal:-
(git remote add origin ←link→)

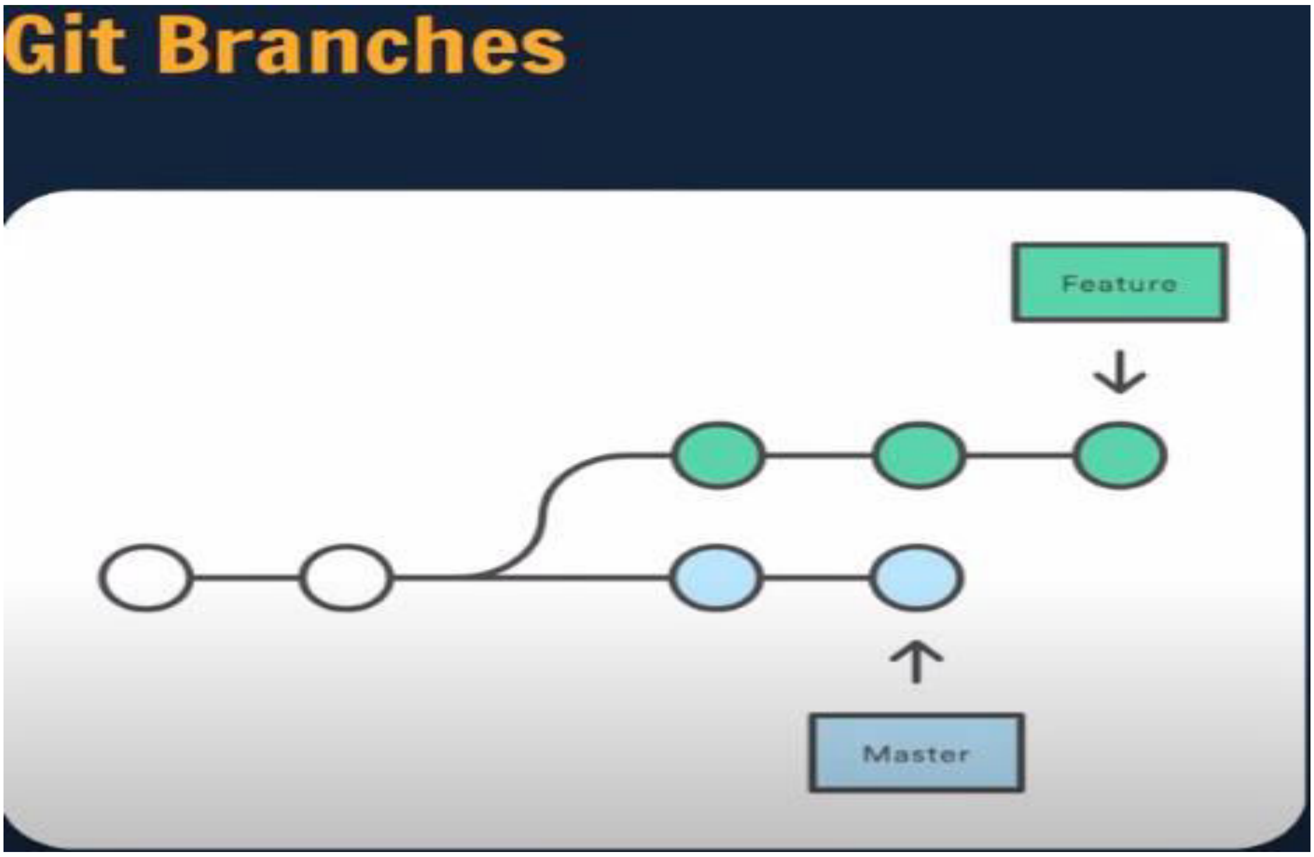
Here remote means remote repository and origin is the name(this name is decided(it is decided just now means when you write any name after git remote add it will be the name of the default repository) by us only) of the default repository(the repository which we have created just now).
overall code means we are adding the default repository which has name origin onto the remote repository/GitHub repositories.

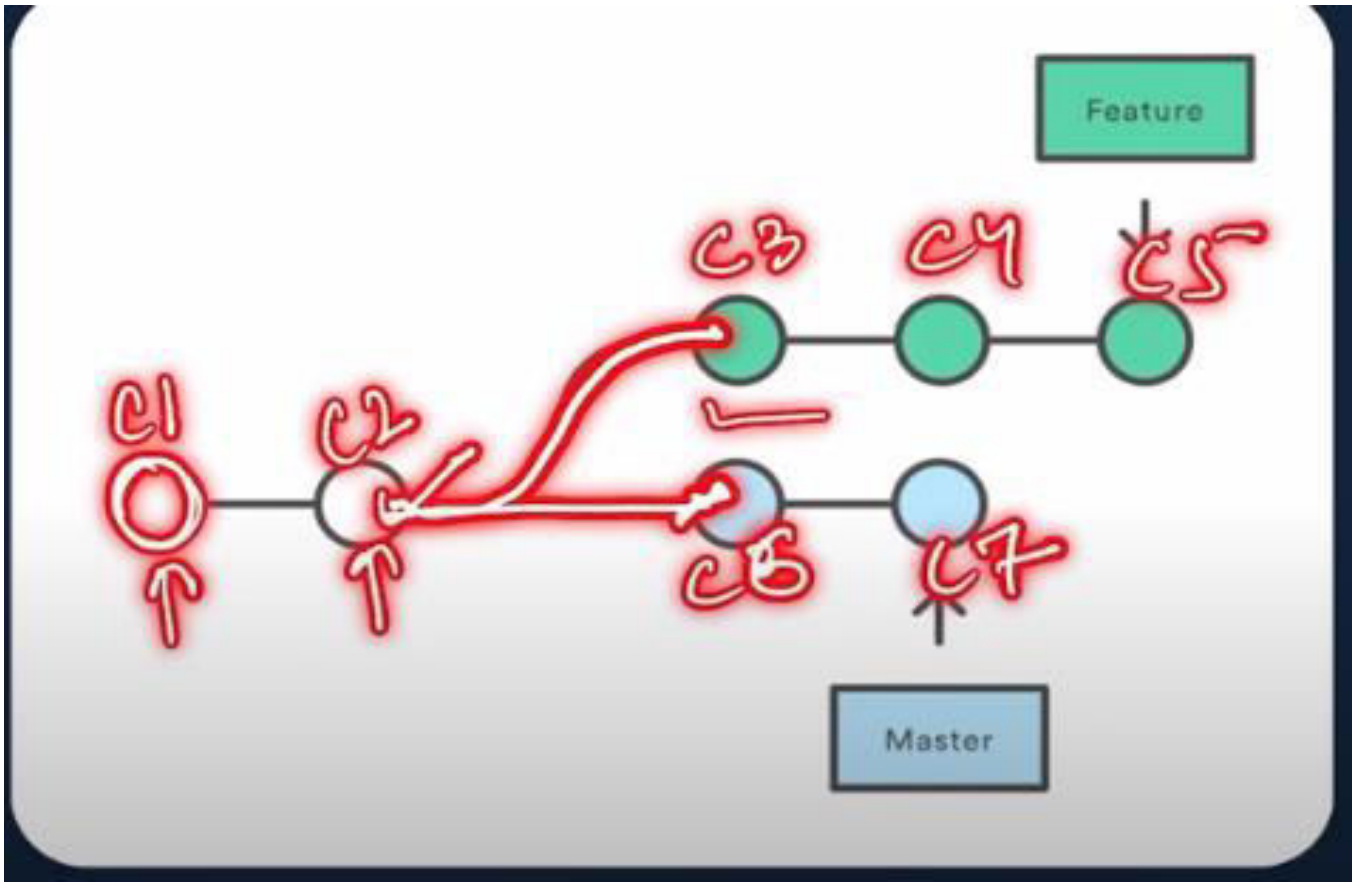
Now (the git branch) will tell us which branch we are currently working on).

Whenever we create our repo on vs code first there may be a change that the current/first branch name is master, so it should be changed to main as now GitHub first/MAIN branch has the name of main.

(git branch -M new_name) is the way to change the branch name.

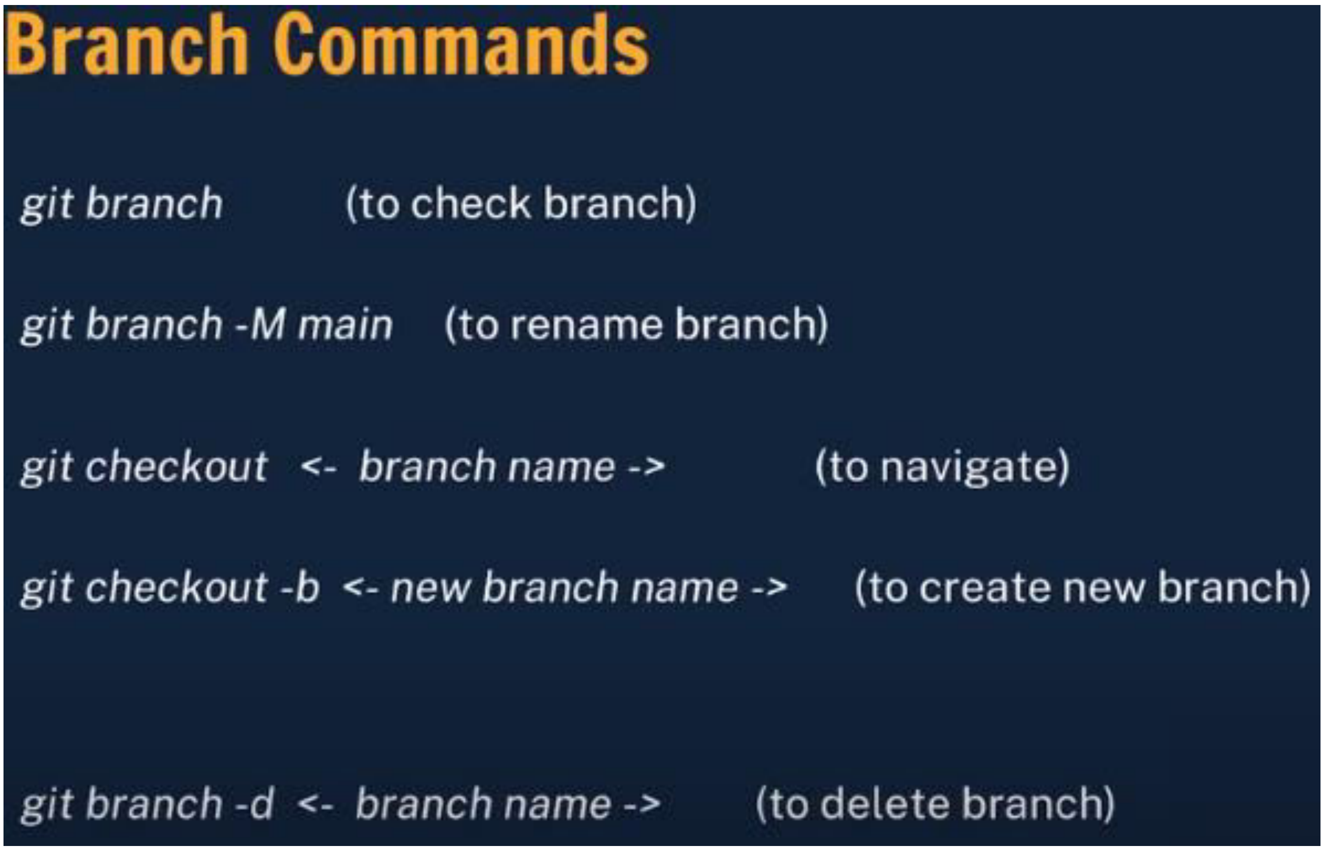
Shortcut:- by doing (git push -u origin main), then from the next time we can do (git push) so it will work as same as (git push origin main).





After the c2 commit, we have made a new branch3 on which we are committing c3 changes so that it can be merged with the main branch. Then simultaneously another person Is still working on the main branch and committing c6 changes.

We make branches so that none the person has to wait for another person to complete his commitment and then he will start, because of branching every person can make a copy of the main branch and can start working on it, later on, these branches features can be merged with the main branch.



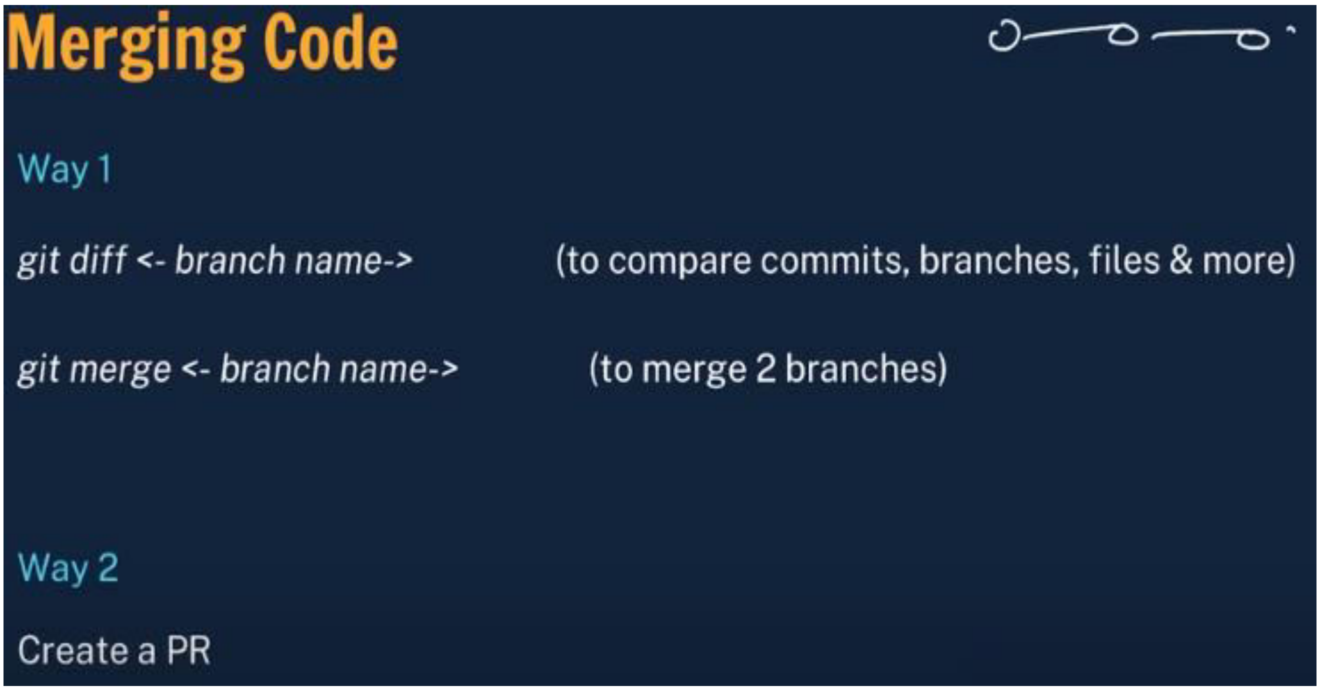
Branch Commands

```
git branch              (to check branch)

git branch -M main      (to rename branch)

git checkout   <-  branch name ->            (to navigate)

git checkout -b  <- new branch name ->       (to create new branch)

git branch -d  <-  branch name ->            (to delete branch)
```

To delete branch1 the current position mustn't be branch1 means, If (git branch) is showing branch1 then we cannot delete branch1, we have to move to another branch by (git checkout another_branch), now we can perform (git branch -d branch1).

To save changes done on branches we have to do (git push origin branch_name).

Now we have to merge our branch features and main features.



Merging Code

Way 1

git diff <- branch name-> (to compare commits, branches, files & more)

git merge <- branch name-> (to merge 2 branches)

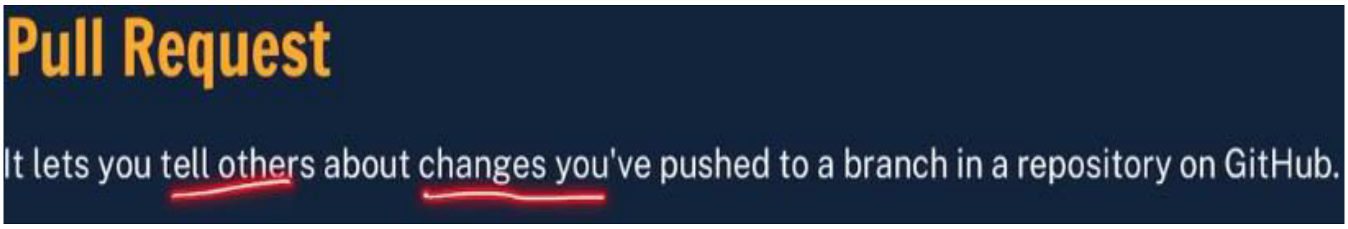
Way 2

Create a PR

Let's say you are at branch1 so when you write (git diff main) it will tell what the different features are in the main branch with respect to the current branch(branch1).

Now If we have to merge our current branch to another branch the we will do (git merge another_branch_name).

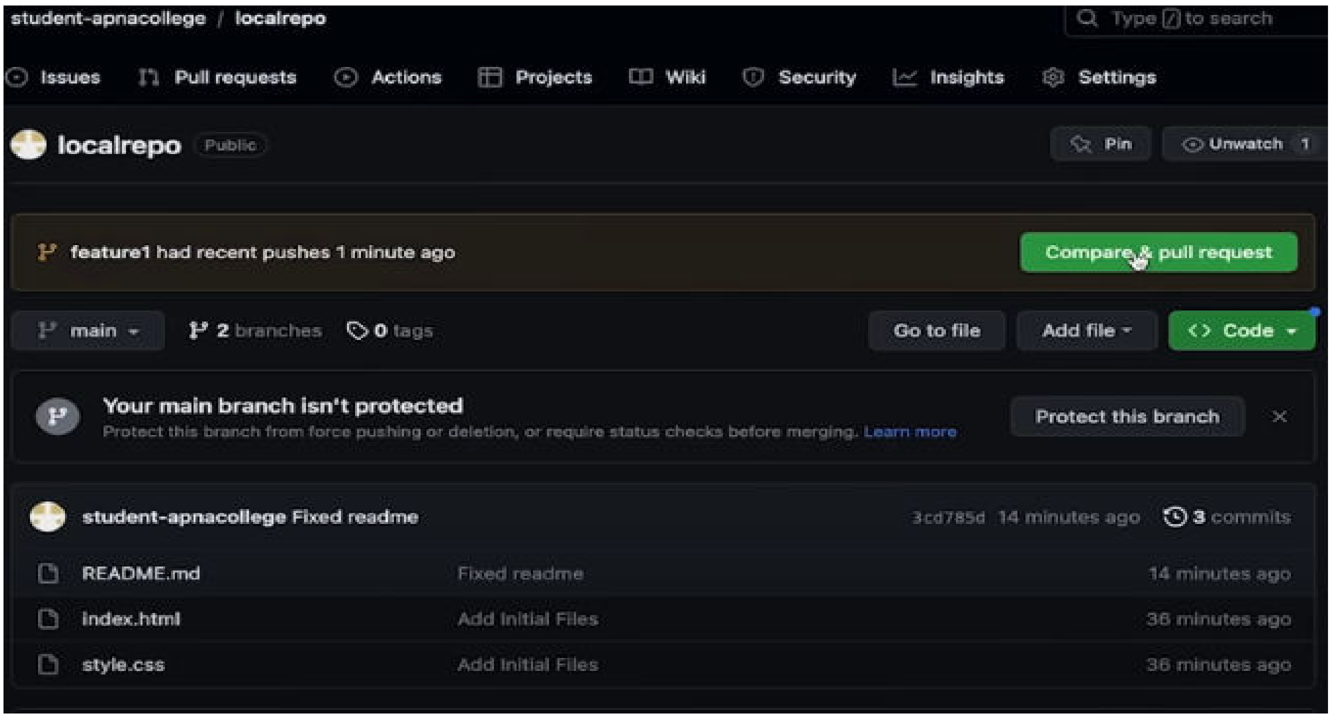
We can also merge two branches by using GitHub website, so first we have to create a Pull Request (PR).



Pull Request

It lets you tell others about changes you've pushed to a branch in a repository on GitHub.

The person wants to merge his branch with the main branch, so he makes a PR through the GitHub website. The request is accepted by the another senior developer.
process goes as:-
1) whenever a person push his code on the feature branch then the repository will look like this:-



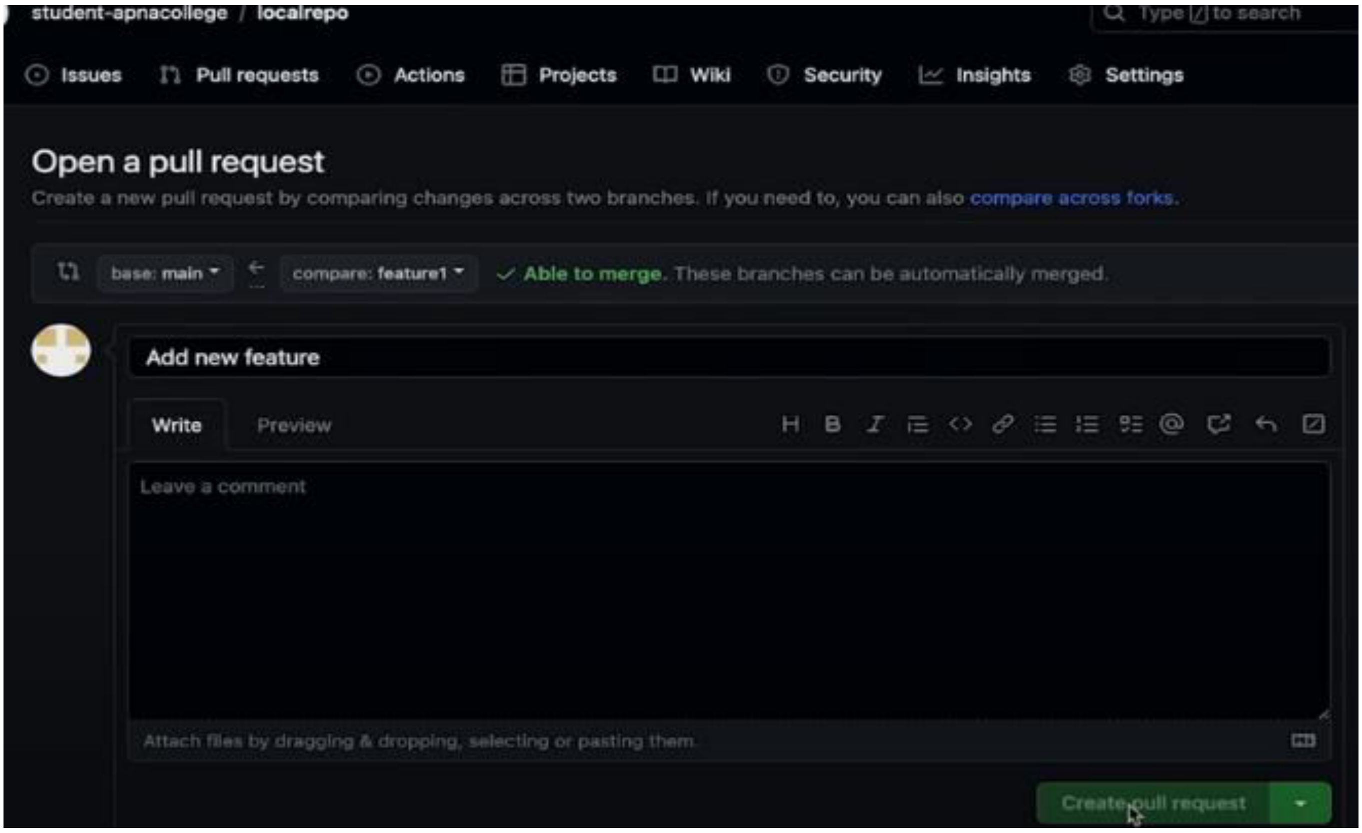
feature1 had recent pushes 1 minute ago

Your main branch isn't protected
Protect this branch from force pushing or deletion, or require status checks before merging. Learn more

student-apnacollege Fixed readme   3cd785d 14 minutes ago   3 commits

```
README.md     Fixed readme        14 minutes ago
index.html    Add Initial Files   36 minutes ago
style.css     Add Initial Files   36 minutes ago
```

This compare and pull request button will come, then after clicking, this will come:-


student-apnacollege / localrepo

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.

base: main ← compare: feature1 ✓ Able to merge. These branches can be automatically merged.

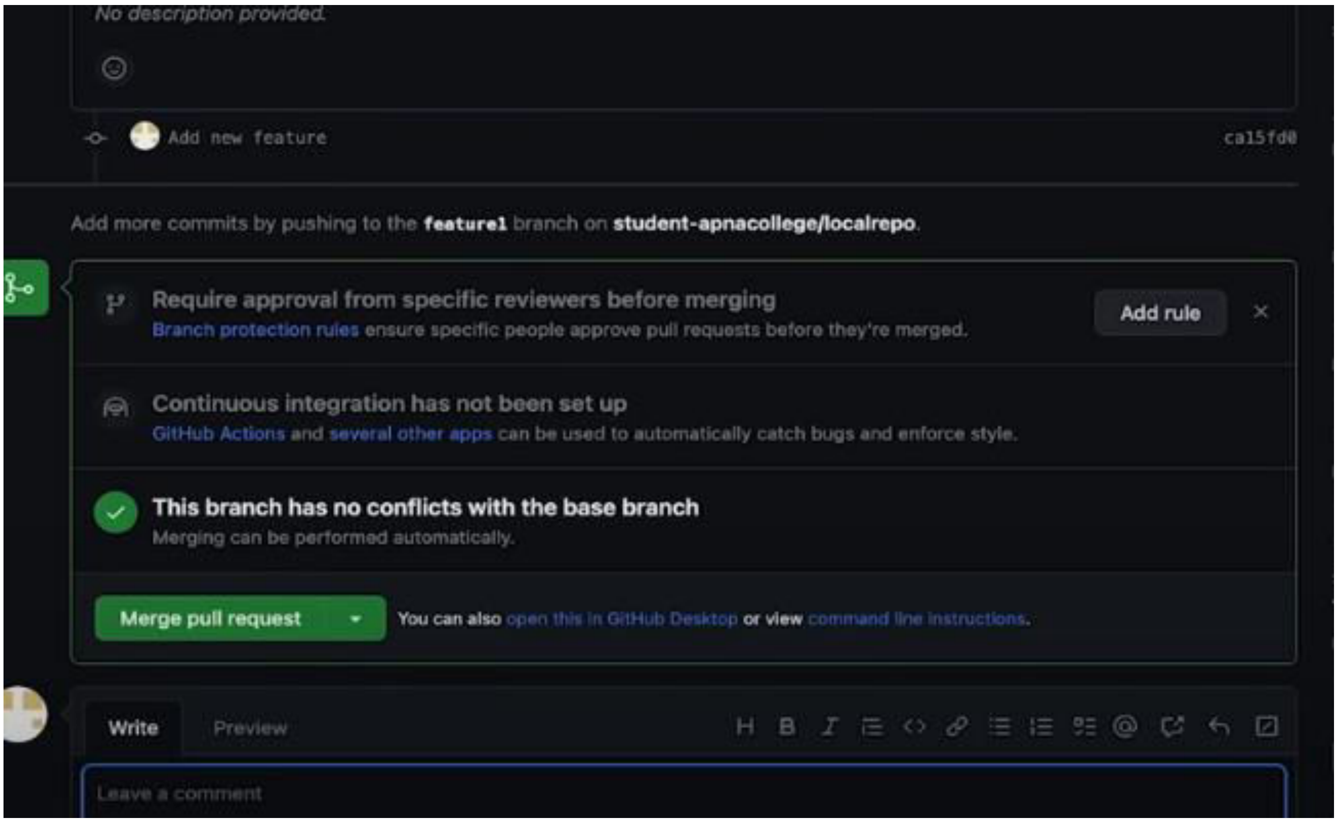
Add new feature

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

This page tells us that we are requesting to add our feature1 branch to the main branch, and we create the PR by clicking on the create PR button.

The developer who has to accept the PR, so on his side the page will look like this:-


No description provided.

Add new feature ca15fd0

Add more commits by pushing to the **feature1** branch on **student-apnacollege/localrepo**.

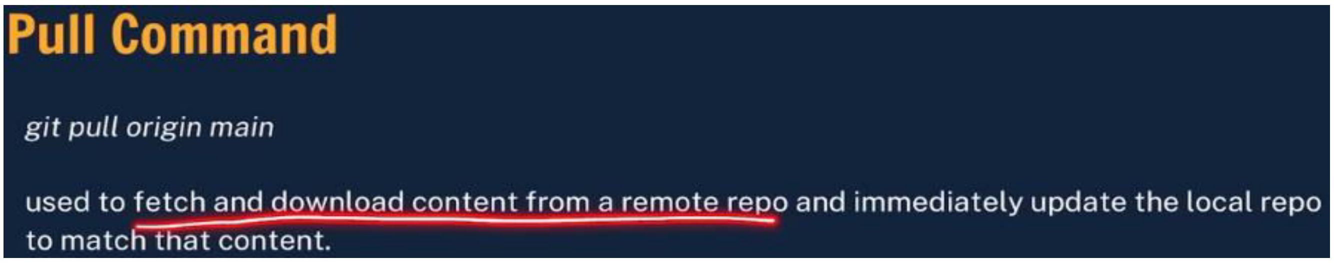
Require approval from specific reviewers before merging
Branch protection rules ensure specific people approve pull requests before they're merged.

Continuous integration has not been set up
GitHub Actions and several other apps can be used to automatically catch bugs and enforce style.

This branch has no conflicts with the base branch
Merging can be performed automatically.

You can also open this in GitHub Desktop or view command line instructions.

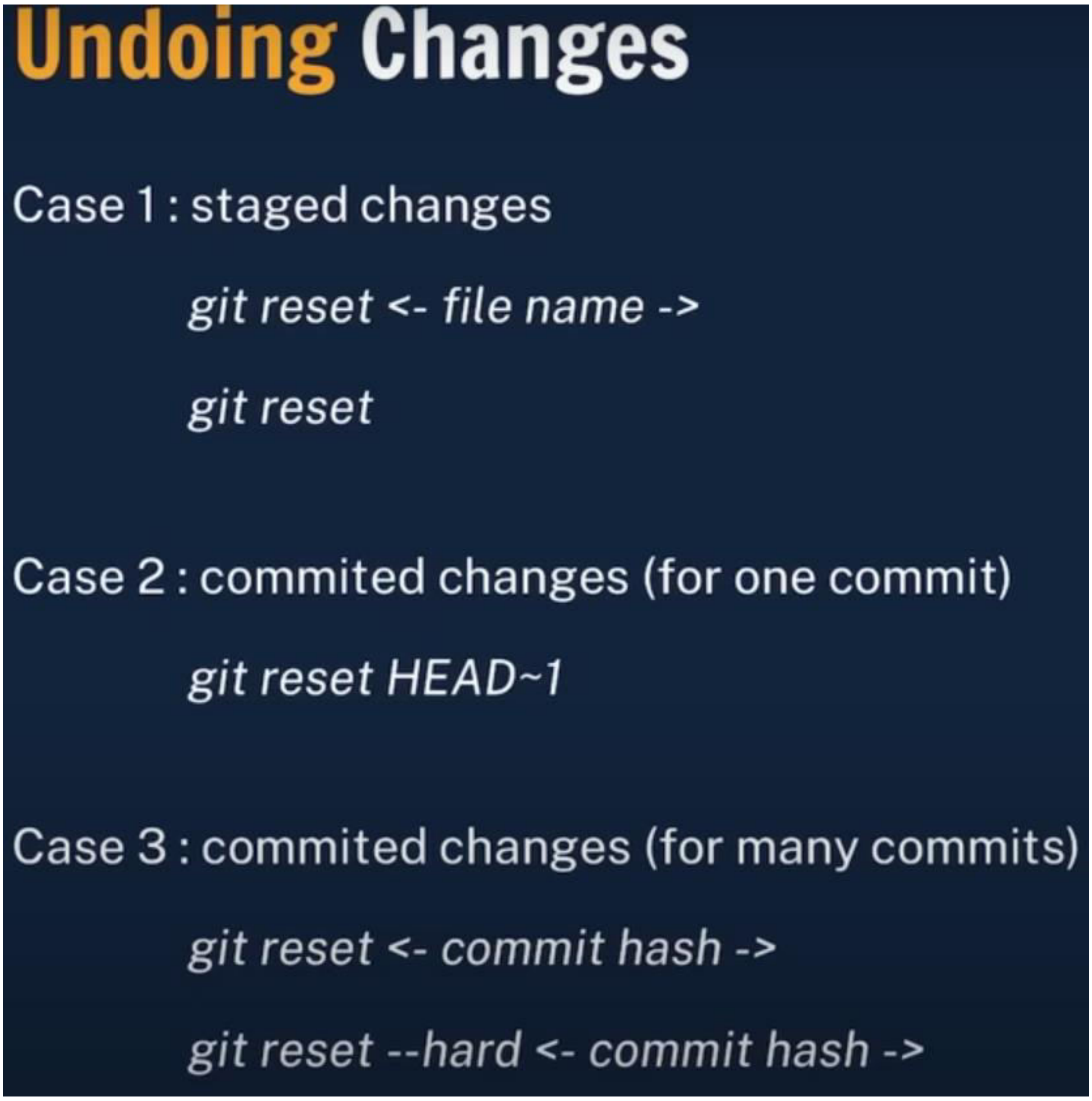
Now,


Pull Command

git pull origin main

used to fetch and download content from a remote repo and immediately update the local repo to match that content.

By doing (git pull origin main) it will the local repository and default repository(the repository on which we are storing out data) of the remote repository.


Undoing Changes

Case 1 : staged changes

git reset <- file name ->

git reset

Case 2 : commited changes (for one commit)

git reset HEAD~1

Case 3 : commited changes (for many commits)

git reset <- commit hash ->

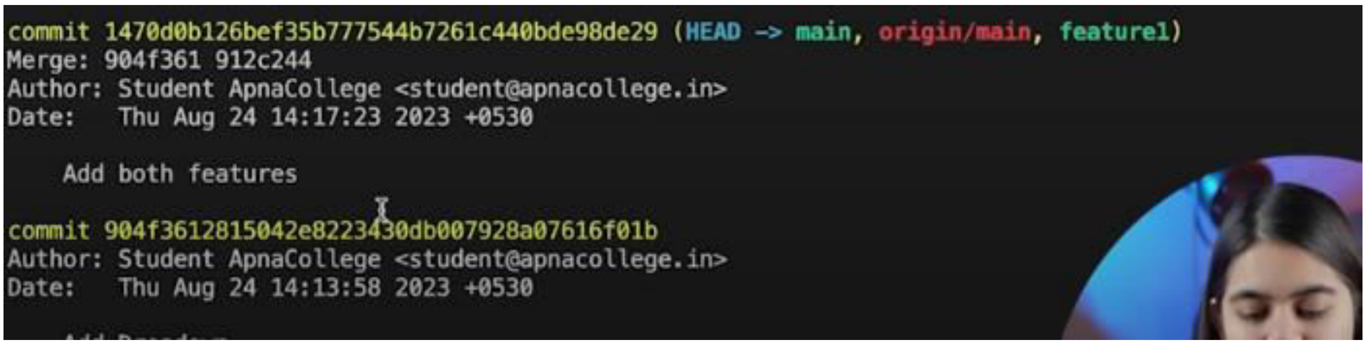
git reset --hard <- commit hash ->

(git reset filename) will reset the changes that was caused due to (git add .), means if have to undo (git add .) then I will do (git reset filename).

Now, (git reset HEAD~1) will undo the recent commit.

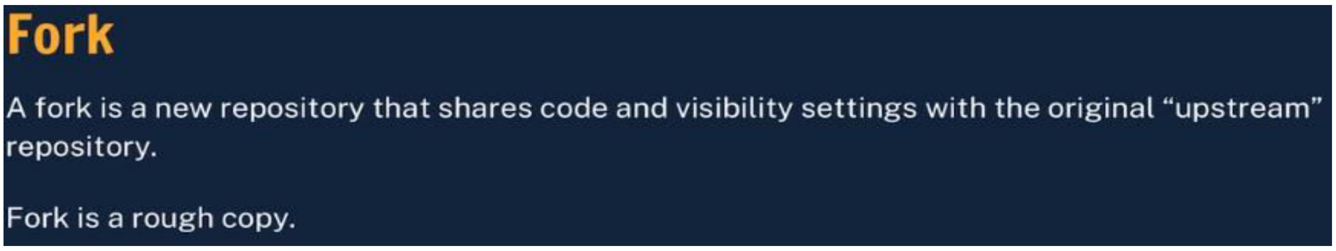
Now if you have to undo many commits let say the commits are 1→2→3→4→5, so you have done 5 commits, now you have to move back to 2nd commit, so do this:-

(git reset hash), this hash will come when you do (git log):-



```
commit 1470d0b126bef35b777544b7261c440bde98de29 (HEAD -> main, origin/main, feature1)
Merge: 904f361 912c244
Author: Student ApnaCollege <student@apnacollege.in>
Date:   Thu Aug 24 14:17:23 2023 +0530

    Add both features

commit 904f3612815042e8223430db007928a07616f01b
Author: Student ApnaCollege <student@apnacollege.in>
Date:   Thu Aug 24 14:13:58 2023 +0530
```

This yellow color code except commit is the hash.



Fork

A fork is a new repository that shares code and visibility settings with the original “upstream” repository.

Fork is a rough copy.

This will make a copy of the repository into you own collection of repositories.