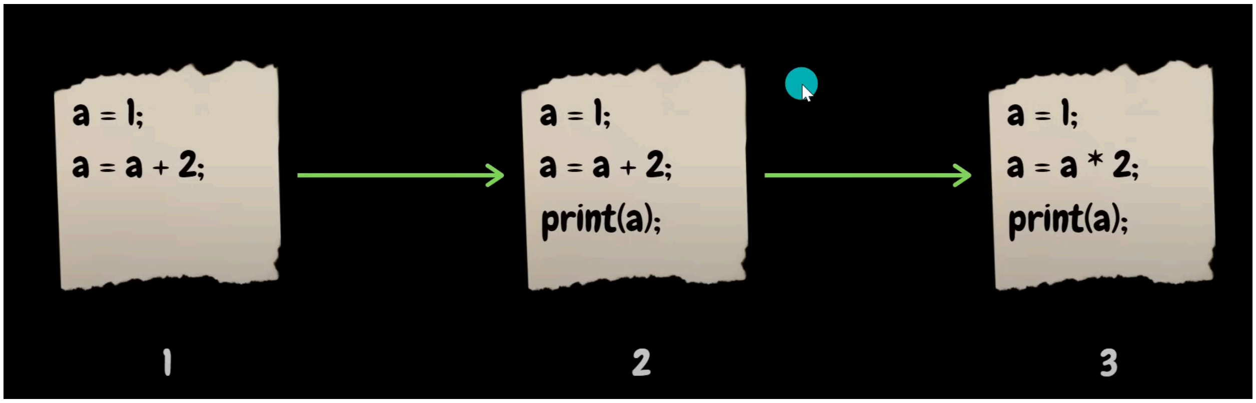
Git :

It is a **version control** tool
Means? 🤔

To **keep track of our code**, to maintain our changes.



```
a = 1;
a = a + 2;
```

```
a = 1;
a = a + 2;
print(a);
```

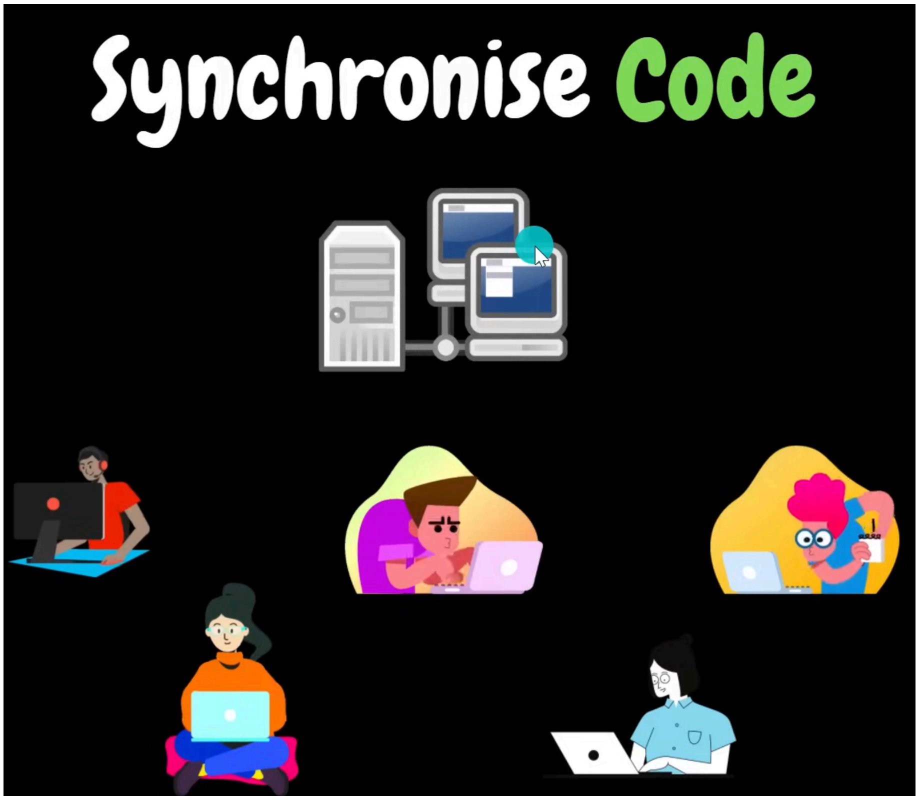
```
a = 1;
a = a * 2;
print(a);
```

Here our code changed
1 is 1st **check point** (saved progress) and so on

Also **GIT** is used to **synchronise Code :**

- Many people will work on the same project
- So every code will be saved in 1 place
- That is **repository**
- **In short, many people get code, make changes and then push.**

Git = tool => to track changes in code.
GitHub = website => to store and share code that Git tracks.



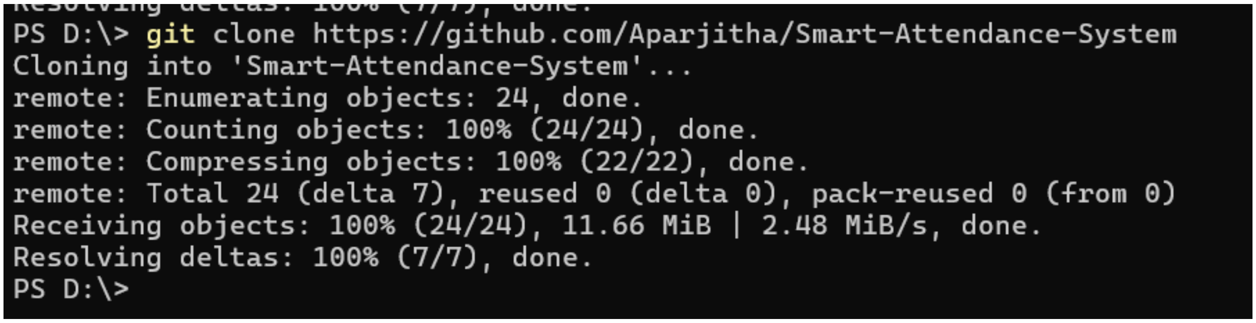
Commands in Git :

Git clone :

Project on repo —-> Local download

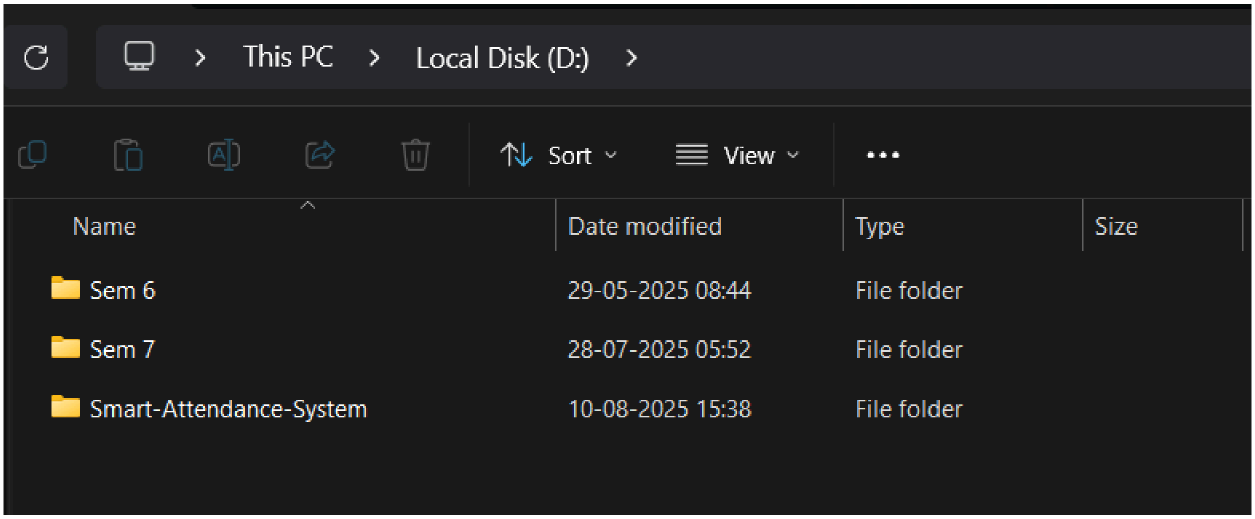
Git clone <url>

git clone https://github.com/Aparjitha/Smart-Attendance-System



```
Resolving deltas: 100% (7/7), done.
PS D:\> git clone https://github.com/Aparjitha/Smart-Attendance-System
Cloning into 'Smart-Attendance-System'...
remote: Enumerating objects: 24, done.
remote: Counting objects: 100% (24/24), done.
remote: Compressing objects: 100% (22/22), done.
remote: Total 24 (delta 7), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (24/24), 11.66 MiB | 2.48 MiB/s, done.
Resolving deltas: 100% (7/7), done.
PS D:\>
```

And we got our local copy in D drive

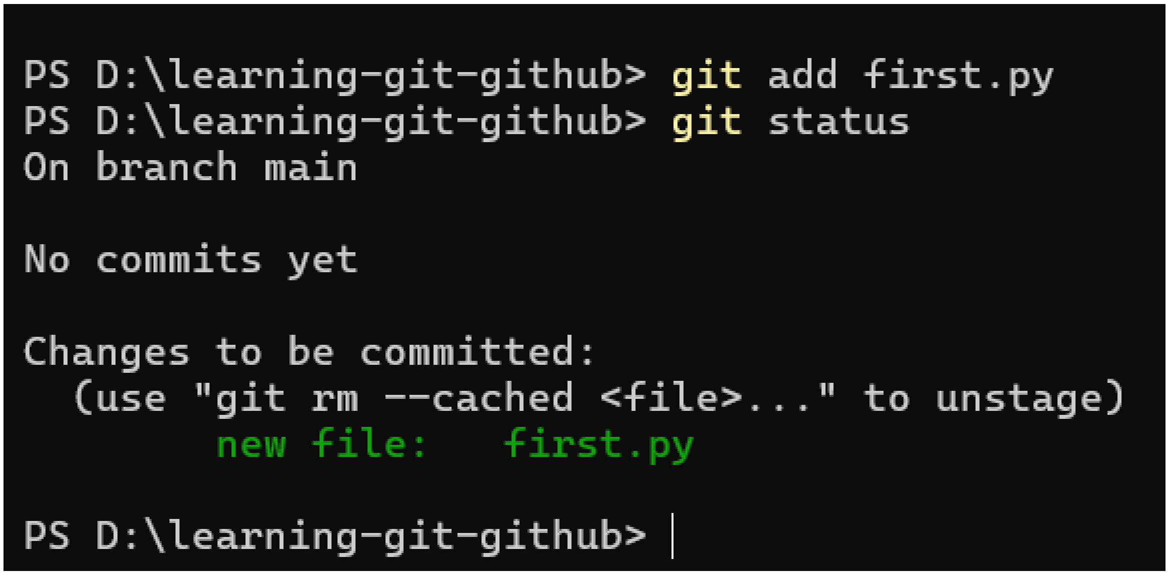


Name	Date modified	Type	Size
Sem 6	29-05-2025 08:44	File folder	
Sem 7	28-07-2025 05:52	File folder	
Smart-Attendance-System	10-08-2025 15:38	File folder	

Now open the file and make any changes needed

Then ????
We need to add the file to repo

Git add



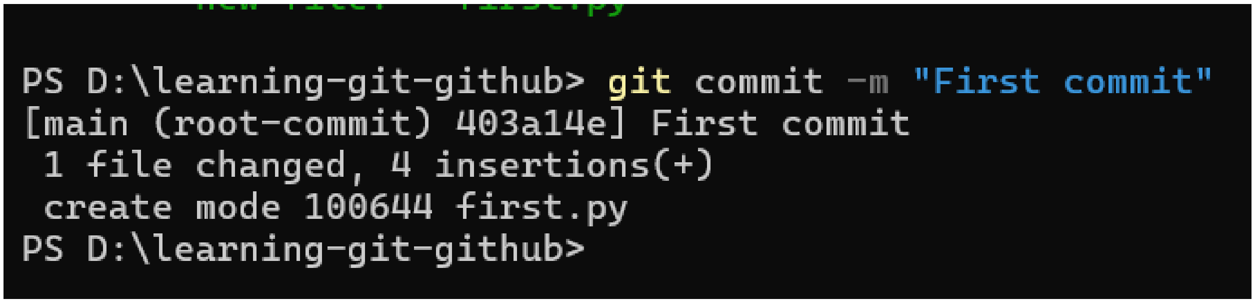
```
PS D:\learning-git-github> git add first.py
PS D:\learning-git-github> git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   first.py

PS D:\learning-git-github>
```

Now we said to git - Hey, git kid, track my files

Git said, Yes I will track yours
Say Commit when you finish updating

Git Commit :

Now we say to git, commit
Git commit -m “message”

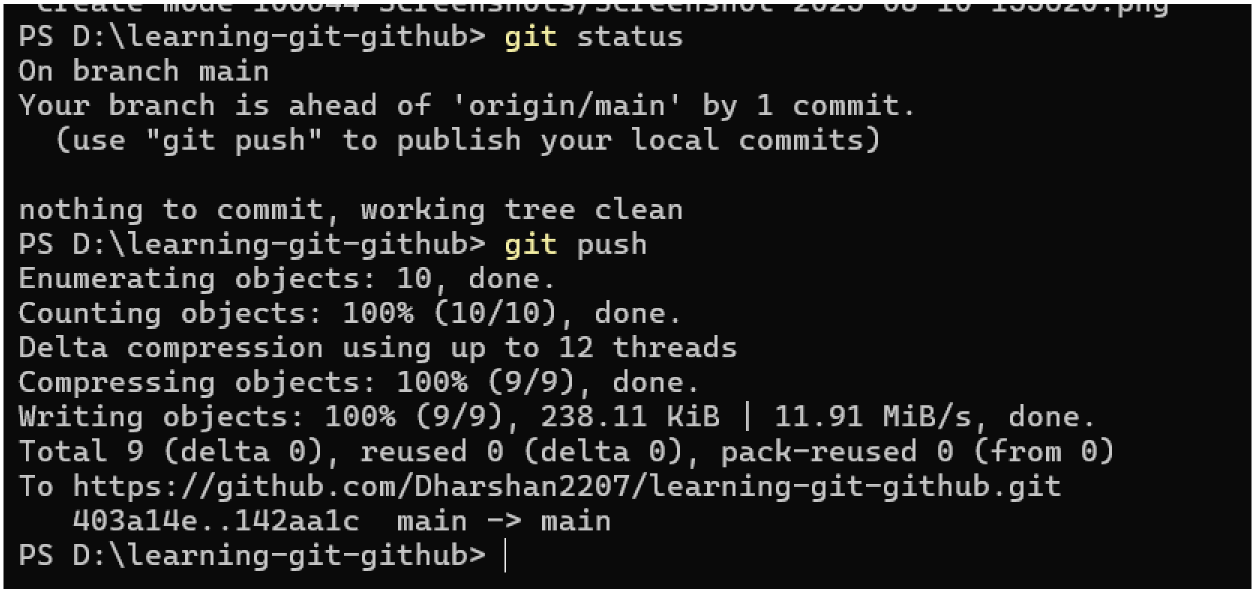


```
        new file:   first.py

PS D:\learning-git-github> git commit -m "First commit"
[main (root-commit) 403a14e] First commit
 1 file changed, 4 insertions(+)
 create mode 100644 first.py
PS D:\learning-git-github>
```

Now what???
Changes saved in our local device, and it’s confirmed (commit)

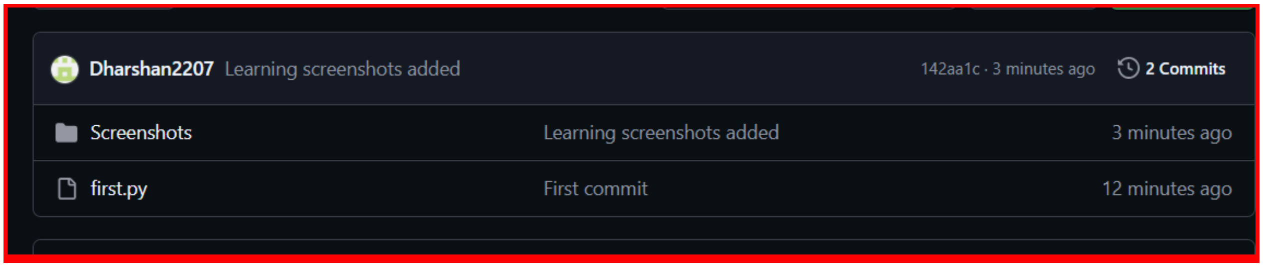
Git Push :

Its like pushing the changes from **local -> repo**



```
 create mode 100644 Screenshots/Screenshot 2025-08-16 153020.png
PS D:\learning-git-github> git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
PS D:\learning-git-github> git push
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 12 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 238.11 KiB | 11.91 MiB/s, done.
Total 9 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Dharshan2207/learning-git-github.git
   403a14e..142aa1c  main -> main
PS D:\learning-git-github> |
```

Git status -> shows **how many commits** done
Just enter **git push**



Dharshan2207 Learning screenshots added		142aa1c · 3 minutes ago 2 Commits
Screenshots	Learning screenshots added	3 minutes ago
first.py	First commit	12 minutes ago

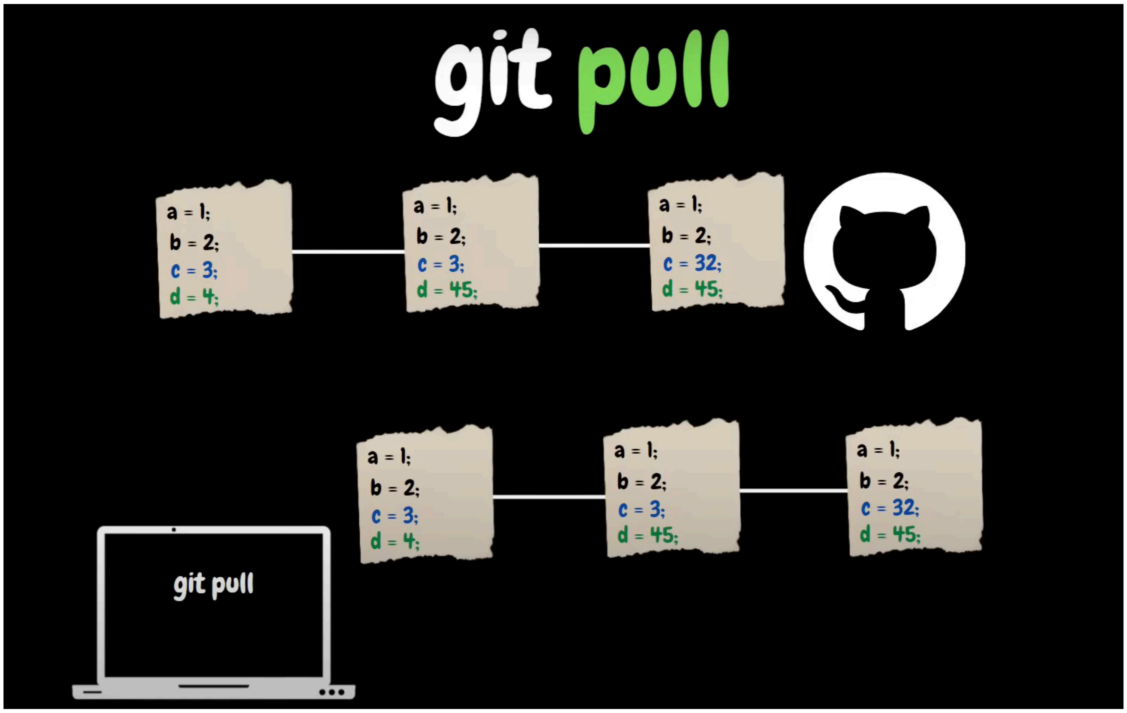
Those texts in the **center = commit messages** of the last commit we or someone made.

Like this, we can detect who tf made some silly changes 😤

Git Pull :

Opposite of push 👍

We pull file from repo, those changes will be applied to our local last committed file .



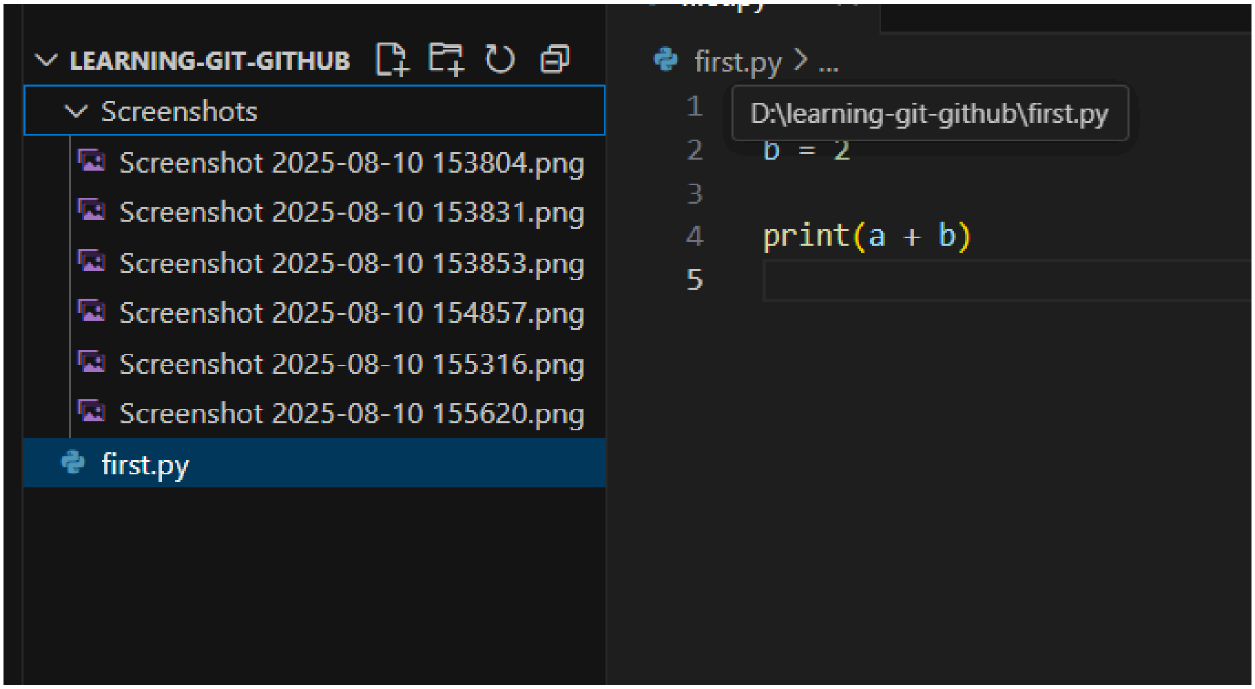
Some mf or our team member made changes our file on git

Real life example :

Me and 4 friends work on a team project building an app

First day, 1 mf made changes and pushed.

Now if i want to continue the project, i need to get the last updated code using (git pull)


LEARNING-GIT-GITHUB
Screenshots
Screenshot 2025-08-10 153804.png
Screenshot 2025-08-10 153831.png
Screenshot 2025-08-10 153853.png
Screenshot 2025-08-10 154857.png
Screenshot 2025-08-10 155316.png
Screenshot 2025-08-10 155620.png
first.py
first.py > ...
D:\learning-git-github\first.py

```
b = 2

print(a + b)
```

Now, all changes applied locally on our device.

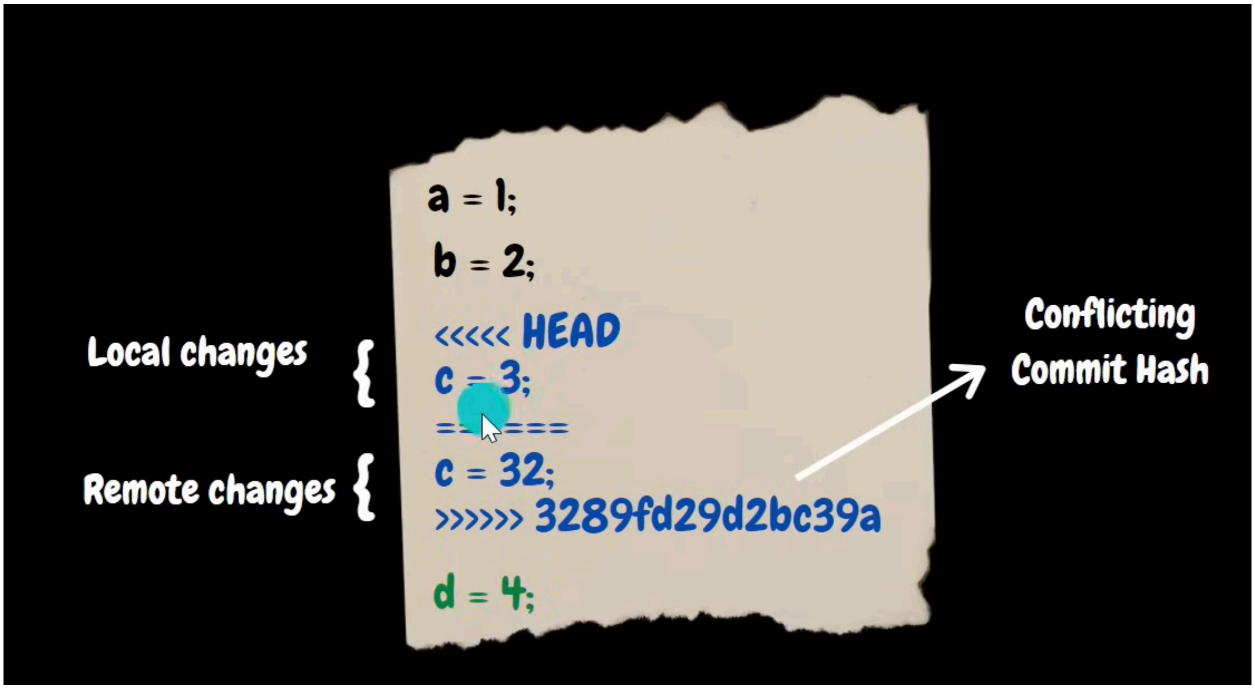
So that we can continue writing new code for group project.

But ??? 🤔

When we pull, what if merge conflicts happens?
Like i set
C = 20
Another mf sets
C = 10

Git is confused 😭

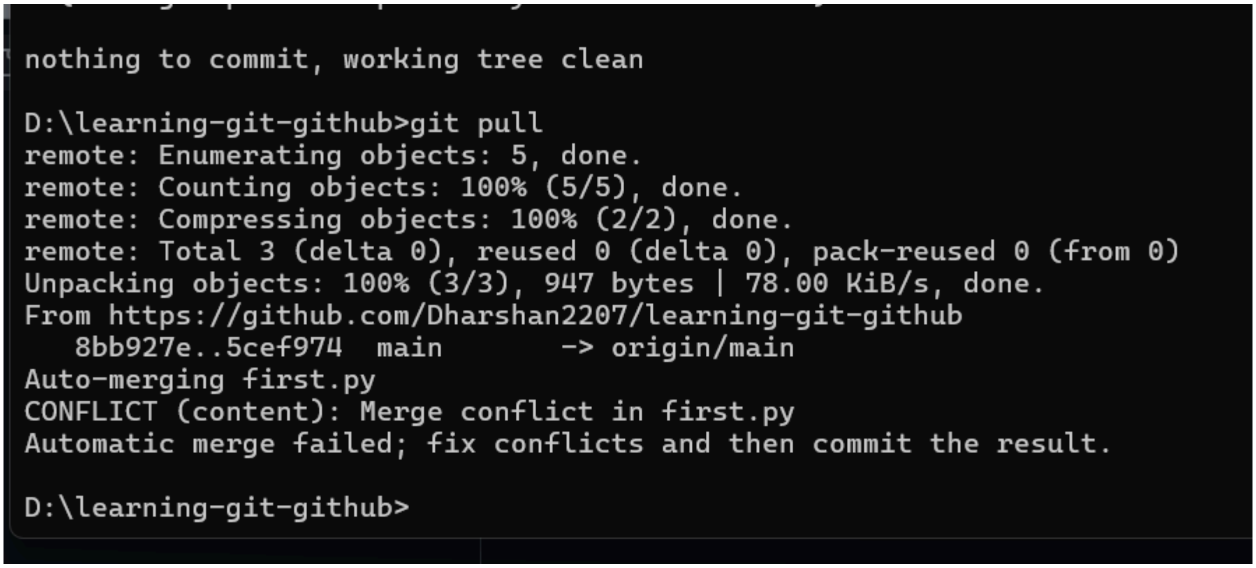
So it gives us something like this,



```
a = 1;
b = 2;
<<<<< HEAD
c = 3;
=======
c = 32;
>>>>>> 3289fd29d2bc39a
d = 4;
```

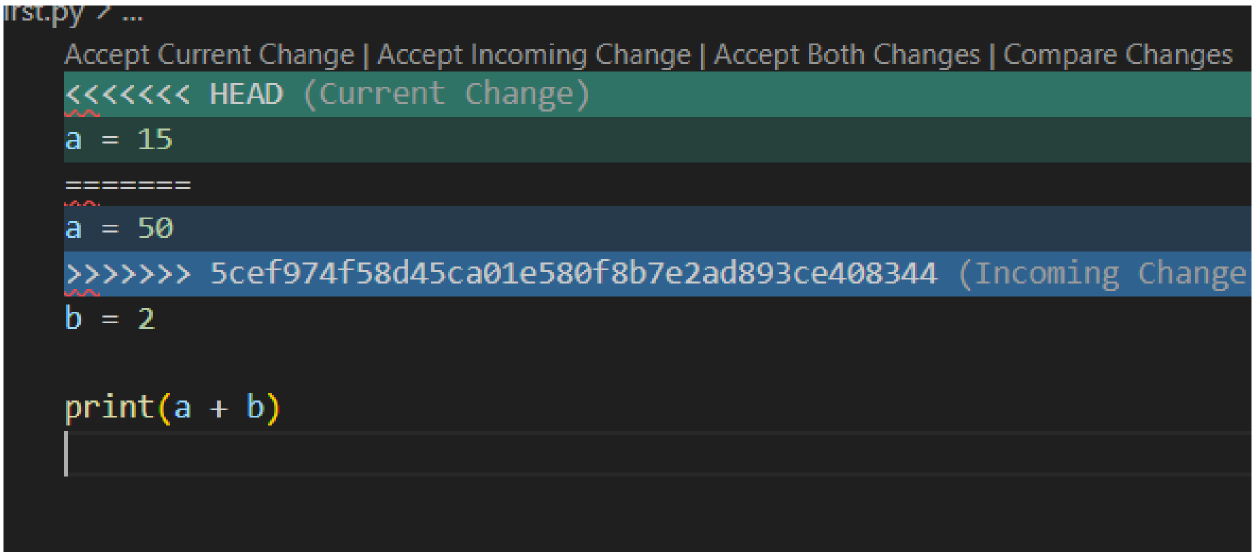
In local we have
A = 10
In repo we have
A = 50

When we pull ???



```
nothing to commit, working tree clean

D:\learning-git-github>git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 947 bytes | 78.00 KiB/s, done.
From https://github.com/Dharshan2207/learning-git-github
   8bb927e..5cef974  main       -> origin/main
Auto-merging first.py
CONFLICT (content): Merge conflict in first.py
Automatic merge failed; fix conflicts and then commit the result.

D:\learning-git-github>
```

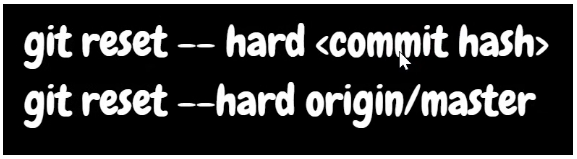
It shows merge conflict


irst.py > ...
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes

```
<<<<<<< HEAD (Current Change)
a = 15
=======
a = 50
>>>>>>> 5cef974f58d45ca01e580f8b7e2ad893ce408344 (Incoming Change
b = 2

print(a + b)
```

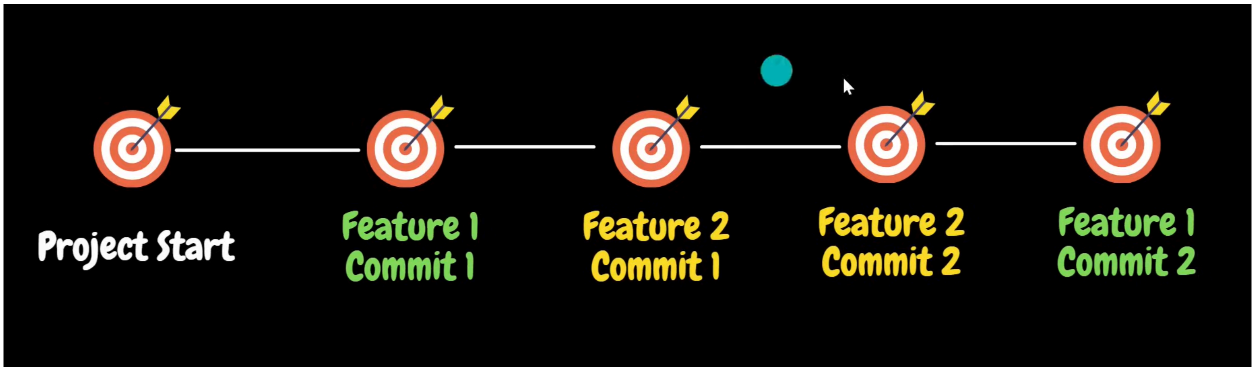
And shows the conflict 😤

Git reset :

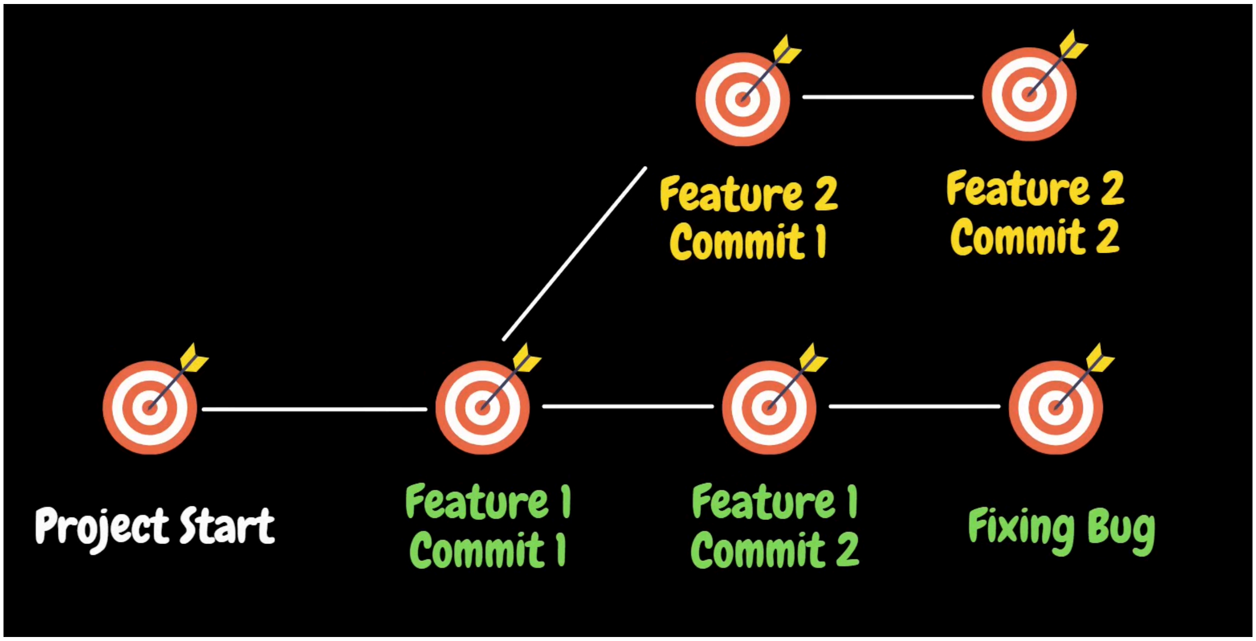


```
git reset -- hard <commit hash>
git reset --hard origin/master
```

Branching :



Linear ? fixing bug is harder,
Useless stuff in between

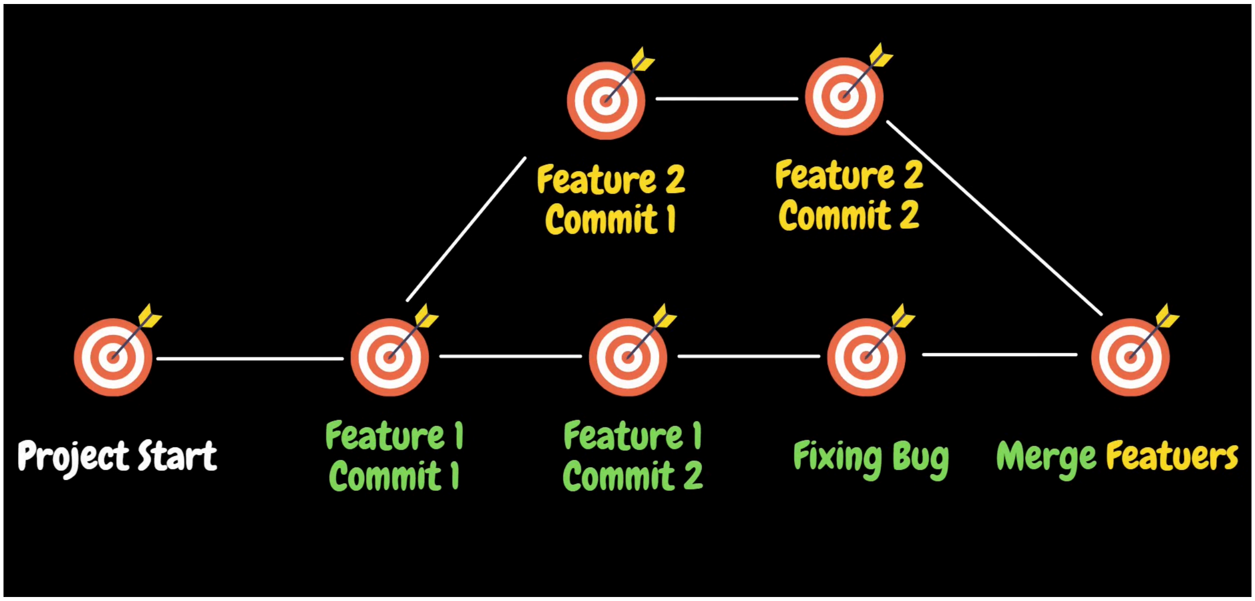


Tree ?
Easy to go to respective branch

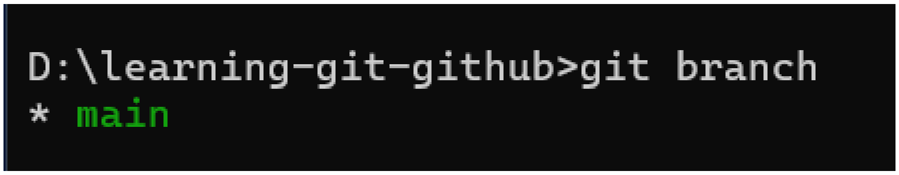
Main branch = Master branch
We can create own branches too 🥳

Branch changes ? Just move Head pointer to the corresponding branch 👍

Al last, we can merge the branches 🔥

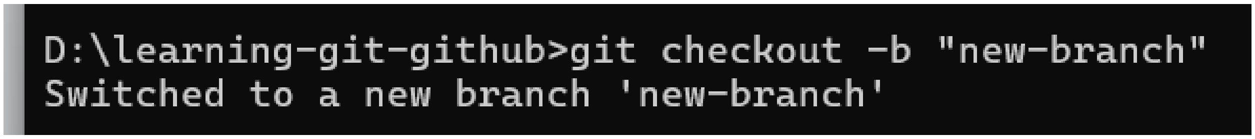


Check current branch ?



```
D:\learning-git-github>git branch
* main
```

Create new branch ?



```
D:\learning-git-github>git checkout -b "new-branch"
Switched to a new branch 'new-branch'
```

After making changes in **2 different branches**, we need to **merge** those 😭

Go to main branch
Git merge new-branch
That’s it 👍