Class 4 :- Git version Control (Contd.)

→ Problem on the know before the git : not able to track project but after the Git exitera locally track projects

→ Git store all track text file in .git directory

① Ct Create a JS file and add Code which add two number

index.JS

```
1. function add (a,b) {
2.         return a+b;
3.     }
4.
5. add (2,5)
```

* How i know is this file tracked or not
  ↳ To check the Run Command (git status)
  ↳ If git status failed not tracked your file
    ↳ To Run (git init)
      ↳ This Command just by — my start to observe/track my project / file :
      ↳ after this Command Run now Run (git status) and you go Something like this

```
On branch main
No Commit yet ...
```

* Git not track all file by default
So just Say [git] hey I have a file track that or put it on staging area.

↳ To put In Staging Area Run Command

```
git add index.JS
```

* Even if you have lot of file then you just say to Git hey put all that file in staging Area Run Command

```
git add .
```

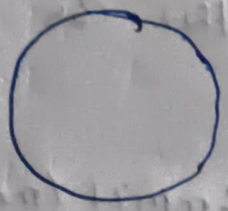
* After it put on staging Area we need to say to [Git] hey now it is time to save or snapshot, Git Carefully as you track any file or Project

↳ To take Save or Snapshot Run [Commit] Command

```
git Commit -m "message"
```

Commit is like memory.

* but assume Commit in shape of round memory



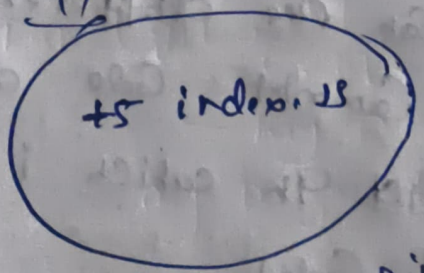
Beside this memory is all memory till now (git add. with 5 line Code)

① git give a ID that you assign each Commit

② you can see that ID with git log

③ With Commit author name date also attached with that Commit and also date (when you perform this Commit action.

* So Now we have see git Commit which is actualy memory that have many data/ Payload



add index file
with 5 lines of Code
1Ca CaF6 ff· ——— ..
Author : piyush Garg <piyushgarg.dev@gmail.com>
Date : Thu Jan 22 21:29:44 2026 +0530

Tree object
Commit object

* These commit store in (.git) directory and

→ is that .git directory any other can be access — No

↳ .git means this commit (memory) which is in .git directory that is Locally store (Local History)

* Update index

```
1. function add (a,b){
2.     return a+b
3. }
4.
5. add(2,5)
6. add (10,23)
```

→ Just add one line not put on staging or not Commit ,just add a like of Code.

→ Now we can see difference b/w my last Commit Code and Current Code (which in which i just add a line of Code)

git diff

↳ ~~these difference b/w : Current Code which is unstaging and Commited Code , and then we get difference~~

git diff

↳ Those differences b/w, committed code and unstaging code (current code), and Boom you can see difference.

git diff

=> -add(2,5)
\ No newline at end of file
+add(2,5)
+add(10,23)
\ No newline at end of file

⇓

But we can clearly see git considered that remove `-add(2,5)` and the add `+add(2,5)` `+add(10,23)`

⇓

Why these changes git consider, actully this `add(2,5)` also changes.

Let Check these fact

① previous commit at the last line of code is `add(2,5)`

② When I was adding `add(10,23)` basically means go to line 5 to line 6 then that time at line 5 we just press enter button which means I said to VSCode (Hey I don't want go to next line), so this next line permission also include.

or Changes track by git. (\n → tracked)
and 2nd changes track to add [add (10, 23)].

Boom mystory has been solved.

1st [add (2, 5)]

when add new code in next
line to add (add (10, 23)).
add (2, 5) —to→ [add (2, 5) \n] → next line

Note :- when you use git then you must
need to leave one line.
to avoid [\n] Changes leave last line
empty.

* Cut Click Photo to my Changes

git Commit -m "add one more function Call for add"

→ git message must be Present.

→ [main c03fb8] add more function Call for add
1 file Changed, 2 insertion(+), 1 deletion(-)

* Now when you git log, you see two commit history.
* So new commit means new memory has been created

add one more function Gll for add

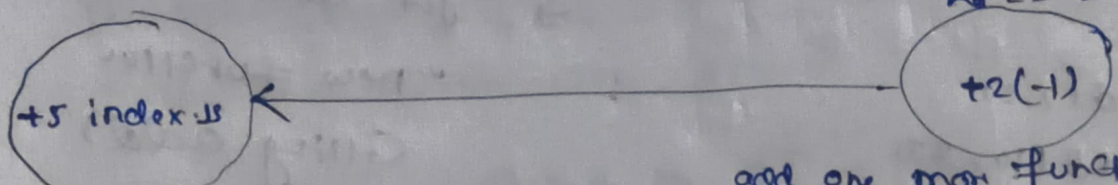
Commit ---

Author : Piyushgang <piyushgang.dev@gmail.com>

Date : Thu Jan 22 21:38:42 2026 +0530

Parent - 1ca0af6ff ---

↳ you can see in previous commit no any parent bcz in this commit is after the previous commit ((visku, you repeat same thing here ha--- ha--- ha--- ha--- .)



add index file with 5 lines of code

1ca0af6ff --- 12d3

Author : Piyush gang <piyushgang.dev@gmail.com>

Date : Thu Jan 22 21:29:54 2026 +0530

add one more function Gll for add

Commit 3b89 ---

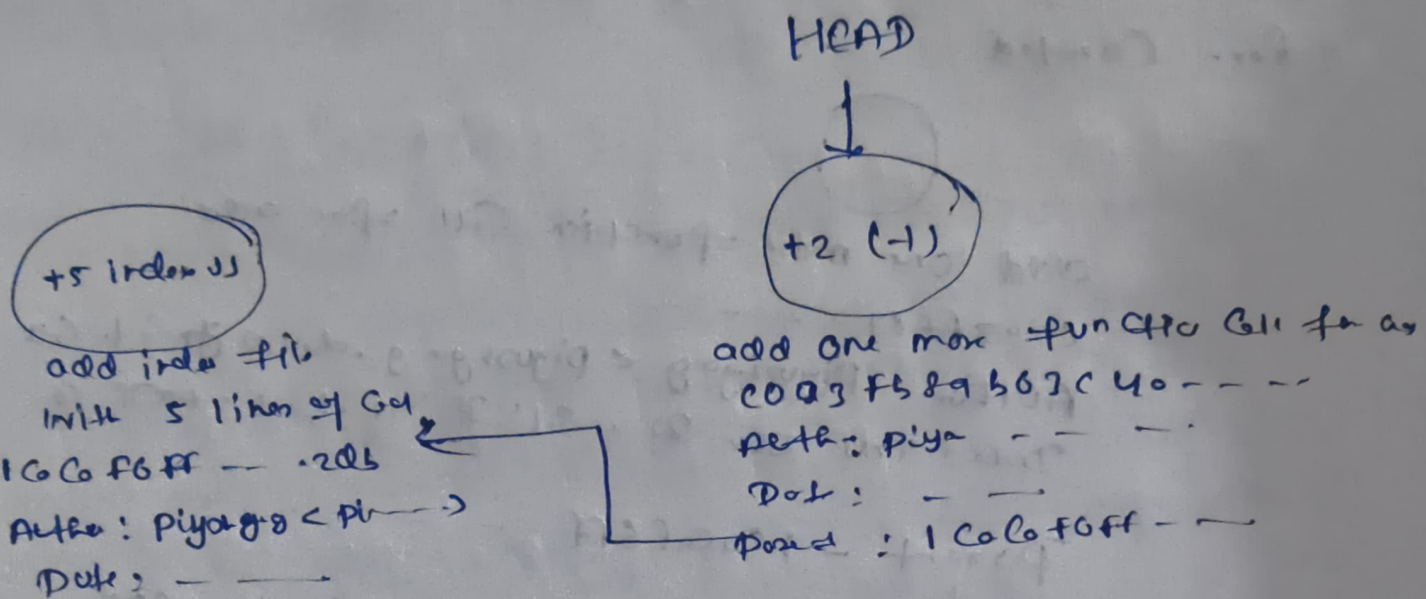
Author : Piyush Garg <piyushgarg.dev@gmail.com>

Date : Thu Jan 22 21:38:42 2026 +0530

Parent - 1ca0af6ff ---

* But how git know latest Commit ?

⇒ Git maintain HEAD and these HEAD has details of latest Commit.



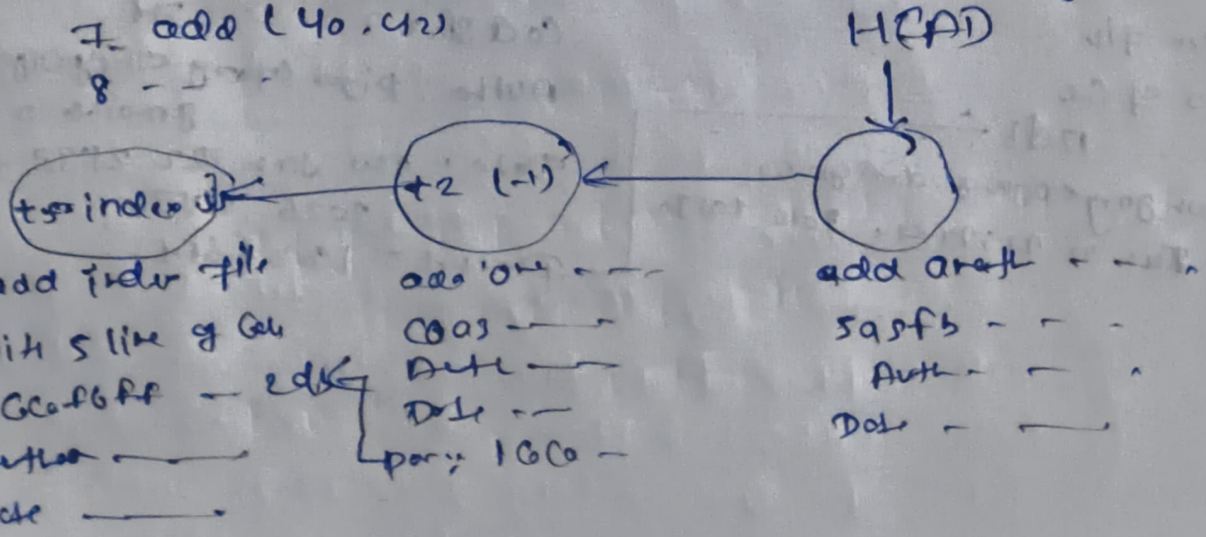
* Let add one line of Code in the last

index.JS

```
1. function add (a, b) {
2.         return a+b;
3.     }
4. - -
5  add(2,5)
6  add(10, 23)
7. add(40, 42)
8 - -
```

⇒ git add index.JS

⇒ git Commit -m " new function Calling add()



* ~~of~~ ~~the~~ Commit HEAD ~~has~~ ~~been~~ moved

* It also happen we put our project on cloud (single point of truth). not along with (.git) directory

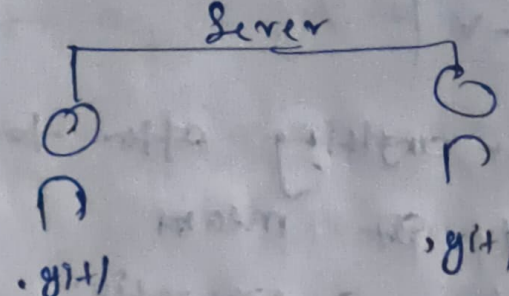
GITHUB
↳ A Server or ~~that~~ Git Installed

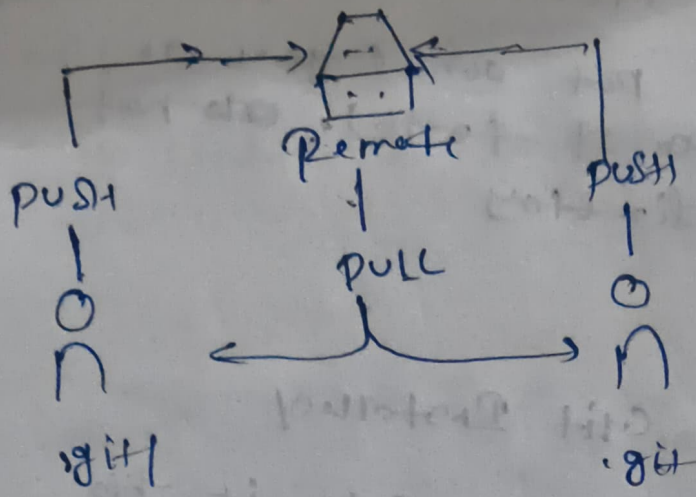
* Many Server that has install Git in ~~his~~

Server
- ↳ GitHub
- ↳ BitBucket
- ↳ AWS CodeCommit
- ↳ AZURE
- ↳ Gitea

* ~~Let~~ two Person and both work on Same Project, ~~but~~ .git directory has both Person and ~~he~~ decide to keep a Server and both push on server.

Note Server Called as Remote





→ Both has decide to PUSH and Pull from same Remote.

(1) This solve user Collobration Problem

(2)

⇒ In my case that server is Github so we need to say to .git/, hey Put your Changes on Github/Server.

So we need to say

`git remote`

or

`git remote -v`

If you not return anything after to Run this Command it means Remote is not been Configuration.

To Config of .git Need to Run a Command

```
git remote add origin git@github.com:vijaya-dev/
```

```
git remote -v
```

-> origin git@github.com (fetch)
-> origin git@github.com (push)

* After that remote configuration need to push git Changes

the Command is

```
git push -u origin main
```

-u means which origin need to push - main branch

-u -> upstream

* Everyone can not be able to push code on github

-> only Collaborative People can be push and pull

-> If you not be collaborative not push there.

So the if a repo which has DSA Repo. then we just Copy In my account

⇓

Technically we not say Copy we just say Fork the repo.

Let's DSA Repo jo hai o hai Aniruddh Sir ka

So we need to see which is git remote -v when we Run this Command

```
git origin -v
origin  git@github.com:aniruddhuu/dsa.git (fetch)
origin  git@github.com:aniruddhuu/dsa.git (push)
```

⇓

So we Can update Remote origin.

~~git remote update origin~~

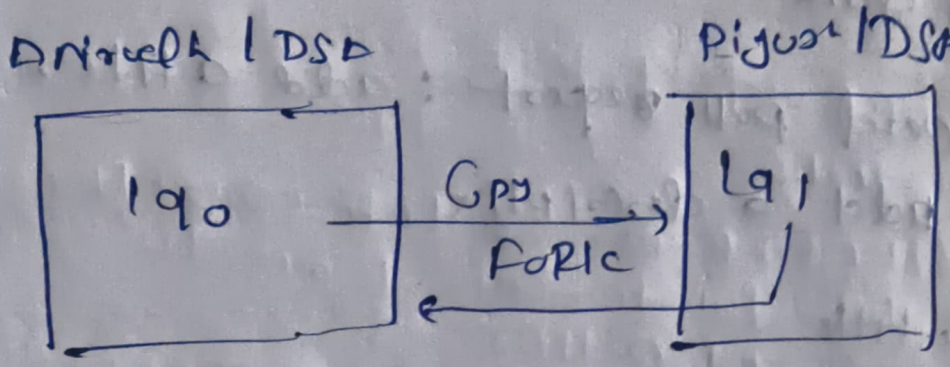
```
git remote set-url origin git@github.com:Piyushgag-dev/dsa.git
```

git remote -v

⇒ origin git@github.com:Piyushgag-dev/dsa.git

origin git@github.com:Piyushgag-dev/dsa.git

Now we can push on my remote, ~~but remote~~ ~~has~~ full access to me



* We Fork Aniruddh Sirs repo ~~Add~~ we also we sent a Pull request to Aniruddh
* Hey Aniruddh Sir

* So when i click on Pull request the we can clearly see ↓

base repository: aniruddh / dsa | base main

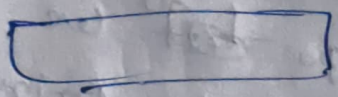
↑ :

head repository: piyushgoyal/dsa | Compare main

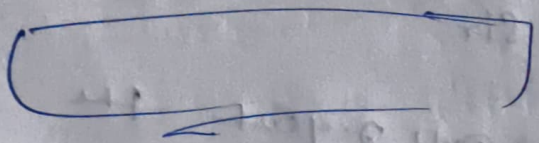
And in the bottom you can see what changes i done.

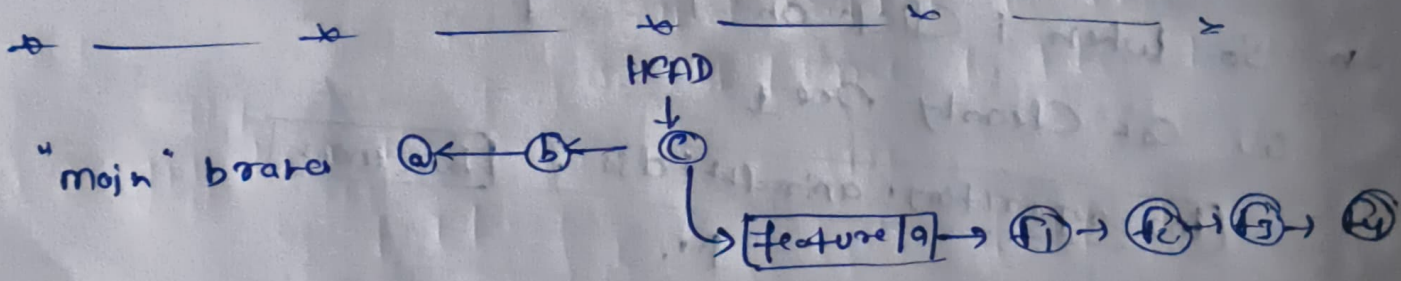
To creel pull request : add a title, and add a discription.

Add a title



Add a Discription





So the first i pull a repo and create feature/a brarch and commit f1 for problem solve and f2 and f3 and finally f4 Complete in f4 Commit.

Now I can merge feature/a to main branch.

Merging Technique is when we try to merge feature branch in main branch

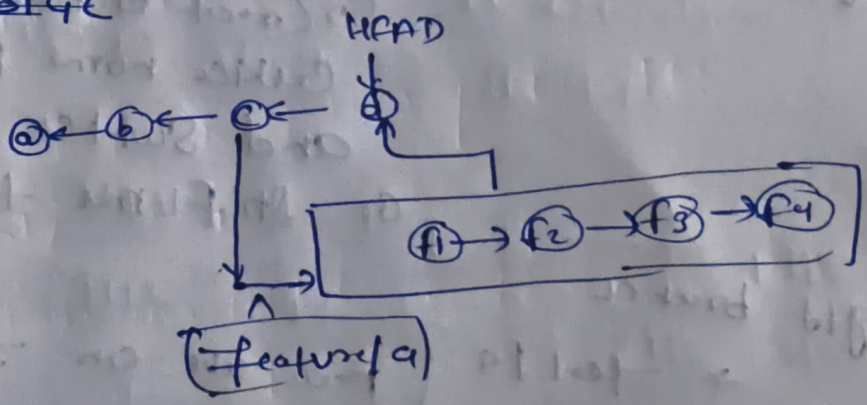
ⓓ Commit created.

→ And ⓓ is just squash of f1, f2, f3 & f4

→ So never think ⓕ4 is ⓓ.

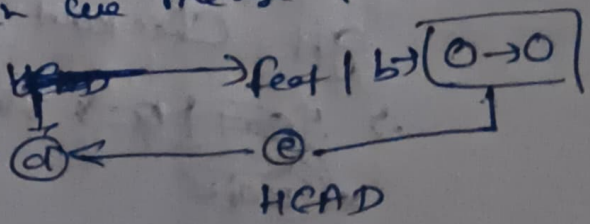
→ f1, f2, f3 & f4 का सब changes ⓓ में होगा। जिसे squash करते हैं और d का parent c होगा, but HEAD अब ⓓ होगा।

→ ~~अब d के अंदर~~



after the merge we can delete f1, f2, f3, f4

→ we also create a new branch and work on that branch and then we merge that



Note

So the everyone can make their branches and merge back to the main one

इससे सारे developer एक साथ काम कर सकते है। जब काम हो जाएगा तो वो वापस आके main में merge कर देंगे।

Cont'd Command Part

① git log --oneline

saa50dc (HEAD -> main, origin/main, origin/HEAD) add another function call to add the additional details

c0a3fb8 add one more function call for add

1caca f0 add index file with 5 line of code

② git checkout -b "feat1a"

↳ Create a new branch which name is (feat1a) and switch to this branch or shift me to this branch

③ git branch

* feat1a
  main

| now we are on feat1a branch

④ Now want to switch back to main

git checkout main

→ ~~feat1a~~ ⇒ switched to main

~~* main~~

⑤ git branch

feat1a

* main

Note : अगर merge करना है तो main में Switch होना होगा।

git branch
-> feat/a
   main

git merge feat/a
-> updating 5aa50dc -- da24103
   fast forward
      index.js | 8 ++---
      1 file changed, 8 insertions (+)

after merge Code merge but all Commit Gone to main branch.

Instead of git merge feat/a

use git merge --squash feat/a

ये Change ले आया। But अभी Staging में ही है तो करना होगा और Snapshot लेना होगा

git merge --squash feat/a
↓
git add .
↓
git Commit -m "message"

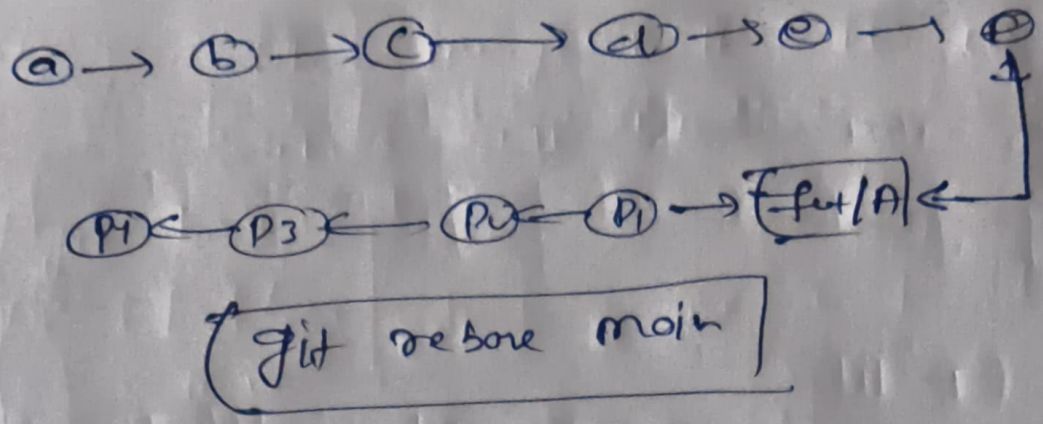
अगर हम एक branch बनाकर उसमें Code
Commit करते हैं और Commit code
को ।

अब हम इस Code को push करेंगे Remote
पर तब हमें Error आएगा। क्योंकि वो
branch तो github पर है ही न
so use this.

git push --set-upstream origin feat/multiple
↳ यह Code को Push करेगा feat/multiple
branch में और अगर branch नहीं
होगा तो create कर देगा

(Note) आप एक Project पर काम कर रहे हो
और भी कई developer काम कर रहे हैं तो
हम अभी एक (A) feature में
ही काम कर रहा पर किसी ने (B)
feature (C) feature बनाकर के merge
भी कर दिया पर हम तो अभी A feature
पर ही हैं (B) और C का तो हमें पता ही
नहीं है।
इसलिए आता है यहाँ Rebase

[git rebase main]



git version Control → rewatch ~~ckt~~ from 2:00 :00

Saas ~~Aol~~ give money ~~from~~ ~~featur~~
~~that~~ give money ~~from~~ maintance