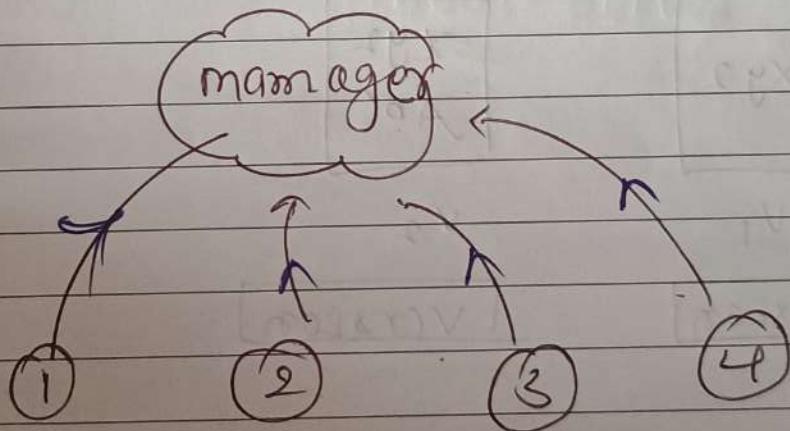


Introduction to git

{ Software Configuration management

{ software / some code management

आज ते ३०/२० साल पहले थे system नहीं था
उस time में manually ही code
रखा पड़ता था



1 } 2 } 3 } 4 } ये लब [mamage] को code में भी

mamager के form difficult हो जाए
रखना तो ③ ने भी किया था.

कृति समी problems को solve

करने के लिए इसे solver

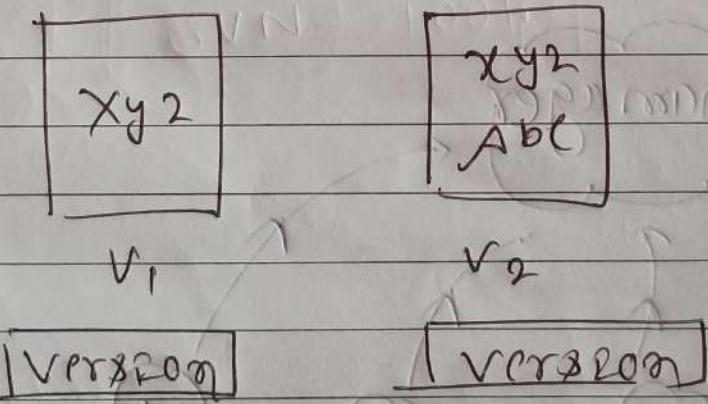
Tool design किये प्रस्तु

कृति &

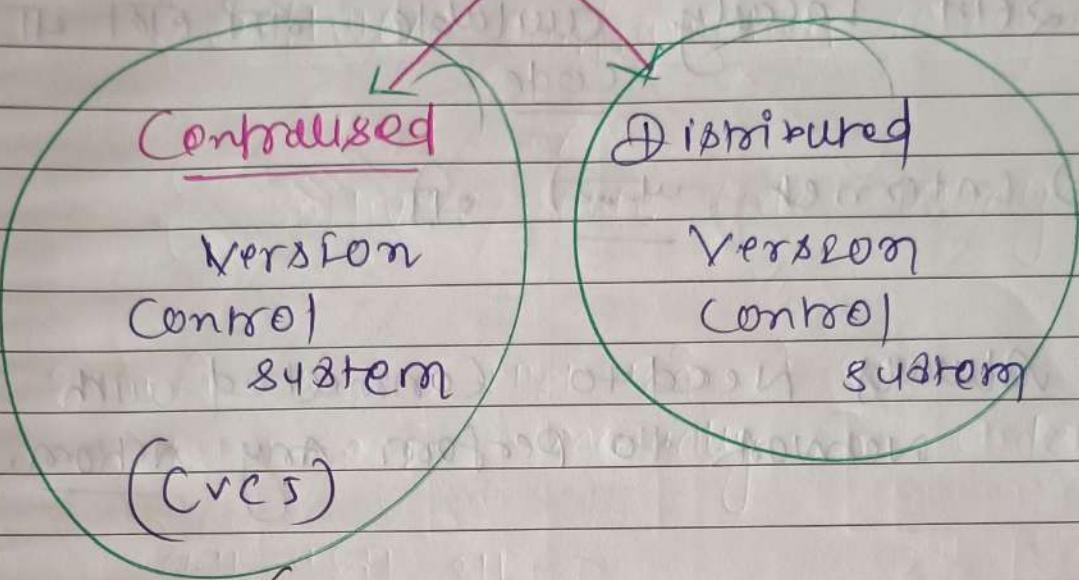
source code management

ये लारे code के version को

संग्रह कर सकता है



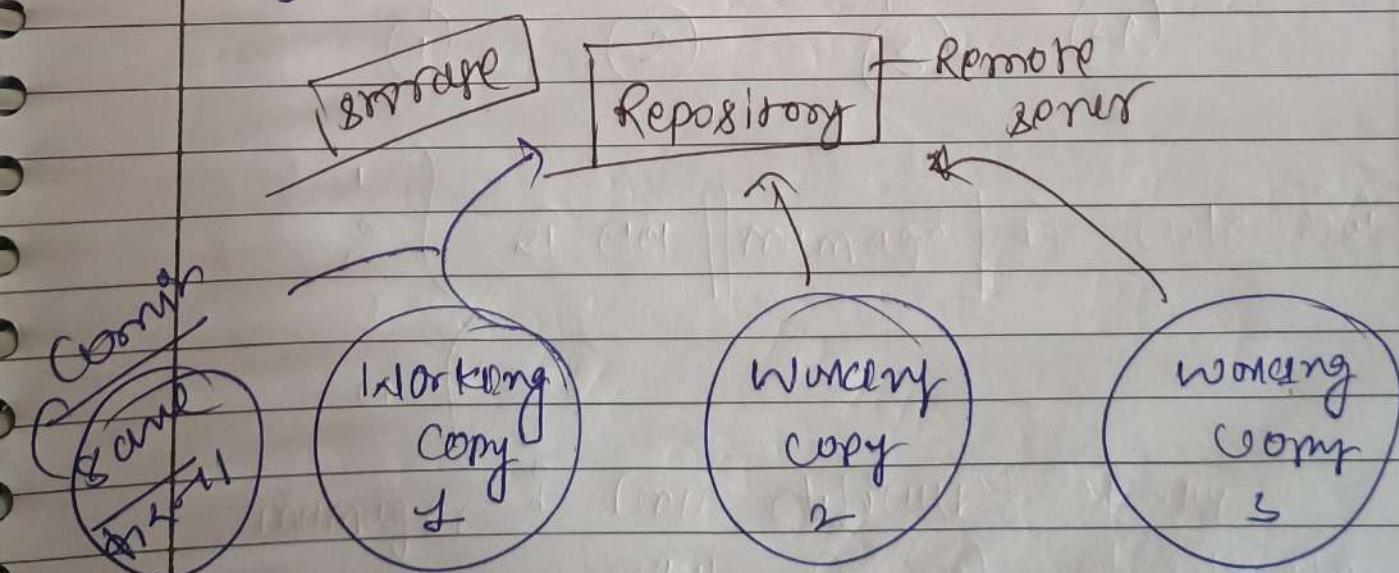
Some code management



{ Centralised Version Control System }

(CVCS)

⇒ git के आने से पहले भी CVS होता था



Drawbacks Grpc

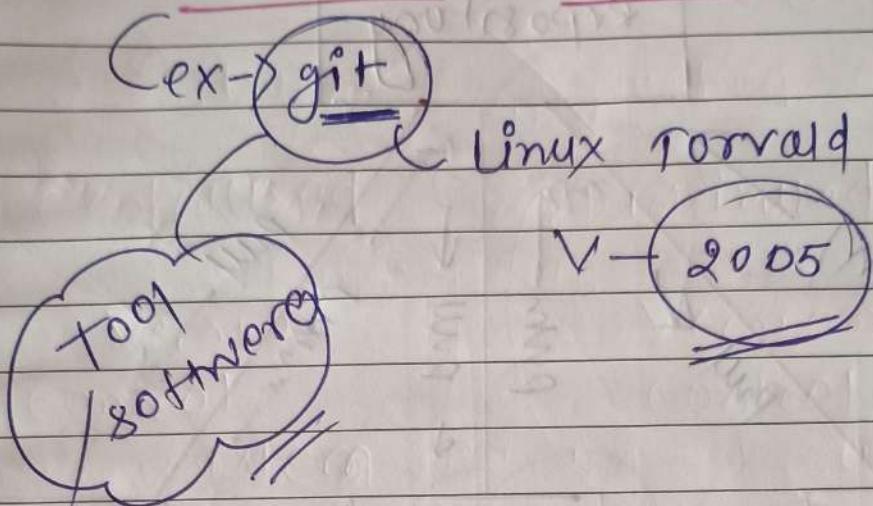
इसमें Locally available नहीं होता था
Code.

⇒ Internet जरूरी थी

⇒ Always need to connect with
network to perform any Action.

⇒ sun tool

Distributed version control system

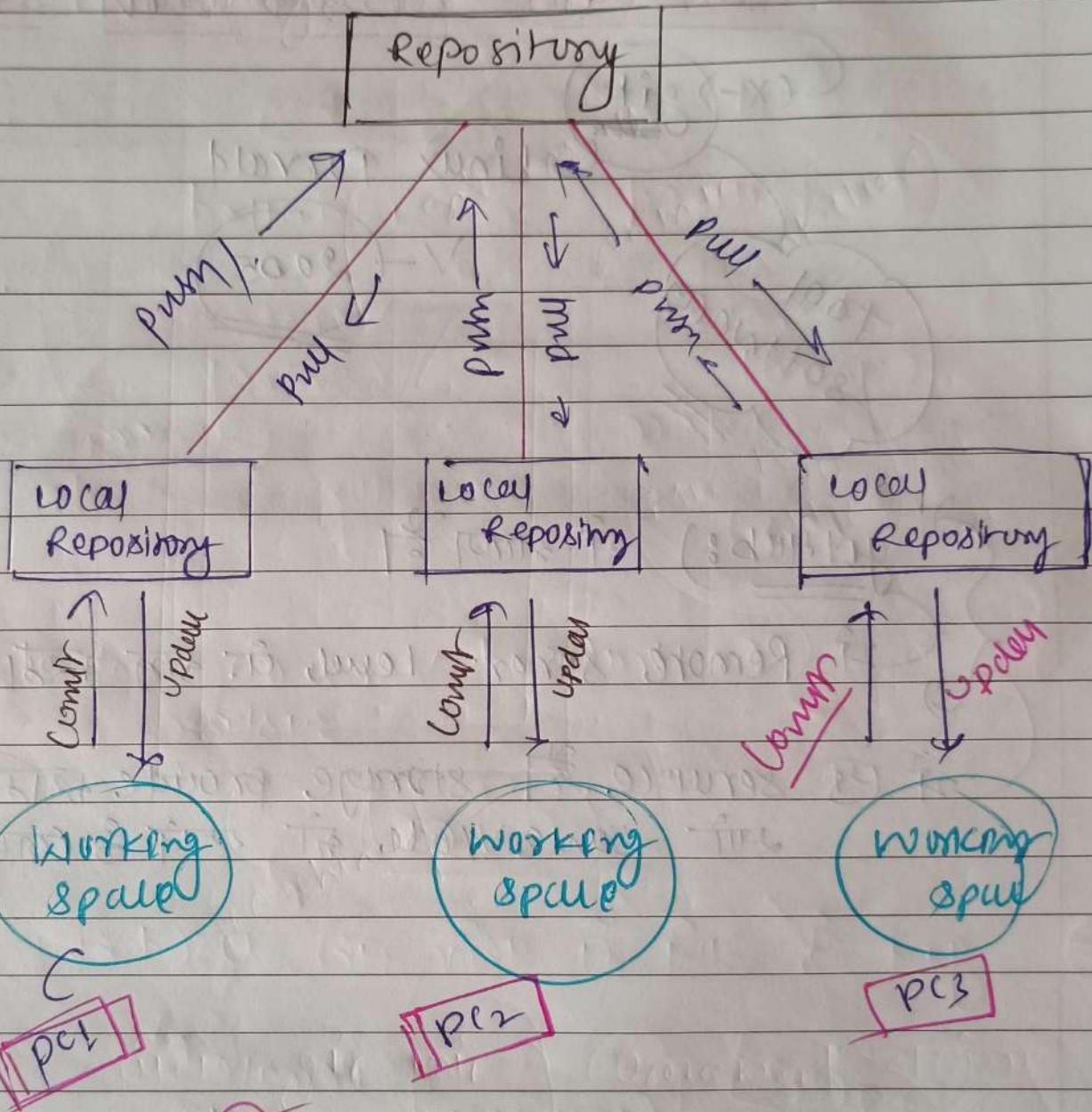


gitHub :) ऐ आज है

2) Remote server Level पर काग करता है

2) यह service है storage provide करती है
जो सारे source code को एवं वे के लिए

Remote server



22 (PC) बड़ा पैकेज
पर्सनल सेवा
गिरियांग एंड कॉम्पनी
फोल्डर और वर्कस्टेशन

git use = ②

software

DVCS

git, mercurial

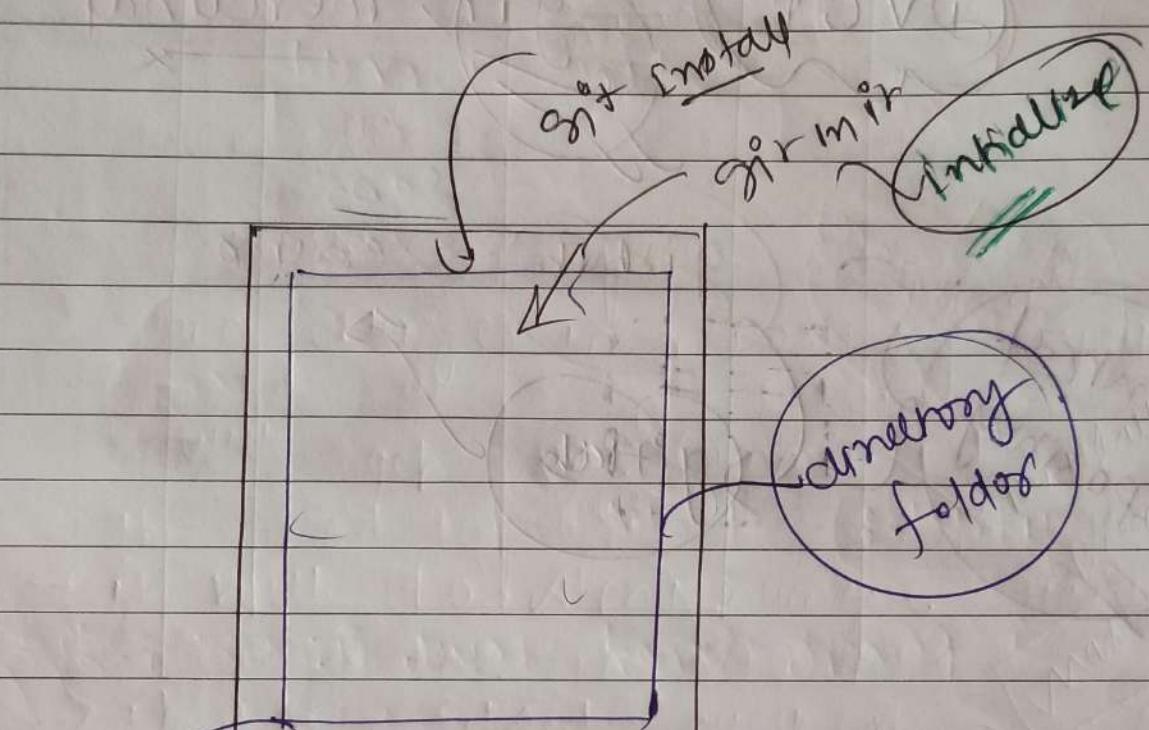
github

service

Central
repository

gitlab

Stage of git / workflow



जैसे `git init` Command दिया गया

इसी folder के अंदर `git` नाम

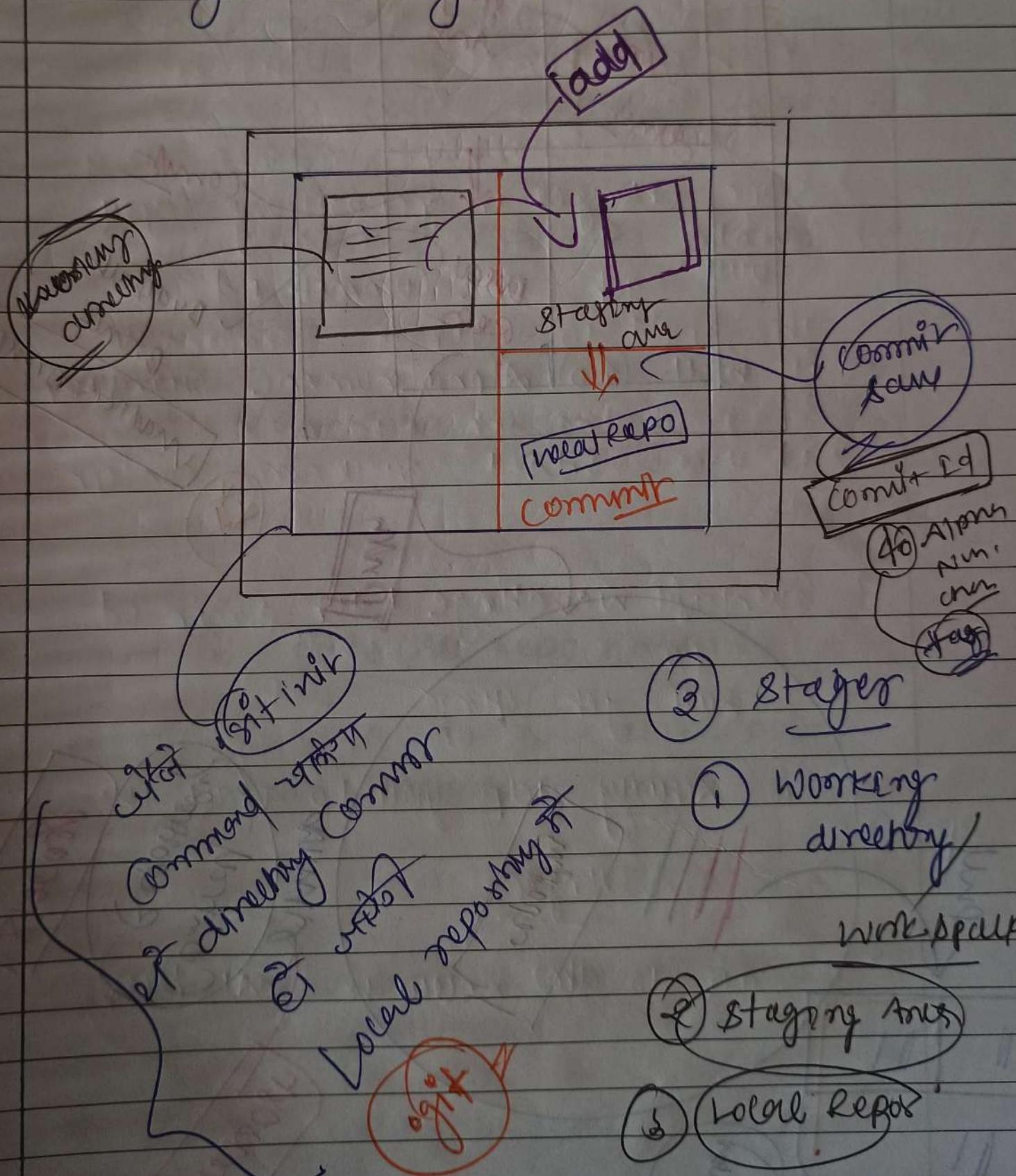
कि folder का नाम होगा

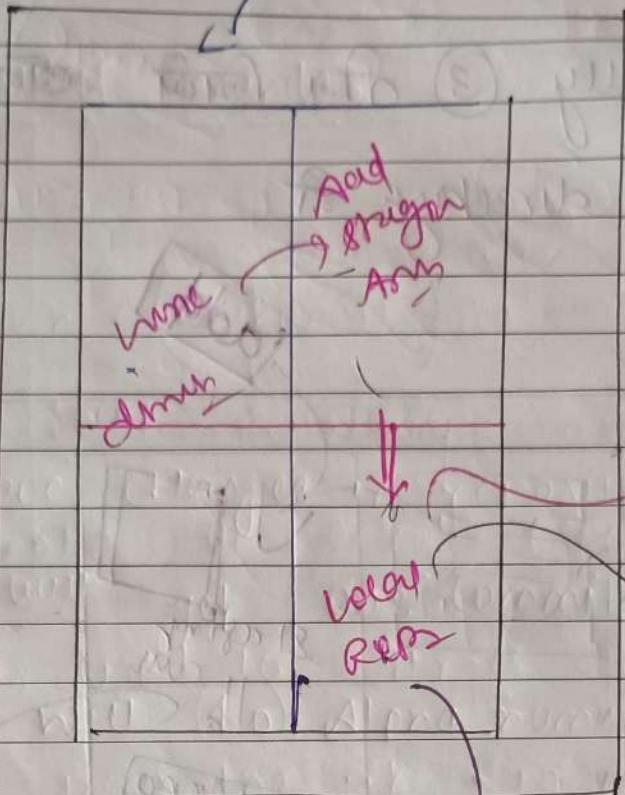
अब वह directory - convert कि

पहली git Repository कि

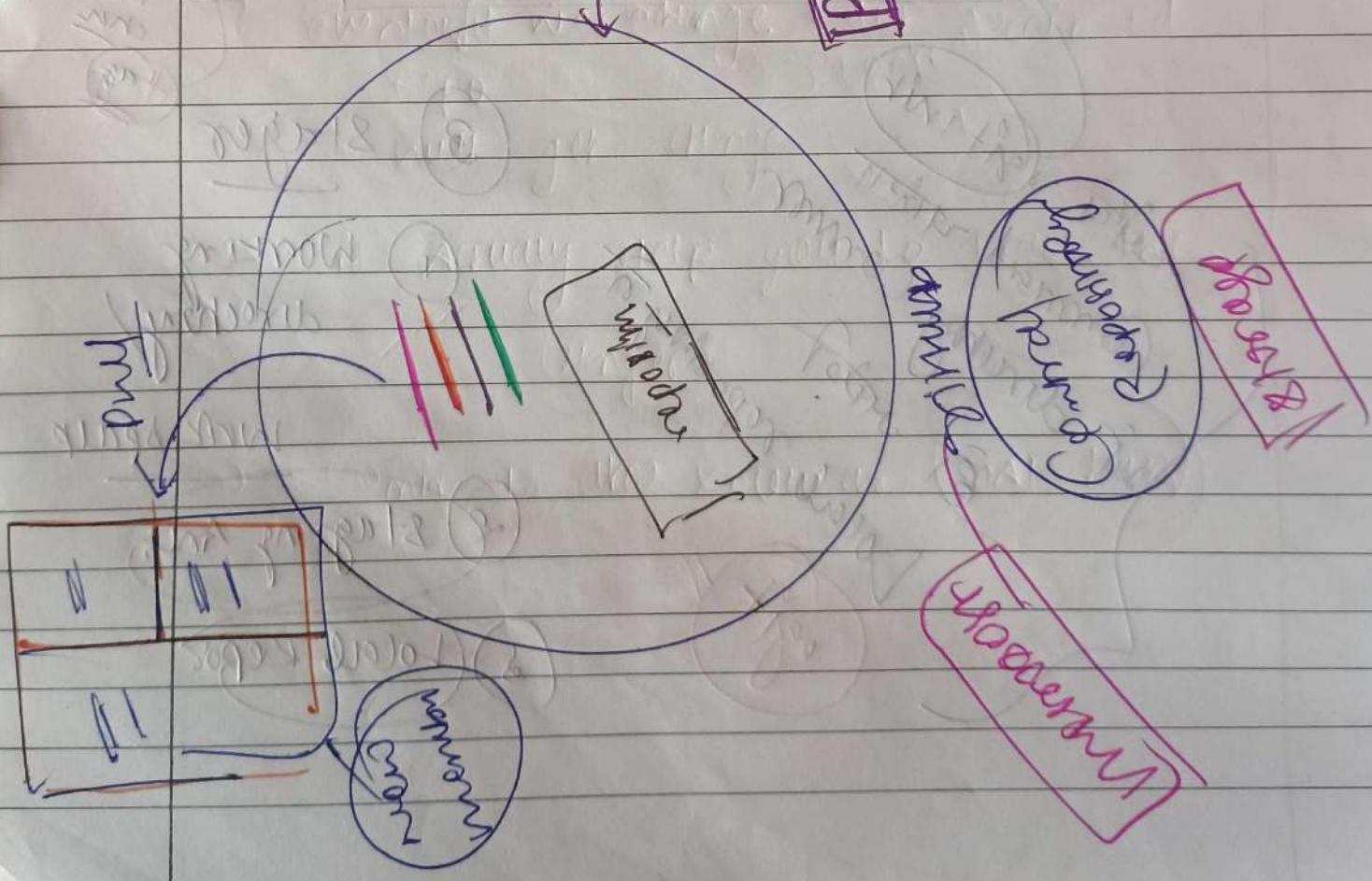
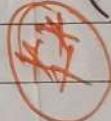
Worlaby ③ ਤੀਨ ਜਿਥੇ ਬਣ ਜਾਣੀ

git directory ??





CRM



Repository :)

- ⇒ Repository is place where you have all your codes or kind of folder on server.
- ⇒ it is kind of folder related to one product.
- ⇒ Changes are personal to that particular repository.

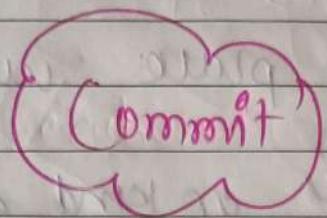
Server:)

- it stores all repository
 - (Local Repository)
- contains meta data also.

Working directory:)

- ⇒ where you see physically & do modification
- ⇒ at a time you can work on

particular Branches.



store changes in repository then

will get one Commit Id

→ it will 40 Along numeric character

→ it uses SHA-1 checksum concept

→ Even if we change 1 bit Commit Id

will get change

→ it actually help you to track the
changes.

→ Commit is also known as SHA Hash

Commit | RD | version

- ⇒ Reference to identify each changes
- ⇒ To identify who changes the file

(Tags :))

To assign a meaningfull name

with specific version in the

repository · Once tag is created
for a particular same commit

If you create new commit.

Snapshots.

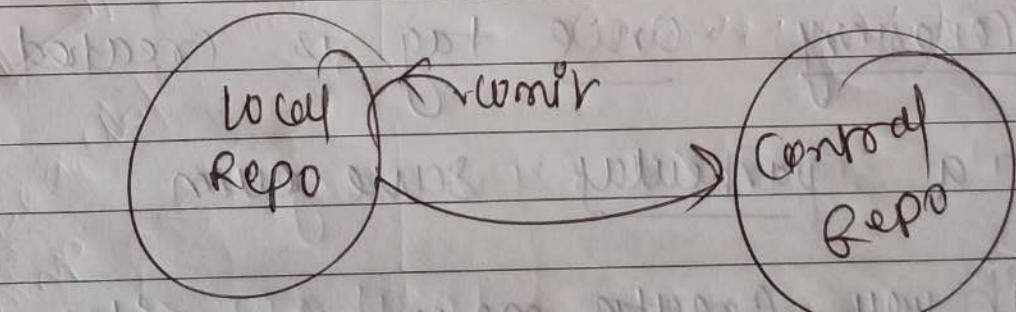
→ Represent some data of particular time
→ It is always incremental i.e it
shows the changes

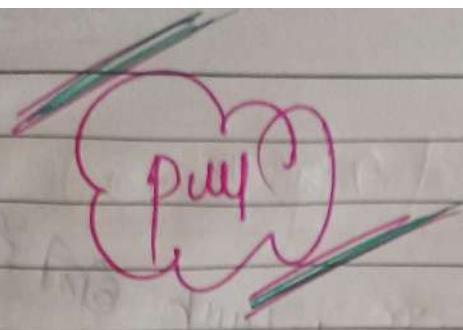
(appended daily) only not entire copy

(push) :-)

→ push operation Copies changes from a
local repository instance to a
remote or control Repo.

This used to changes permanently
into the git Repository.



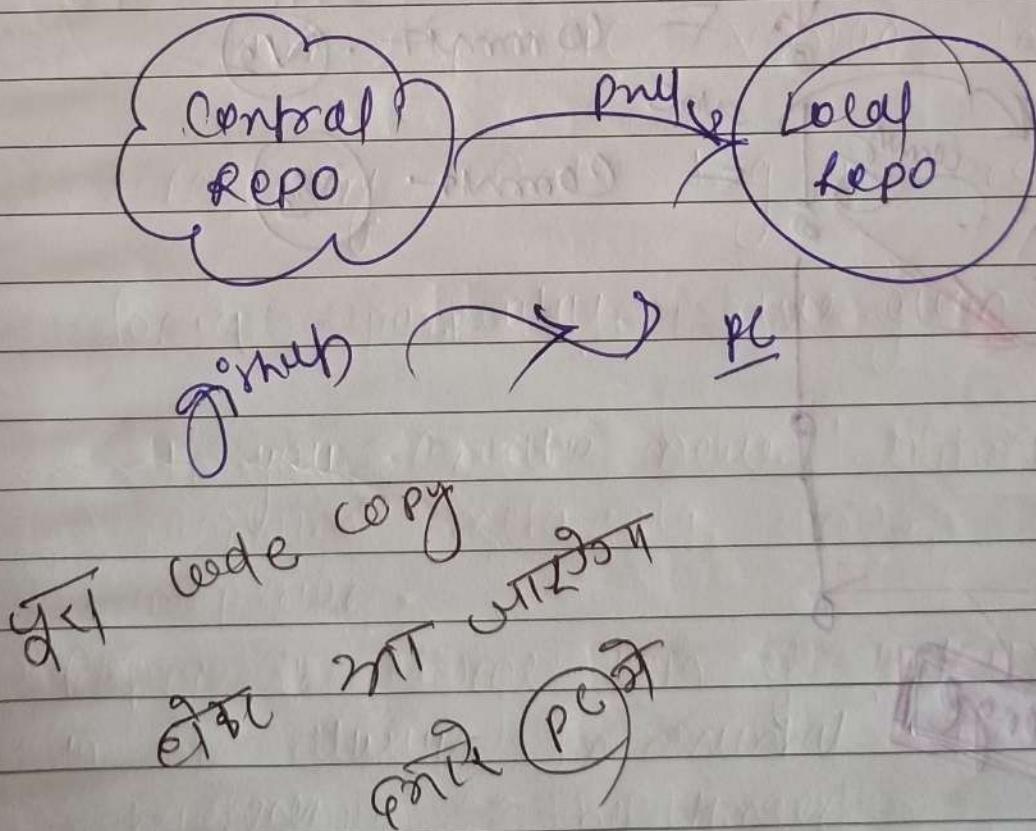


pull operations Copies the changes
from a Remote Repository

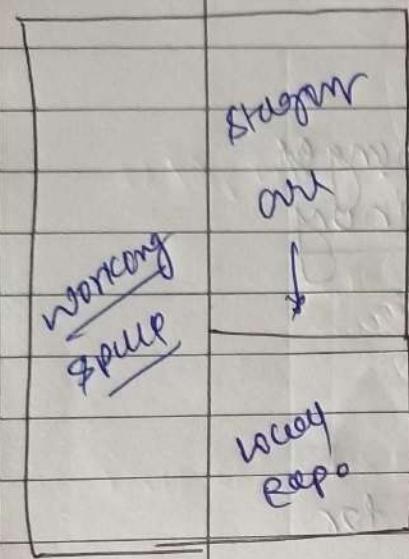
do on local machine

The pull Operations is used for

synchronization between two repo.



Branch



master

By default

index.inode

Commit ID - 1

(1)

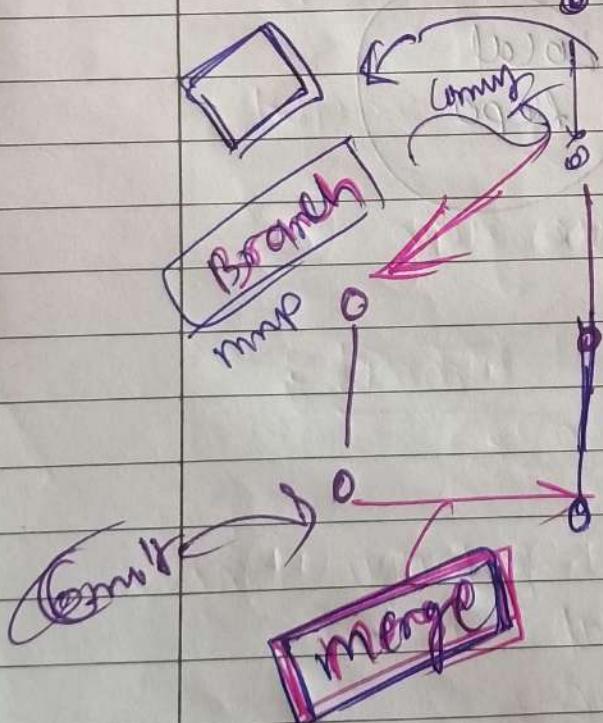
css + xyz

E Commit - (1)

and

F Commit - (1)

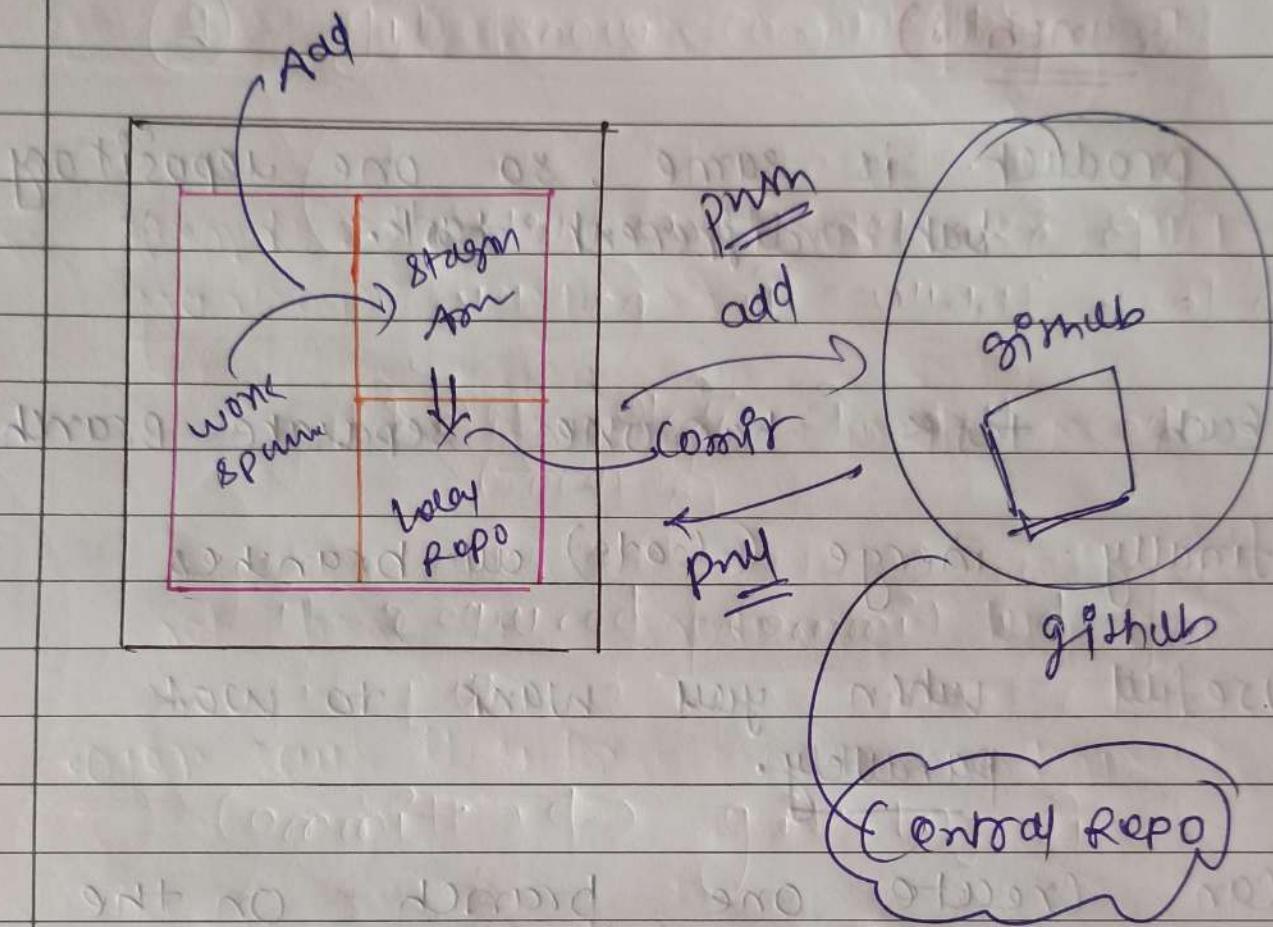
(2)



Branch :)

- ⇒ product is same, so one repository but different task.
- ⇒ Each task has one separate branch
- ⇒ finally merge (Code) all branches
- ⇒ useful when you want to work parallelly.
- ⇒ Can create one branch on the basis of another branch.
- ⇒ Changes ~~are~~ are personal to particular branch.
- ⇒ Default branch is master branch
- ⇒ file created in workspace will be visible in any of the branch workspace until you commit file you commit then that file belongs to that particular

Branch.,



steps

① create one repository / directory

② git init

git initialize करना

हमारे file local repo में convert

हो जाएंगे

git

③ create file → put some devq

④ git status → अपार इद लिस्ट होती बन फूल

⑤ git add • ← dot

⑥ git commit -m "1st commit from amu".

⑦ git status → - यहाँ दिखता है

⑧ git log → यह अपार के commit कितनी पुरानी है

⑨ git show <commitid>

इसका उपर्युक्त करते हैं
जब इन सभी जानकारी देता है

उसका अनदेखी ने यहाँ code
लिया है

it required Commit Id

Commit का → git log में
निहित है.

⑩ git remote add origin

Central git url

⑪ git push -u origin master

User

(enter username & password)

git push
username

* git status से untracked
file पता चाहा है
किन तरीके file हैं

* git log - से भी पता चाहा है
क्या commit किए हैं

create one directory and
go inside it

- git init
- git remote add origin <github rep or >
- git pull -u origin master
- git log
- git show <commit id>

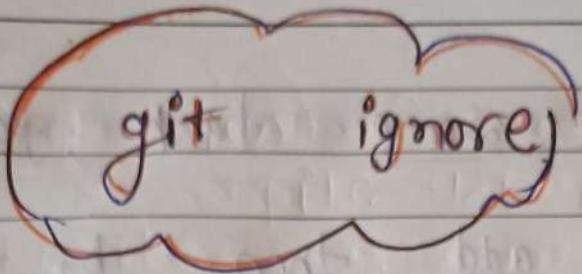
Now add new code in file

- git status
- git add .
- git commit -m "Sync done"

git stash

git merge

git pull origin master,



> To ignore some file while committing.

> Create one hidden file (.gitignore)

and enter file format which you
want to ignore

for eg -

| yi .gitignore

> git add .gitignore

> git commit -m "Latest update excluded .gitignore"

> git status.

> Create some text, jpg, & cs.file

and add them by running

"git add"

for

eg:

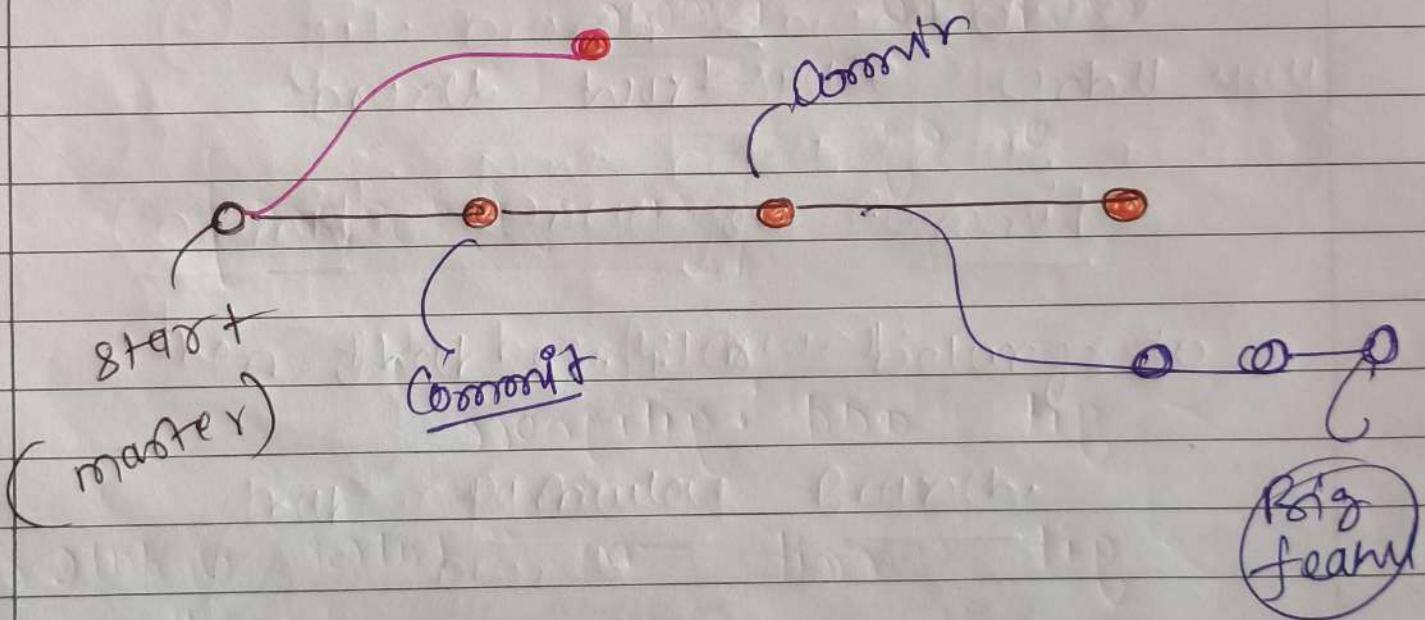
touch file1.txt
file2.txt
file3.jpg
file4.css

ls
git status
git add .
git status
git commit -m "my text file
only".

How to create branch and merge.

जब git initialize किया जो हमारे पास default branch होती है

उसे **master branch** कहते हैं।



The diagram above visualizes a Repository with two isolated lines of development one for a little features and one for a large - running features By developing them in Branch it's not possible to work

on both of them in parallel but it also keeps the main master

Branch free from error.

- Each task has one separate Branch
- After done with code merge other branches with master.
- This concept is useful for parallel development.
- You can create any no. of branches.

⇒ changes are personal to that
particular Branch.

⇒ default Branch is master.

⇒ file created in workspace will
be visible in any of the
branch workspace until you

Commit Once you commit

then that file belongs to

that particular Branch.

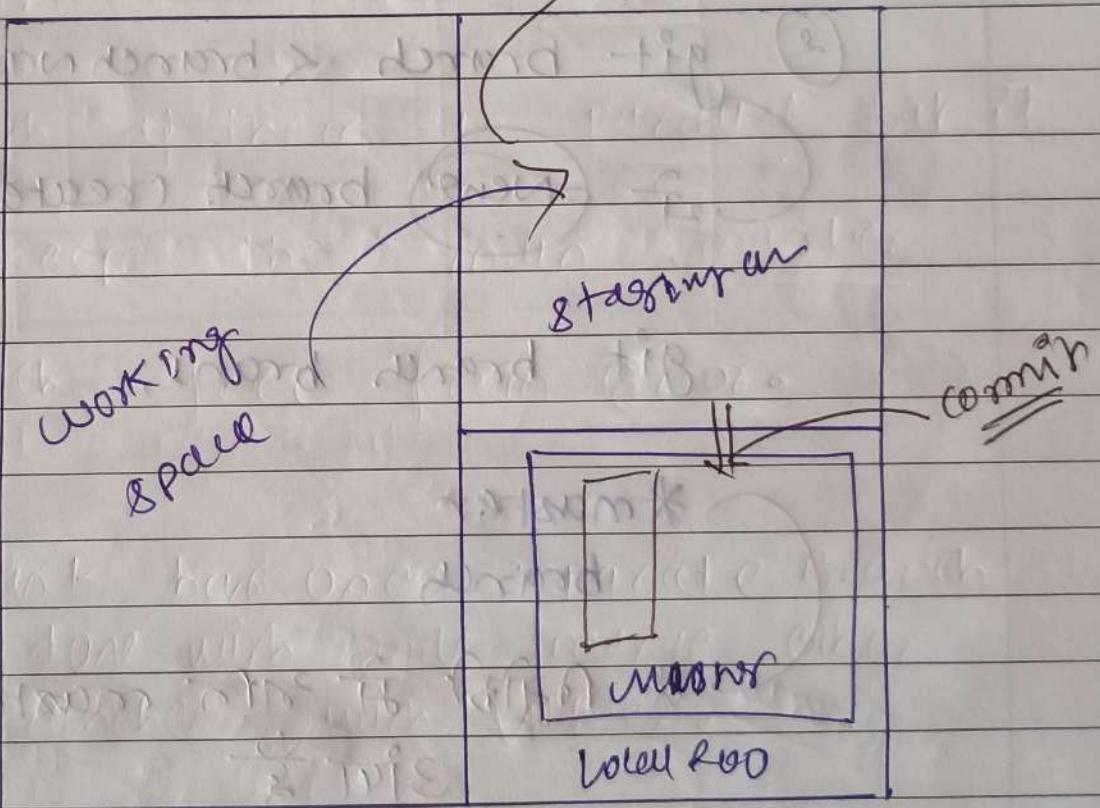
⇒ when created new Branch (data)

of existing Branch is copied

of new Branch.

Branch बनाने से ८३ लाख (multiple)
कम कर सकते हैं।

We can create (unlimited) branch



⇒ Branch से parallel development करते हैं।

Branch में जो अस होगा वो Branch

मिट जाएगा.

① git branch

जो जी जो Branch हो वो सब
Branch के दैना,
*master

② git branch <branch name>

* new branch create कर दूँगा

git branch brnt;

* master

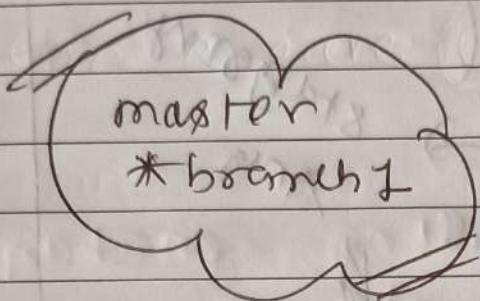
branch

* नयी में आगे master के

अंदर है

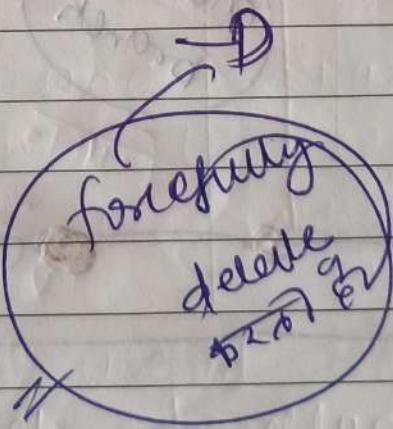
? How to move one branch to another branch

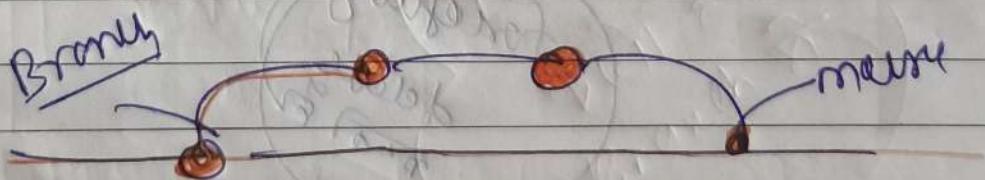
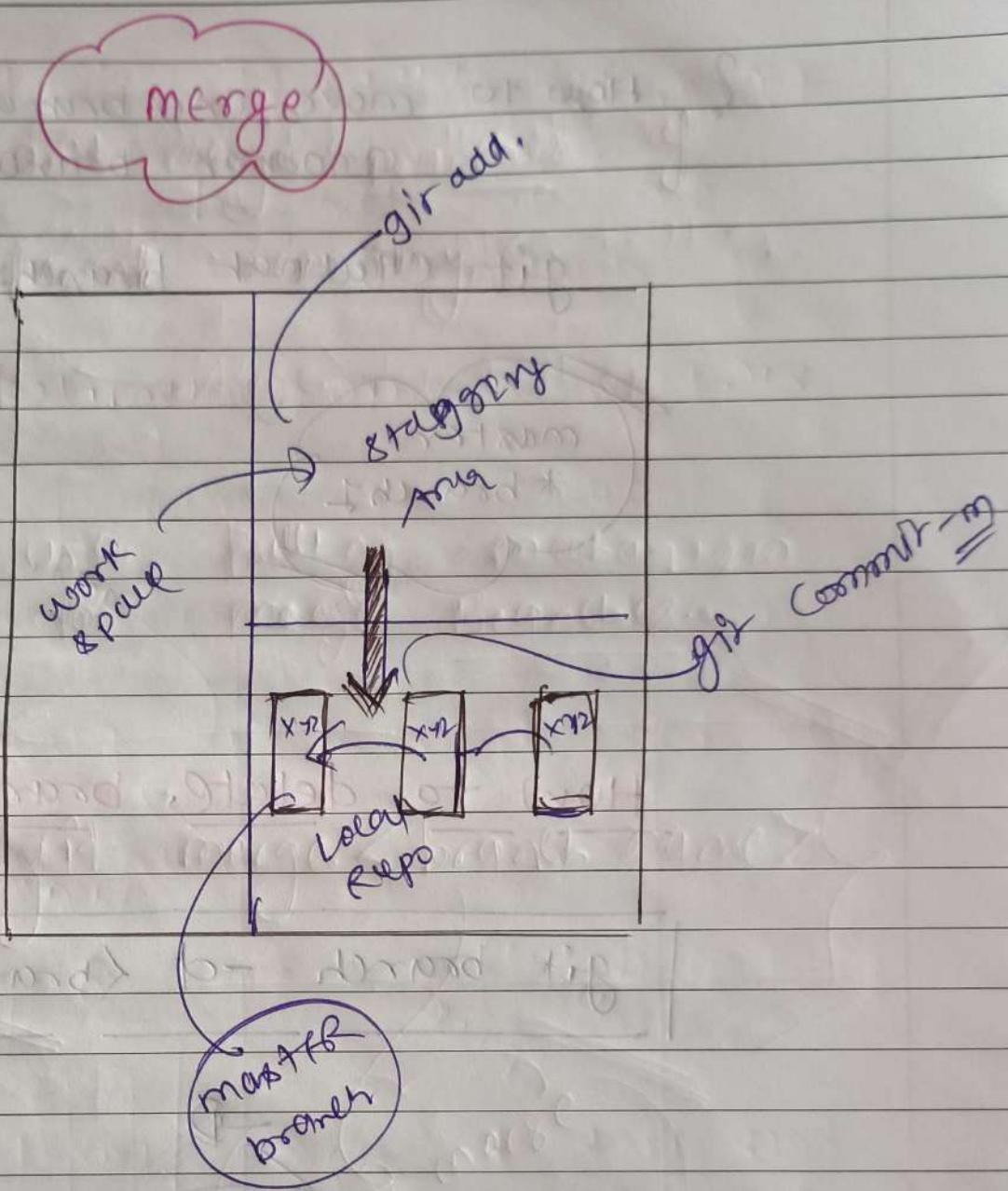
git checkout branchY



How to delete branch

git branch -d {branchName}





merge करने की ही pulling mechanism
करने ही
Z

→ यिल नि Branch मे merge करनी हो
New data pasthe हो जाएगा.

(ये data updated है।)

github पर
उत्तर दें
2015 में

Logical connection

इसका logo क्या होता है
ये Twitter & Facebook

होता है

Microsoft के Buy 2011
7.5 billion dollars के

purchase

Open source है।

जिस ने Branch में merge करनी थी

New data pastre को मर्टेगा.

(ये data updated है)

github यार

वाला एम

एम

Logical Connection

हल्का logo की बोल्ड

पर्दे Twitter & सिर्फ

बोल्ड

memosoft & Buy pass

7.5 billion dollars

Open source

Original

→ we can't merge branches of
different Repository

→ we use pulling mechanism
do merge Branches.

git merge <branch name>

PULLS merge

git
Branch

git Conflict

when same file having different content
in different branches if

you do merge, Conflict occurs

(Resolve Conflict then add and Commit)

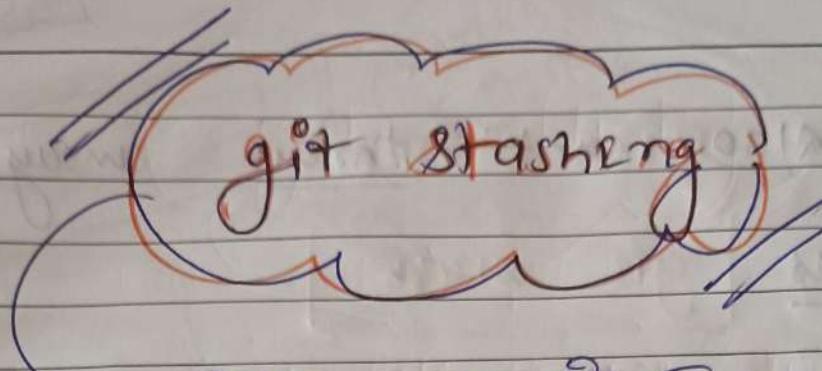
* "Conflict occurs when you
merge Branches"

Same नाम को दो file in different
branches से लिया है तो git
Confused हो पाता है. ऐसे Conflict
आएंगे.

जबकि conflict आ जाती

matter को resolve करना है

जो-जो merge की जरूरत है



अब तक से Temporary
Storage का

Local repository के अंदर ही रख
Storage होता है

Suppose you are implementing a new feature
for your product. Your code is in
progress. And suddenly a customer

escalation comes

Because of this you have to keep
aside your new feature work for
few hours

You can't Commit your partial code

And also can't throw away yours

Changes

so you need some temporary
storage.

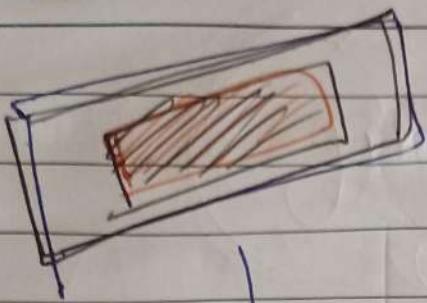
when you can store yours

partial changes and latter on

Commit it

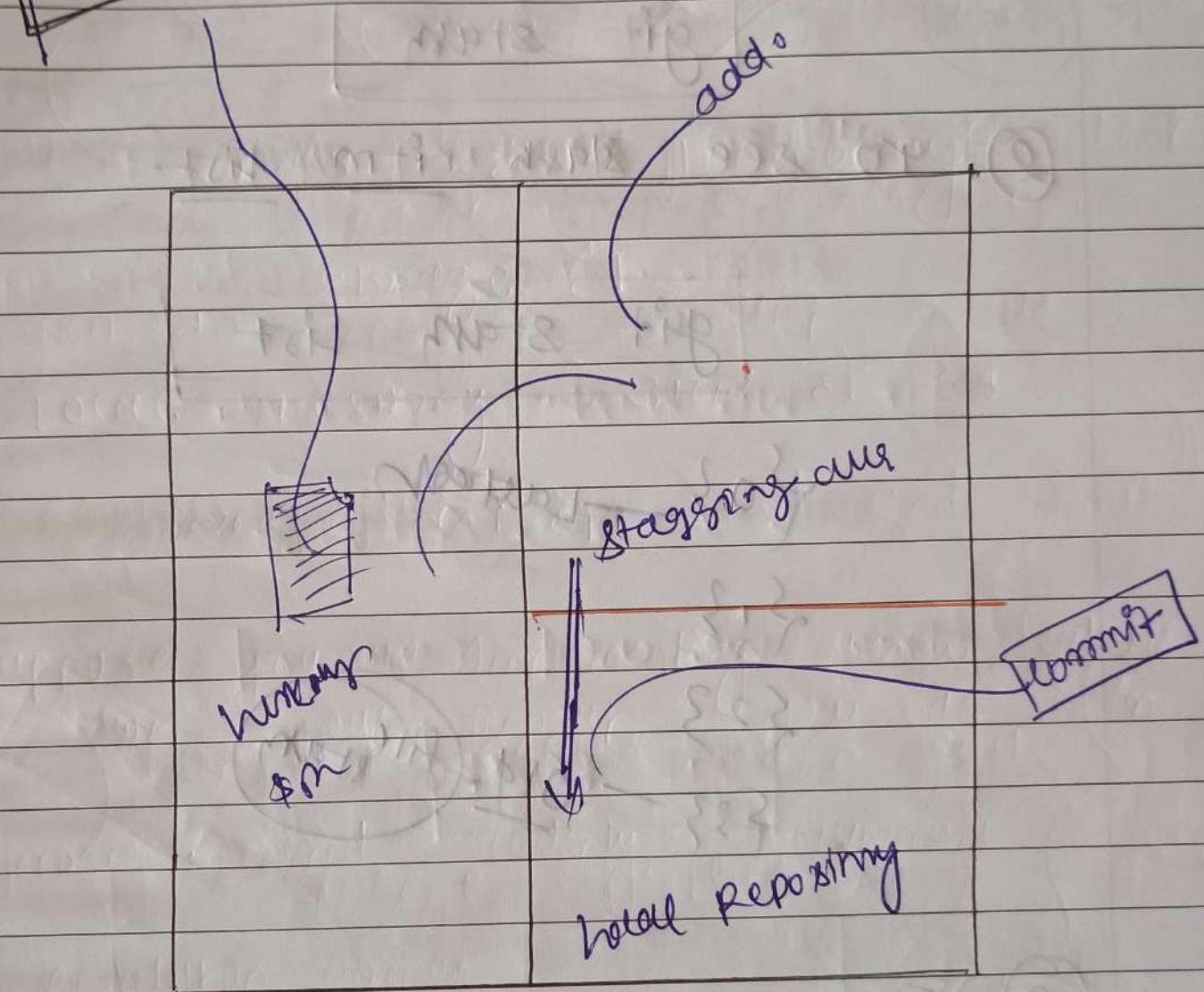
To stamp an item (only applies to

modified files



Bash
store

Date _____
Page _____ NRP



① go stash an item

git stash

② go see stash items list

git stash list

{03} - ~~laptop~~

{13}

{23}

{33} - org - ~~file~~ ram

③ To apply stashed items

git stash apply stash@{03}

Then you can add & commit

To clear the stash items

git stash clear

यदि हम कागज स्टाम्प ले working space

में लाते हैं तो कहे कागज **stamp** में
कि रहता है.

Copy & paste होता है।

git Reset

→ मात्र एने working space से

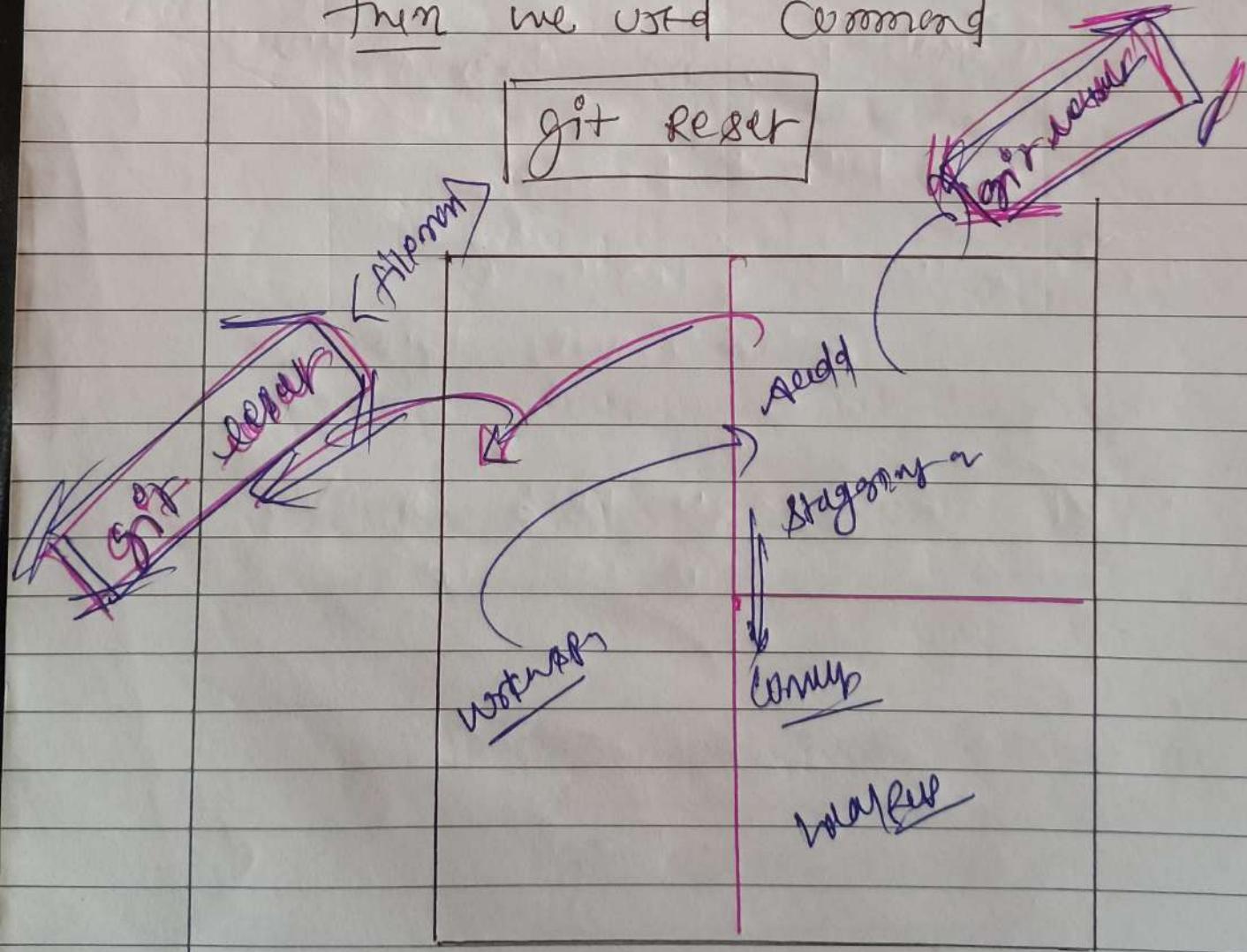
Staging area की code अपे दूर हो

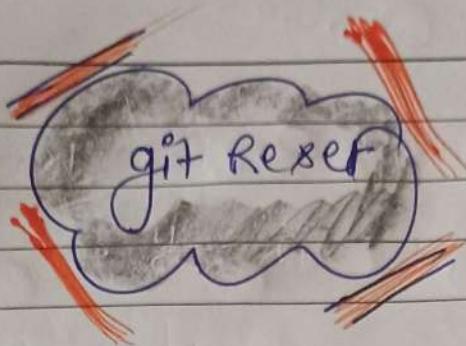
fir we want to remove

Code → from Staging area

then we used command

git reset





git reset is a powerful command
that is used to undo local
changes to the state of git repo.

To Reset staging area:

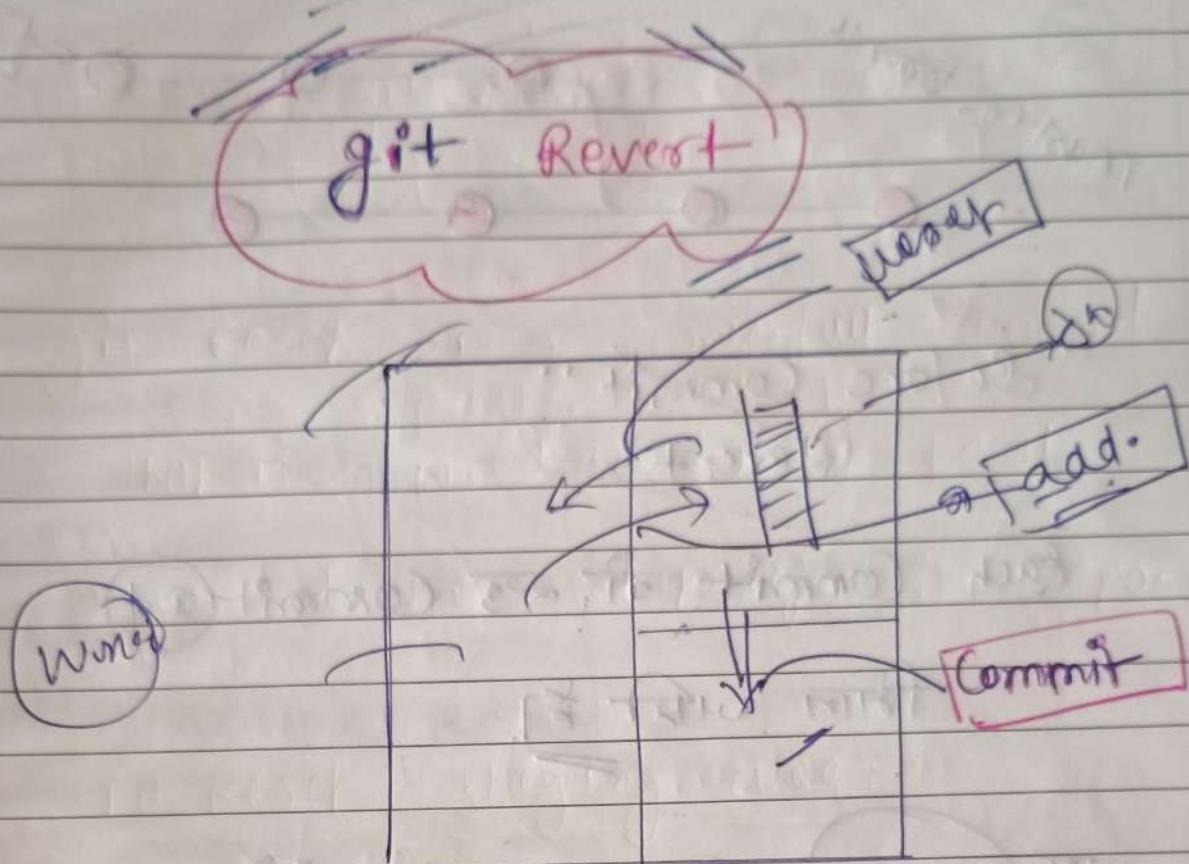
git reset <filename>

git reset .

To reset the changes from both
staging area & working directory

git reset

git reset --hard



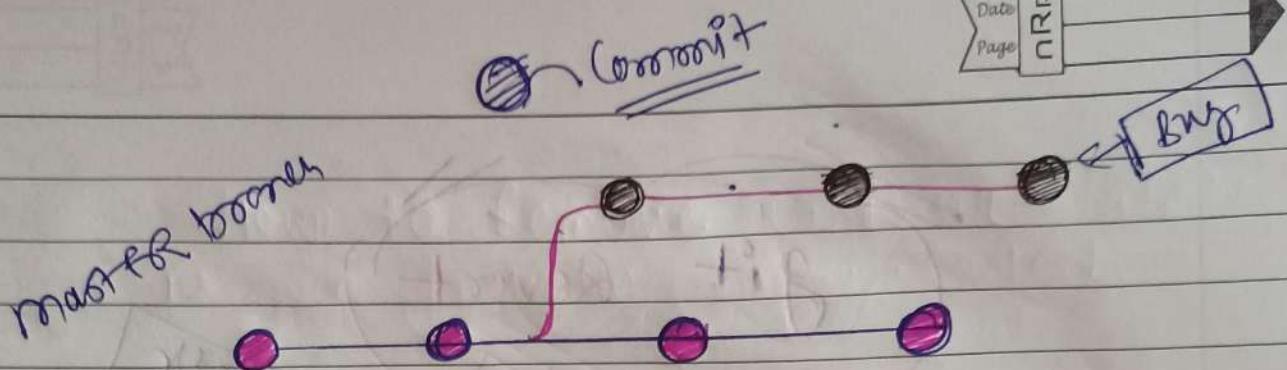
Commit जैसे हो जाए वो वापस
आ जा सकता है नहीं कर सकता

{
Reset → Before Commit }
{
Revert → After Commit }

⇒ The revert command help you undo

an existing commit

Undo → पिछे आना पाहता है



" Before Commit "

Revert

Each Commit को एक Commit है

मिल जाती है

जब Revert Command चलायें तो

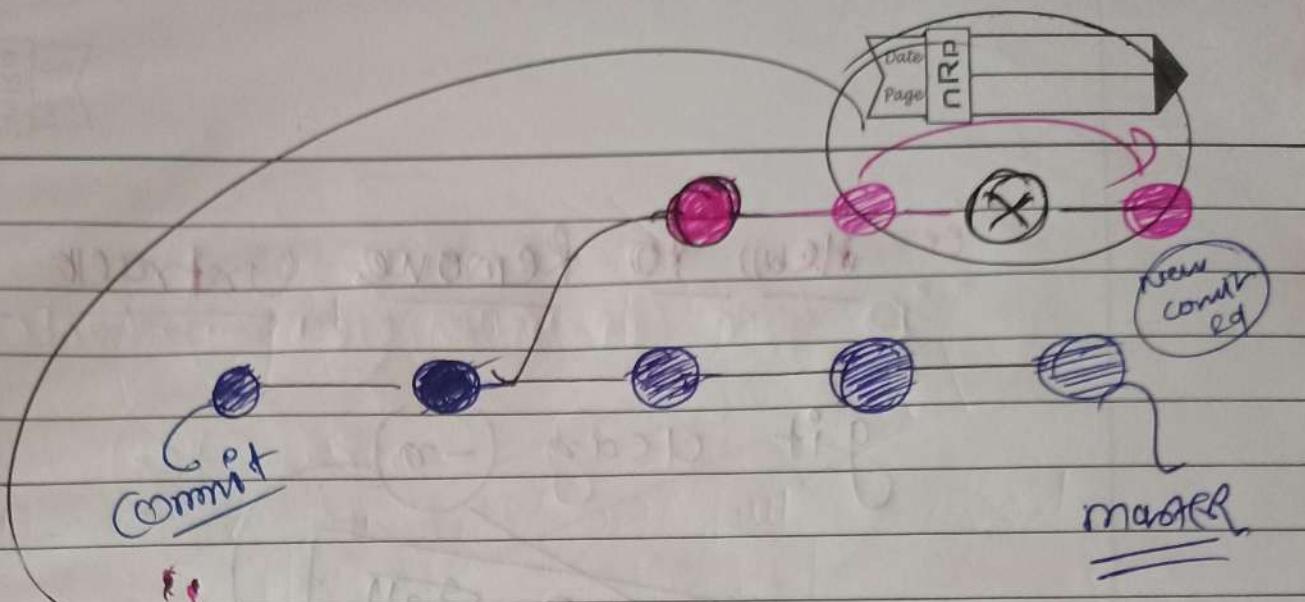
Commit है के साथ

में चाहता है वे जो को किया गया

हैं Revert के बाद

Means इसलिए पहले भी को किया गया

उन्होंने नहीं किया



Remove changes from the
Revert commit"

"After Reverting"

अब git Commit करोगे तो

git revert अपने आप उत्तम }
Commit लगा कर देगा.
} Commit लगा कर देगा.

git revert <Commit-id>

"How to Remove Untrack file"

git clean (-n)

✓ अपेक्षित

नहीं वार्निंग
दरात्रि

git clean (-f)

Forcefully

परिवर्तित
करने का लिए

git commit -m "remove file"

Tag

Commit ID
40

Alphanumeric

Commit ID

Commit ID - - - (Important)

→ Tag operation allows giving meaningful
name to a specific version in the
repository

git tag -a <tag name> -m <message>
 <Commit ID>

To see list of tag

→ git tag

To see particular Commit Content
By using tag

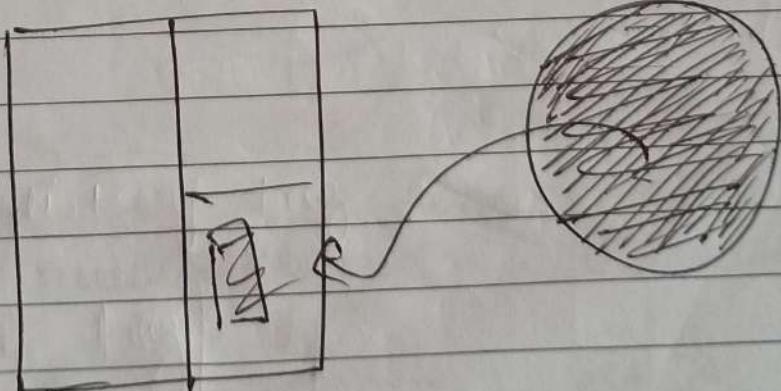
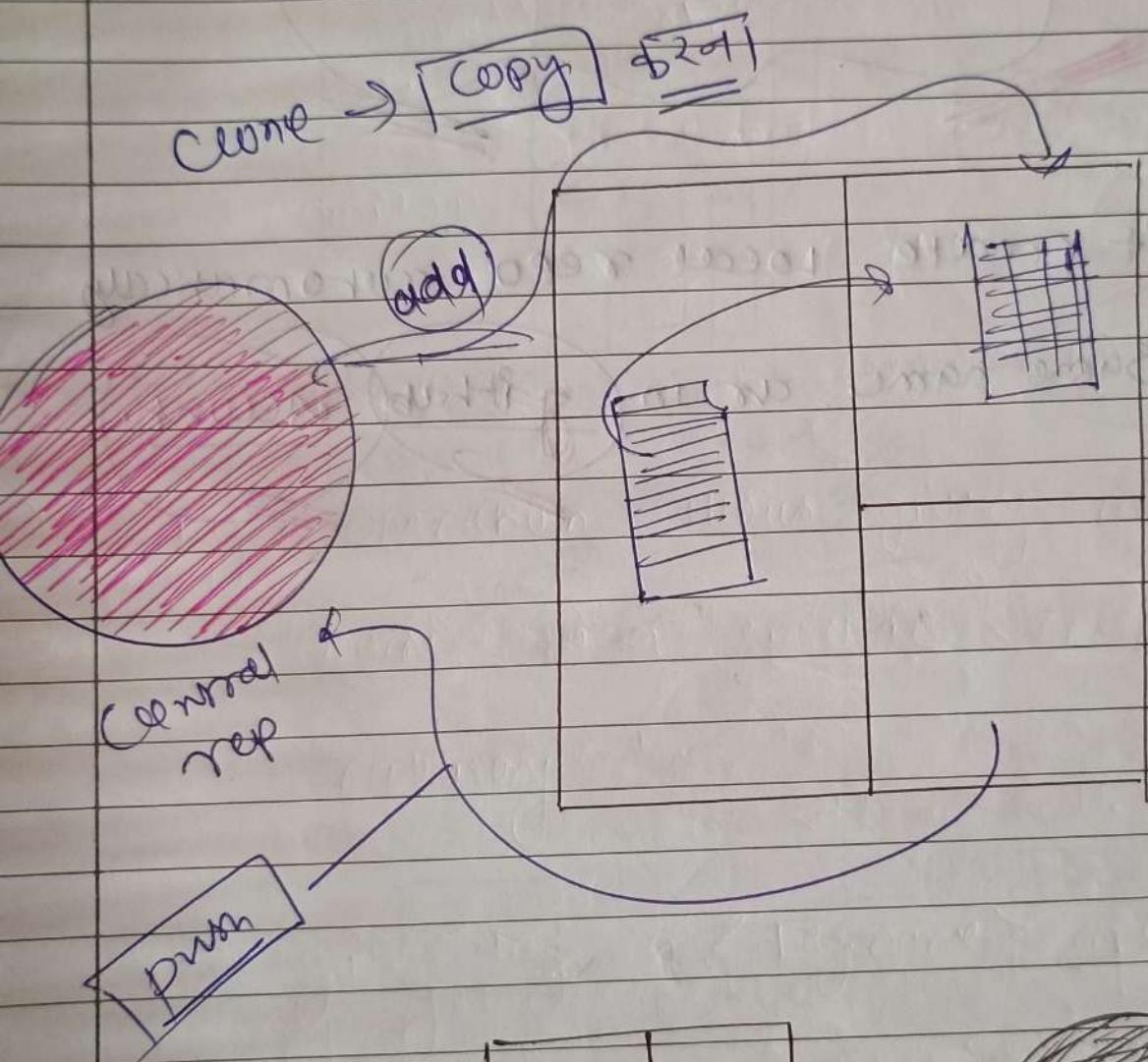
git show <tag name>

or Committed

To delete tag

git tag -d <tagname>

github cloning



git clone < url >

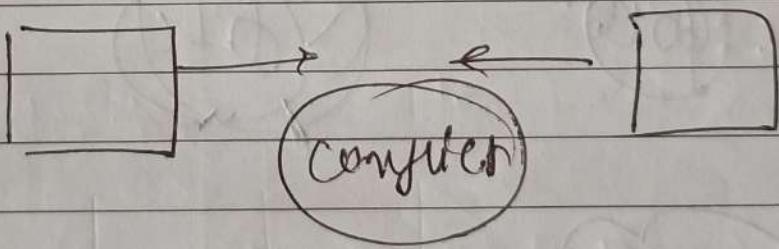
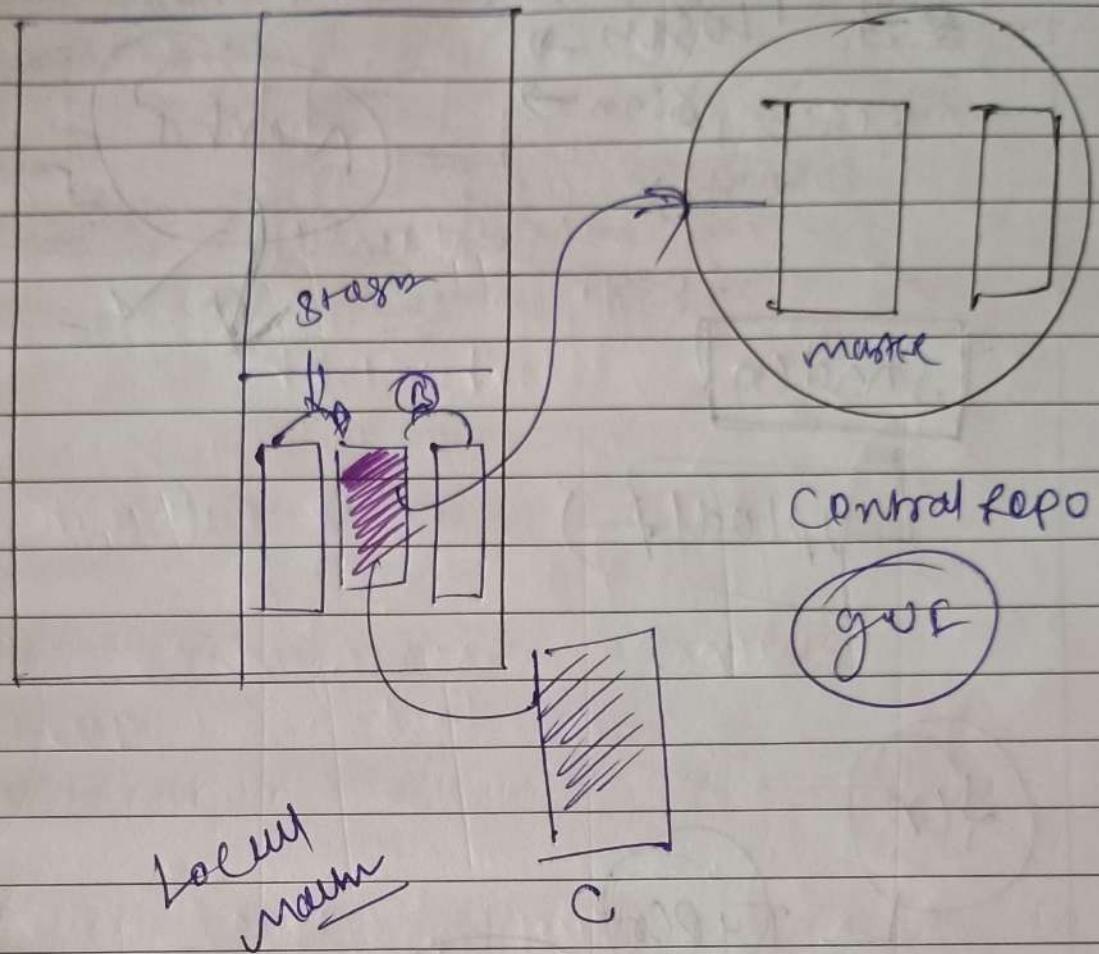
↳ local repo

it create local repo automatically

Same name as in

github

account.



✓

pull request

merge इसे ले पाए
परन्तु मार्गदर्शक