

GIT

24 July 2025 12:26

0. To init

`git init`

1. to Check

`git status`

2. Add GIT Email

`git config --global user.email "abc@gmail.com"`

3. Add GIT Name

`git config --global user.name "XYZ"`

4. Add file in Git

`git add .` (To add all file)

5. Add Commit the to the added file

`git commit -m "initial commit{Any message}"`

6. To See the Branch

`git branch`

7. What is Head?

Head is the Latest Branch

8. WHAT is log (kind a History)

`git log`

`git log --oneline` (easy to seee)

9. Git do not trach empty folder

10. How to trach empty folder?

just add `.gitkeep` file in the folder

11. To see the config list

`git config --global --list`

12 . To know where to push the repo

`git remote add origin "https://github.com/MohitJo27/GitTutorial.git"`

13. To see the github is there or not

`git remote -v`

14. To change the branch name

`git branch -m main`

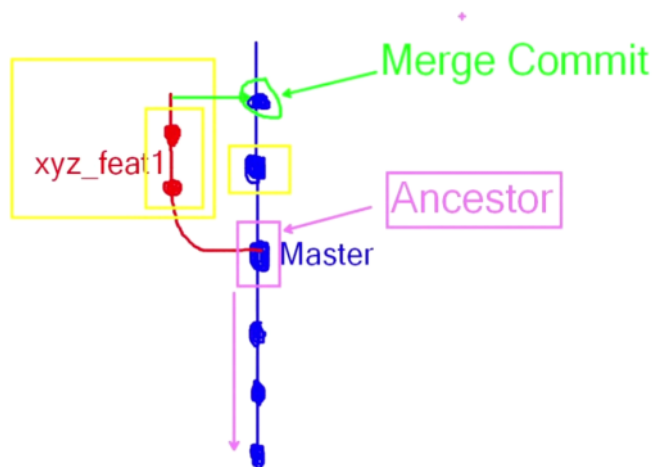
19. TO RESTORE

`Git restore`

Branches

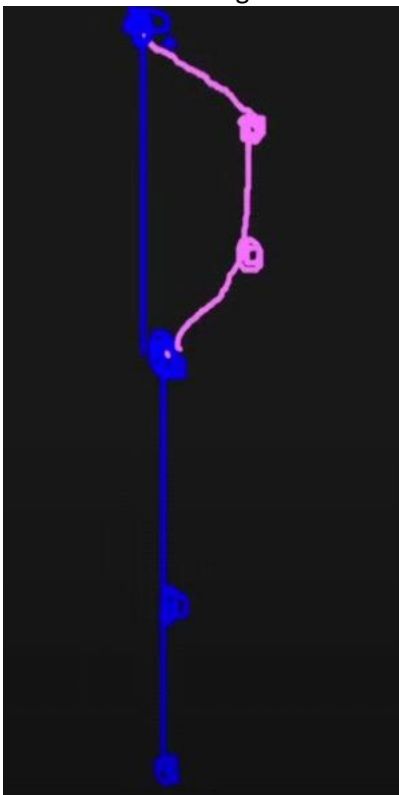
24 July 2025 12:59

=====

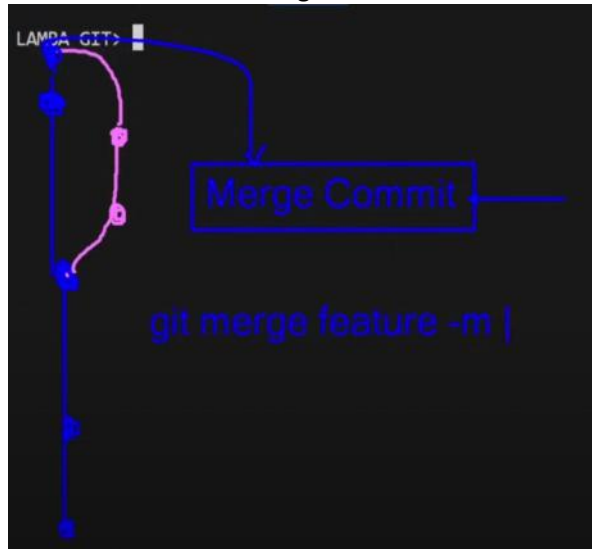


1. To Switch the Branch
git switch master or git checkout master (OR to check if we are into master branch)
2. To create the new Branch or switch if its presented already
git switch -c feature_mohit_1 or
git checkout -b freature_mohit_1
3. To Merge the commits
> git merge branch name

-- Fast-forward merge



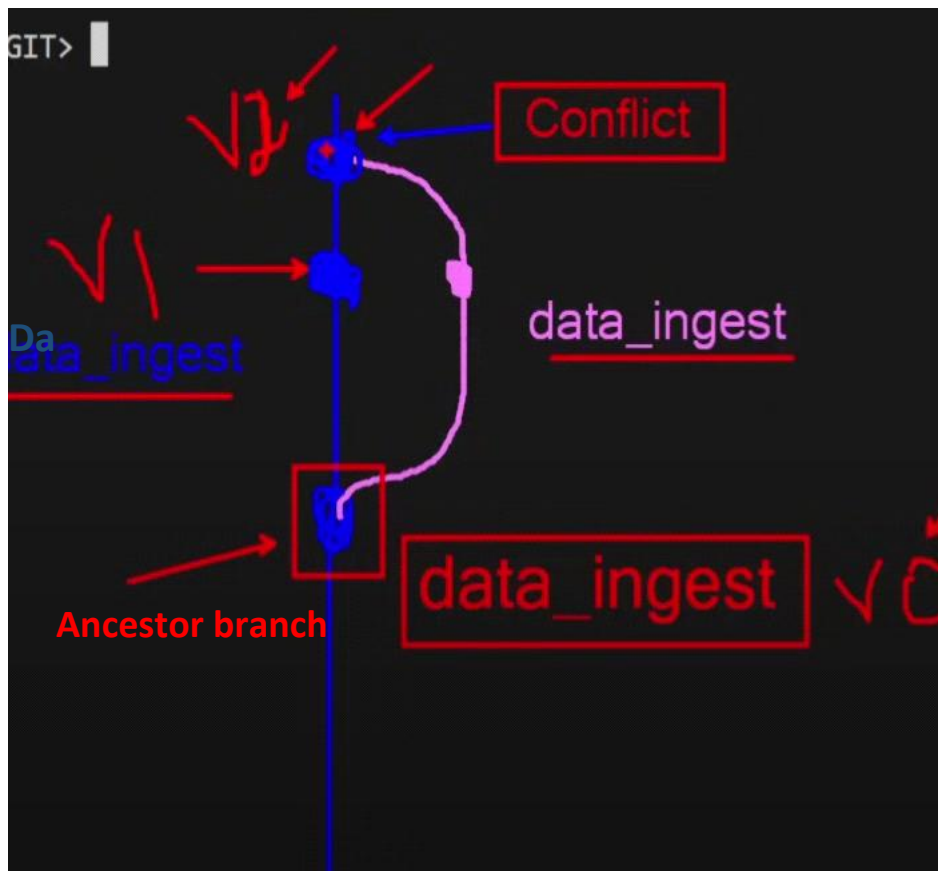
-- Non Fast-forward merge



+++++ Merge Conflicts +++++

4. When merge conflicts occurs?

When 2 dev work in the same file but they don't know they are working on the same bug. They commit both the same thing.

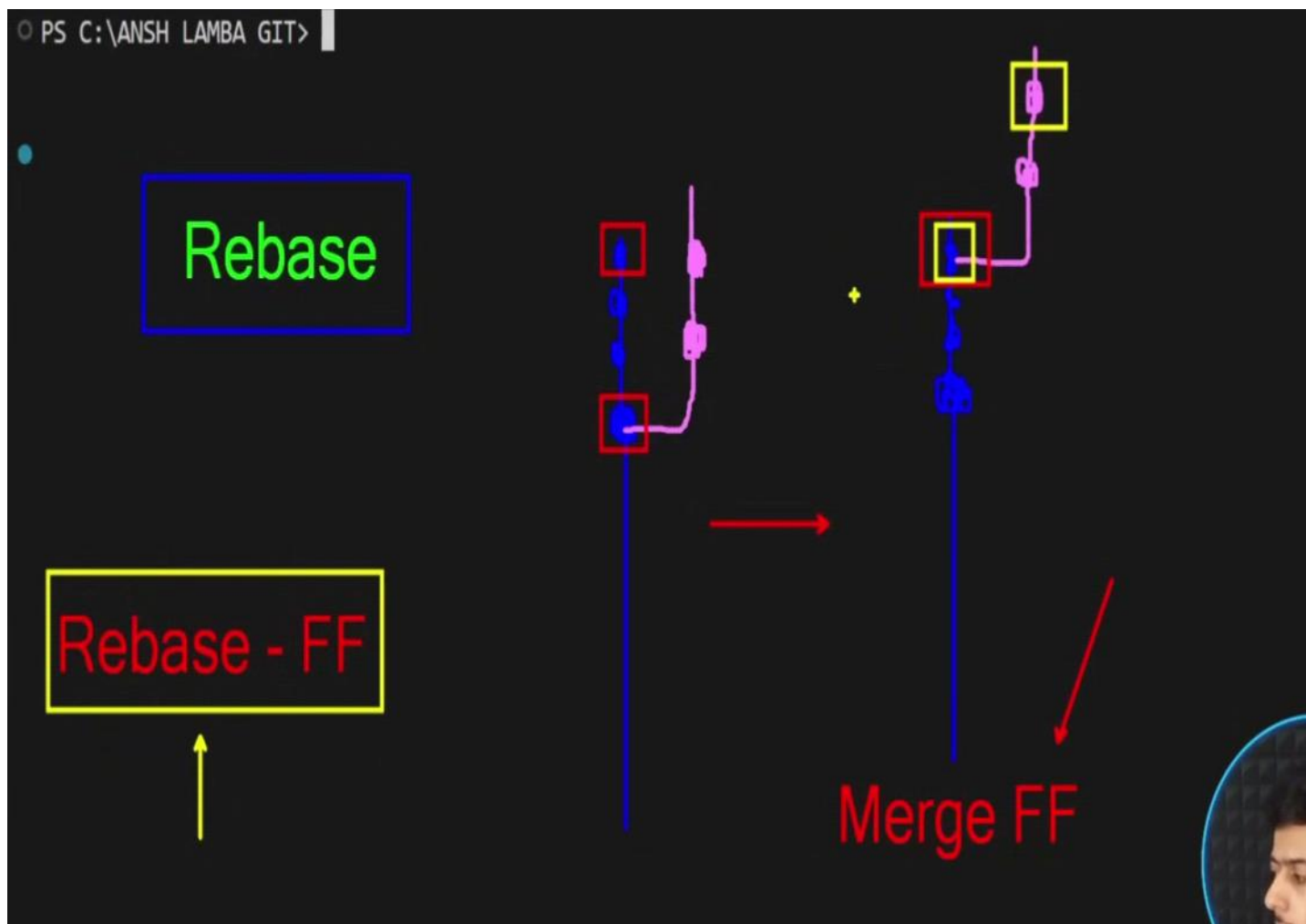


5. How to Abort
git merge --abort

6. After resolving the conflict do
Git add .
Git commit -m "--"

Rebase

25 July 2025 18:15



First switch to the branch which you want to rebase

And then in terminal

---git rebase master

```
PS E:\GIT\ GITHUB> git switch rebase_feat
Switched to branch 'rebase_feat'
PS E:\GIT\ GITHUB> git rebase master
Successfully rebased and updated refs/heads/rebase_feat.
PS E:\GIT\ GITHUB> git switch master
Switched to branch 'master'
PS E:\GIT\ GITHUB> git merge rebase_feat
Updating f1429fb..52af4db
Fast-forward
 data_ingestion.txt => data_ingestion.txt | 5 +++-
 1 file changed, 4 insertions(+), 1 deletion(-)
 rename data_ingestion.txt => data_ingestion.txt (83%)
PS E:\GIT\ GITHUB>
```

It will be fast-forward merge or new name is 3 Way merge

Reflow & Time Traveling

26 July 2025 11:31



To see the head number

1. git relog --- is a advance version of the git log --oneline

Output of the relog

```
PS E:\GIT\GITHUB> git relog
f0d1a12 (HEAD -> master) HEAD@{0}: commit: delete data_ingest.txt
52af4db (rebase_feat) HEAD@{1}: merge rebase_feat: Fast-forward
f1429fb HEAD@{2}: checkout: moving from rebase_feat to master
52af4db (rebase_feat) HEAD@{3}: rebase (finish): returning to refs/heads/rebase_feat
52af4db (rebase_feat) HEAD@{4}: rebase (pick): inget big data
f1429fb HEAD@{5}: rebase (start): checkout master
f01b054 HEAD@{6}: checkout: moving from master to rebase_feat
f1429fb HEAD@{7}: commit: ingets batch data
c53e34a HEAD@{8}: checkout: moving from rebase_feat to master
f01b054 HEAD@{9}: rebase (finish): returning to refs/heads/rebase_feat
f01b054 HEAD@{10}: rebase (start): checkout 505c5ab596e1aedfcc2d3db0446f9c8d4671f3f^
505c5ab HEAD@{11}: commit: ingets batch data
f01b054 HEAD@{12}: commit: inget big data
c53e34a HEAD@{13}: checkout: moving from master to rebase_feat
c53e34a HEAD@{14}: commit: finally conflict resolved
9197ed9 HEAD@{15}: merge bigfix2: Merge made by the 'ort' strategy.
69cdf37 HEAD@{16}: commit: bug is fixed
37ca628 HEAD@{17}: rebase (finish): returning to refs/heads/master
37ca628 HEAD@{18}: rebase (start): checkout 6f84ddf5ddcc34e39f11eb669f8c39ac4add3c53^
6f84ddf HEAD@{19}: commit: fixed bug 2
```

2. To go to the prev commit
-- git reset --hard HEAD~1 But not prefed
Or git reset --hard "HAED@{1}"

--- git reset --hard f0d1a12 prefed

Git diff --staged

And to reset it in staging only before commit just do.

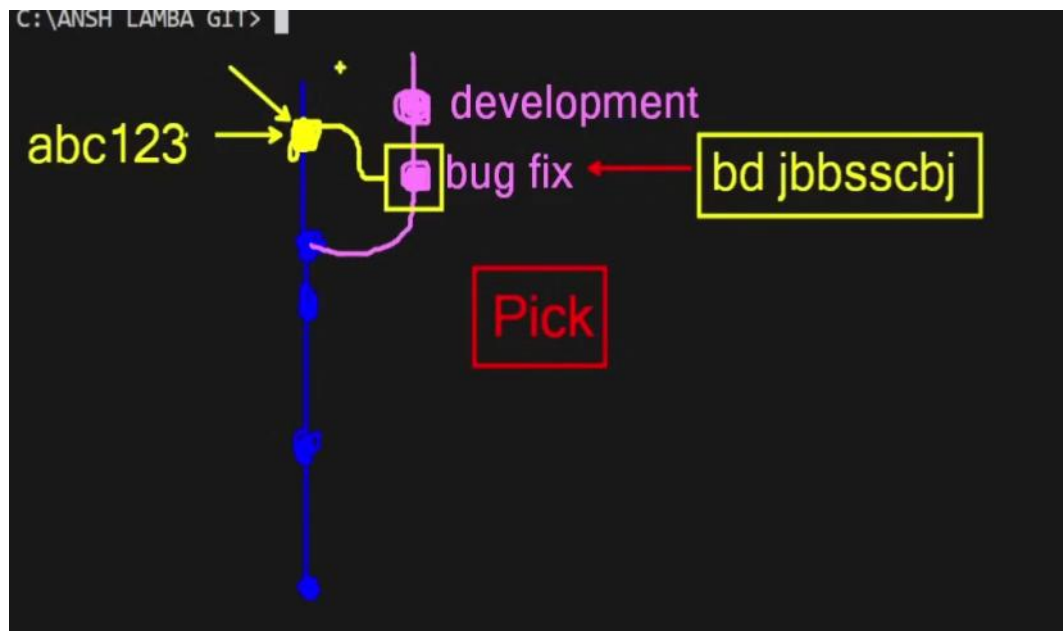
Git reset <file name if want only to changes in one file>

Cherry Pick

26 July 2025 11:59

`git cherry-pick <commitID>`

For multiple `git cherry-pick <commitID commitID ...>`



Git Stashing

26 July 2025 12:13

Stashing is a temporary storage location

Syntax

```
git stash push -m "My Dev Paused"
```

To bring back from the stash

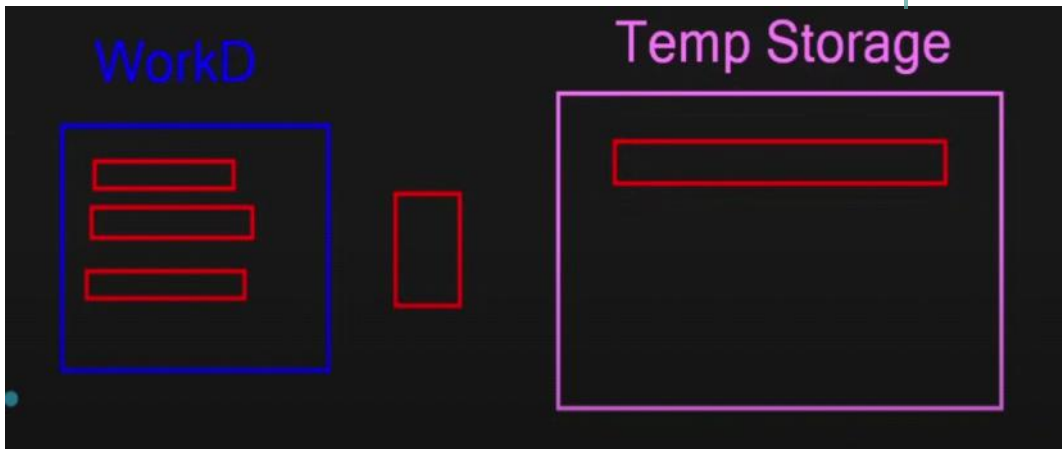
```
git stash apply <Stash~0 >
```

OR

```
git stash apply "stash@{0}"
```

1. To see the list of stash present
Git stash list

2. To delete the stash
Git stash pop



HOW it store

```
LAMBA GIT> git stash list  
On dev1: My Dev Paused  
LAMBA GIT> |
```



Git hub clone

27 July 2025 08:45

1. HOW to CLONE the Repo
git clone repo HTTP or SSH
2. If created new branch after doing git clone then first push the branch and merge it there
git push origin featureBranch
3. git pull origin main --allow-unrelated-histories --- pull from git hub
4. Git push origin main ____ to push in the git hub