

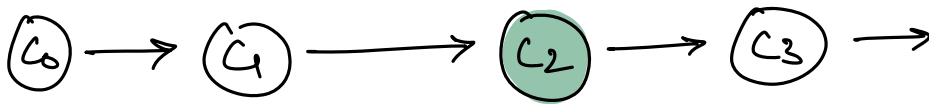
Agenda:

- Basics of GIT ✓
- Branching
- Merge
- Rebase
- Merge vs Rebase
- HEAD pointer.

⇒ GIT Commit.

↓

Distributed
Vcs.



Every commit stores the changes/delta w.r.t previous commit.

Every commit is Immutable.

↓ message

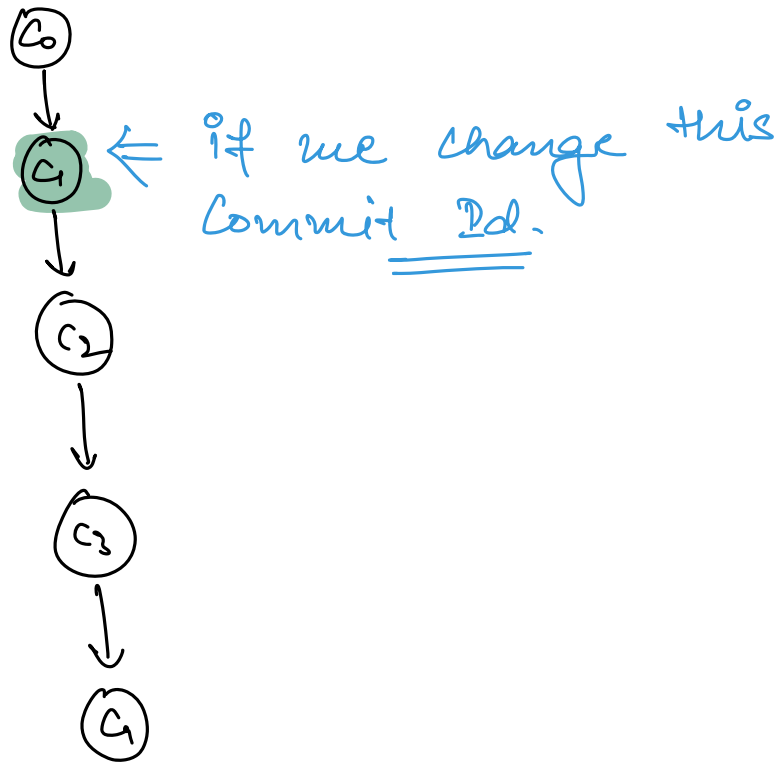
Id

timestamp

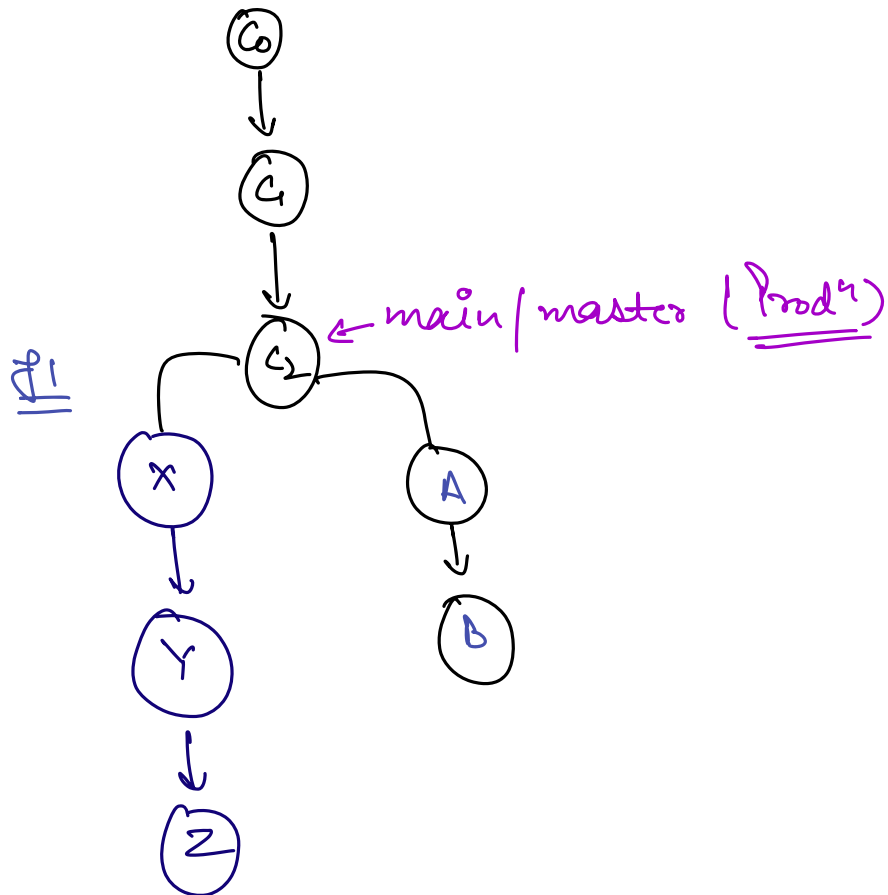
author

====

Commit Id = hash(author details, timestamp, ...
List<previous commit ids>)



Branching.

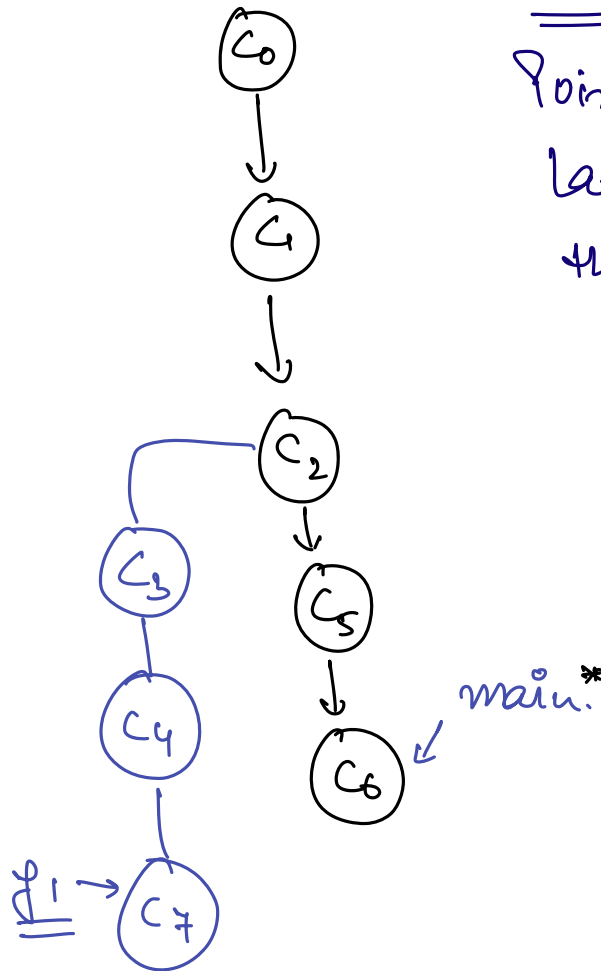


⇒ To work on a new feature, we create a separate branch to work parallelly.

Merge.

branch. name:

Pointer pointing to the latest commit of that branch.

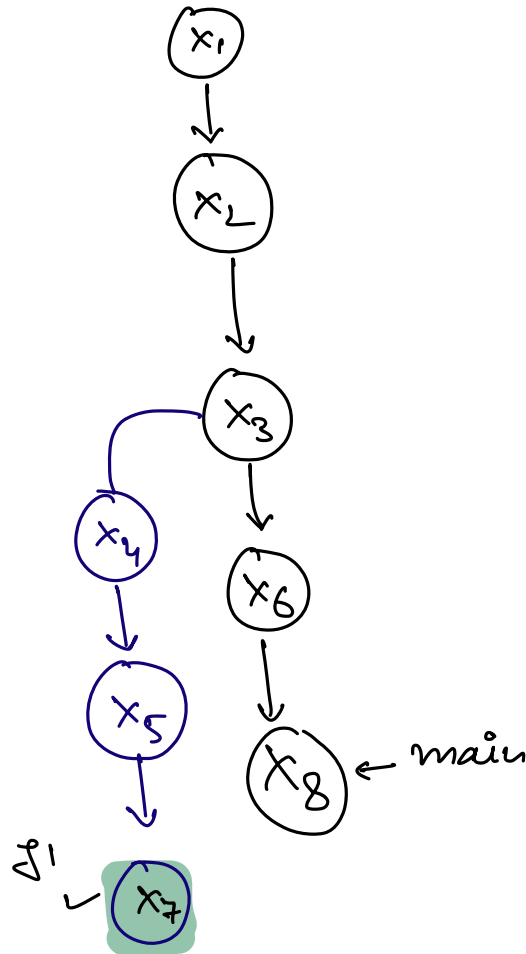


⇒ Once the feature has been implemented & tested completely, we'll add our branch changes to the Main branch

⇒ Merge

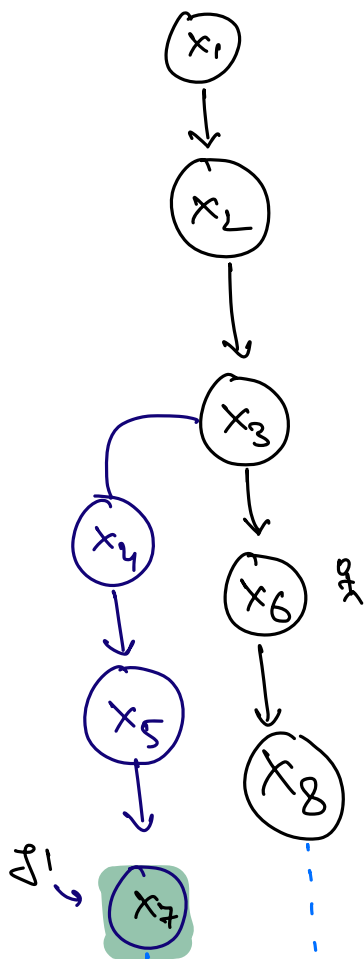
GIT Merge.

Command used to bring changes from one branch to another branch.



bringing our changes
(x4, x5, x7) to
the main branch

⇒ Merge f1 branch into Main.



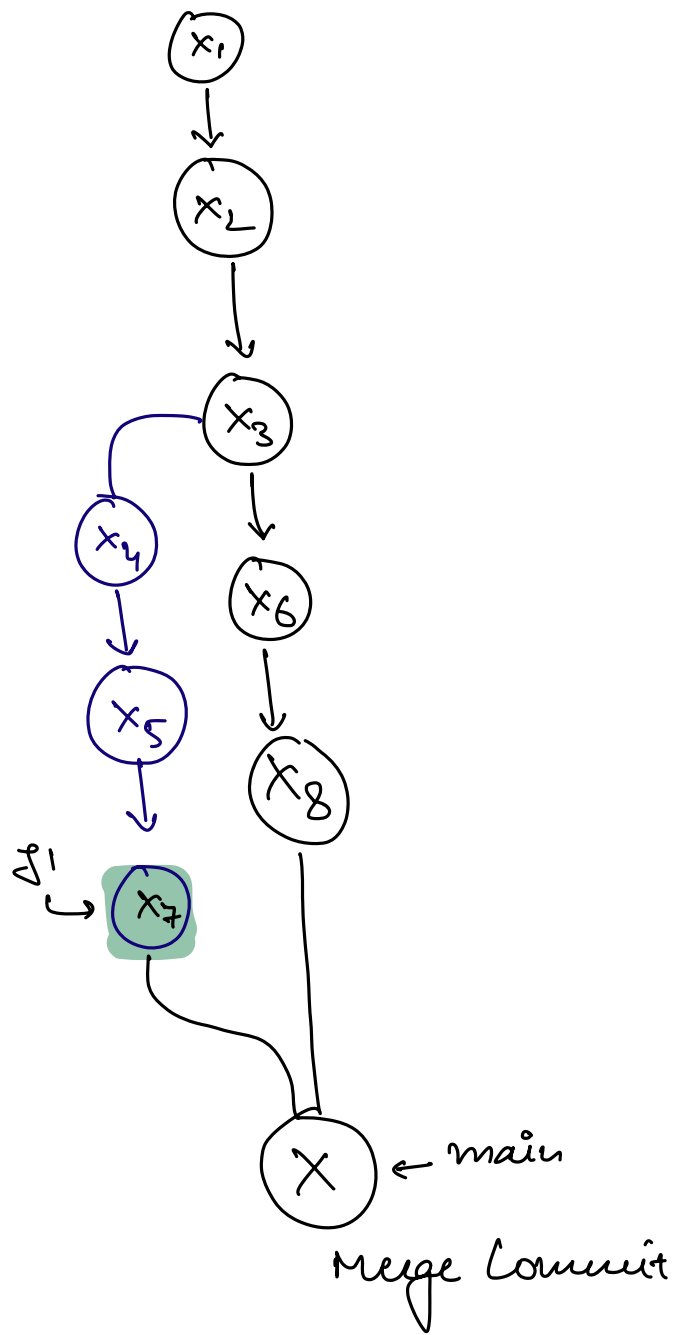
1) go to main

git checkout main

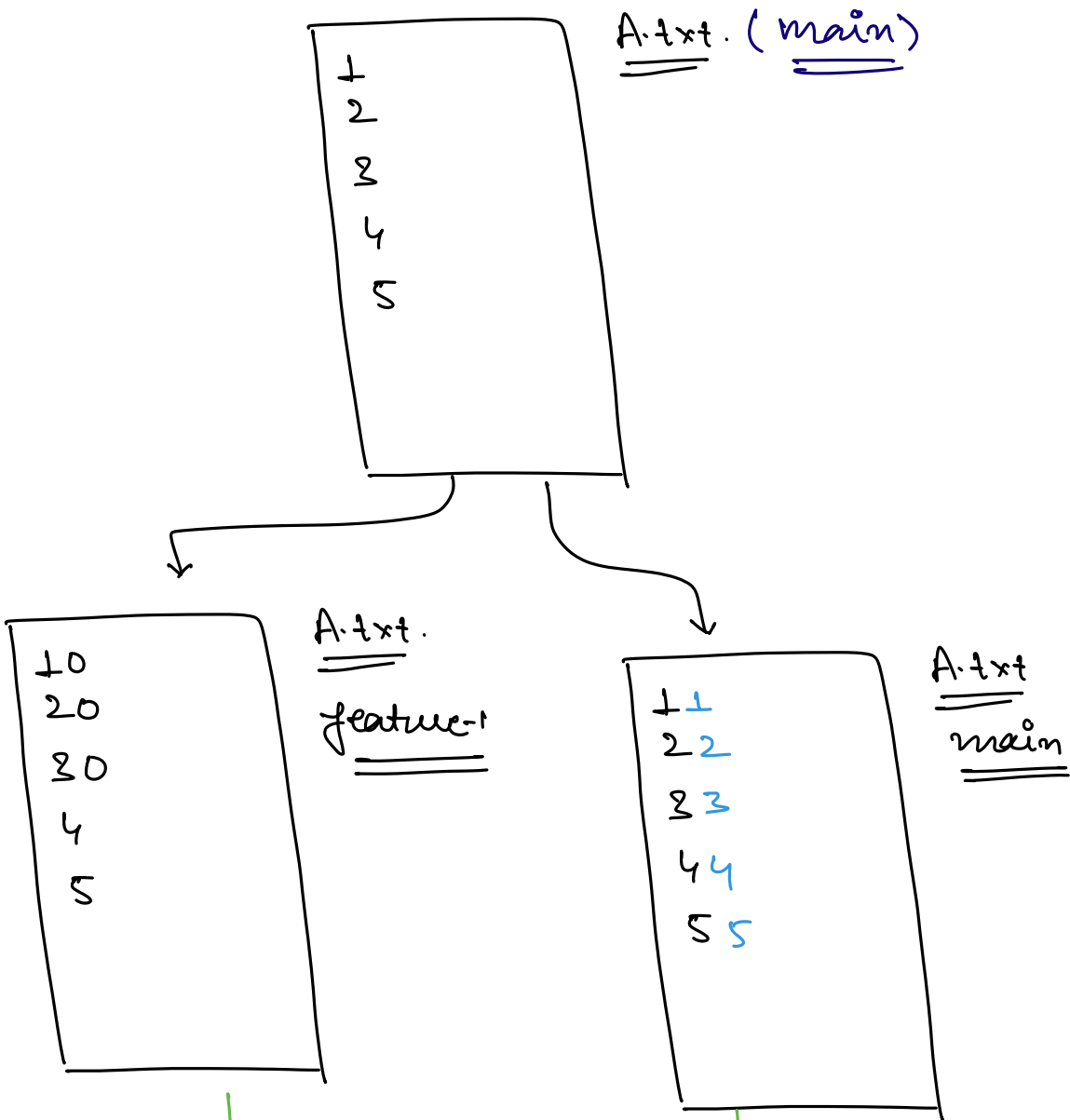
2) git merge #1

← main*

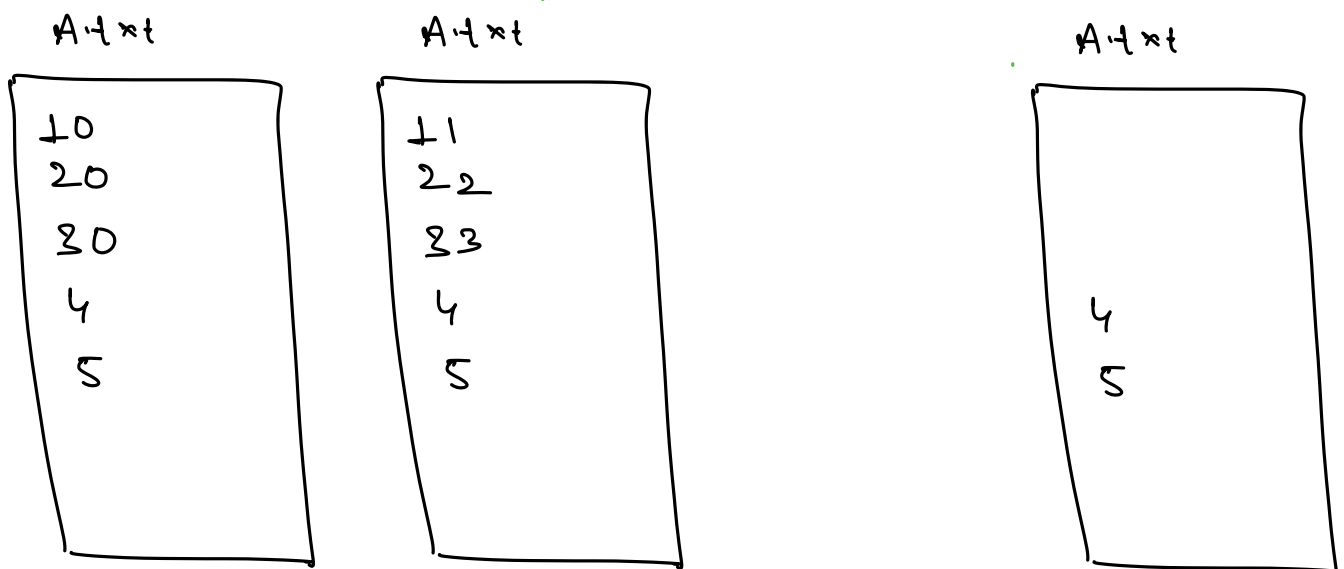
← Merge Commit (2 Parent)



\Rightarrow Merge



Merge Conflict



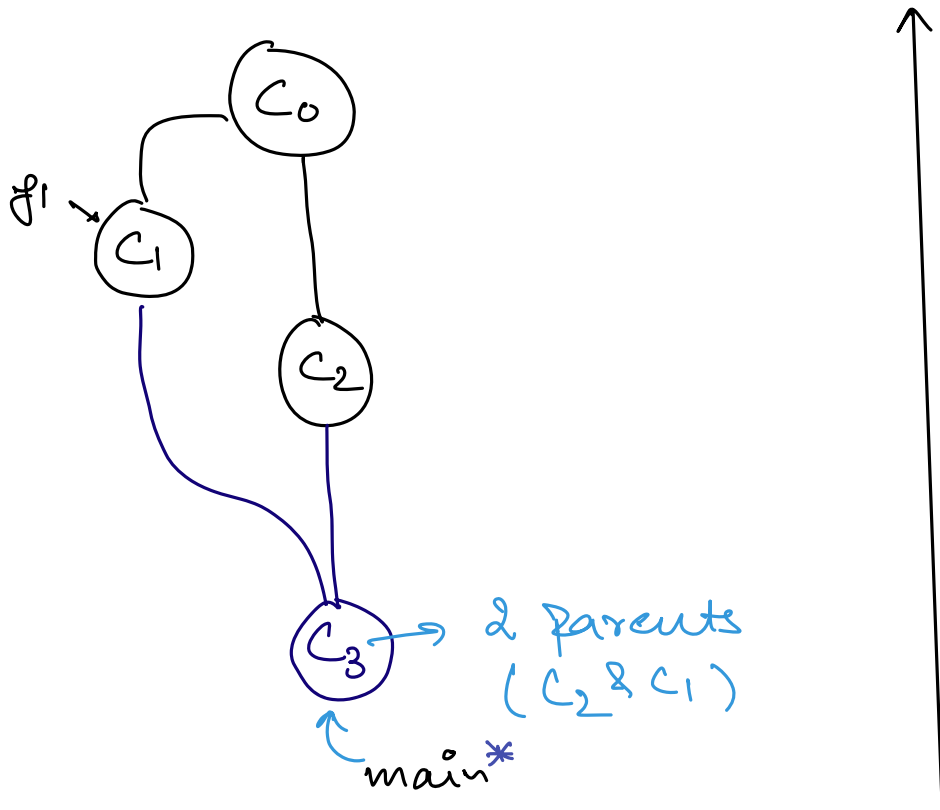
A.txt

```

10
20
30
11
22
33
1
1

```

⇒ We'll have to resolve conflicts manually.

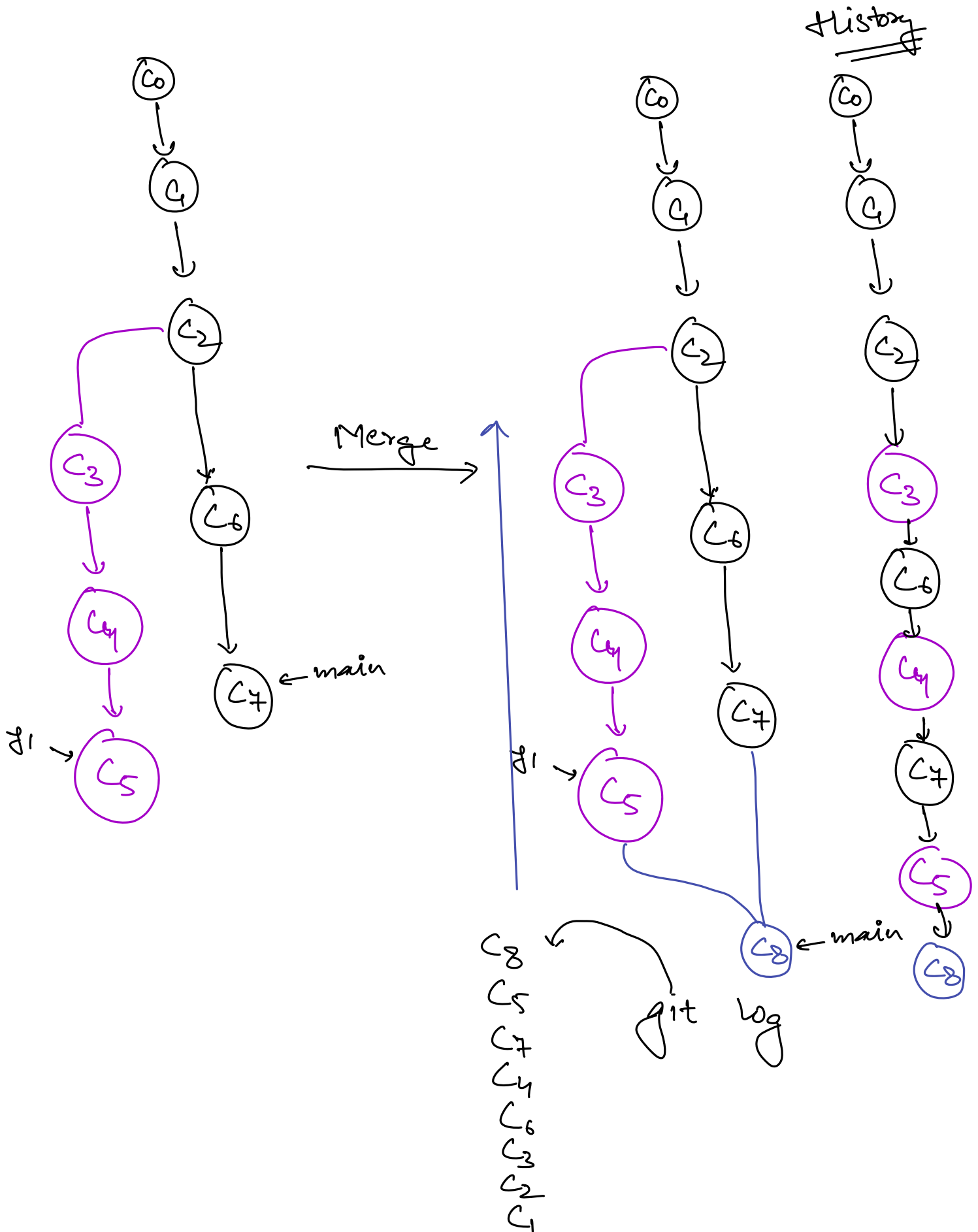


⇒ git log

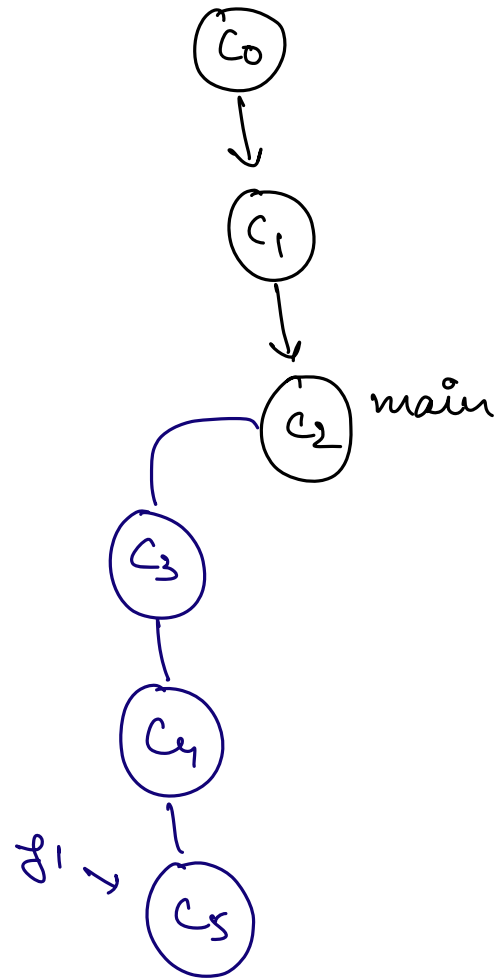
history of commits

- C₃
- C₂
- C₁
- C₀

git checkout -b j2 \Rightarrow create branch j2
& checkout to j2.



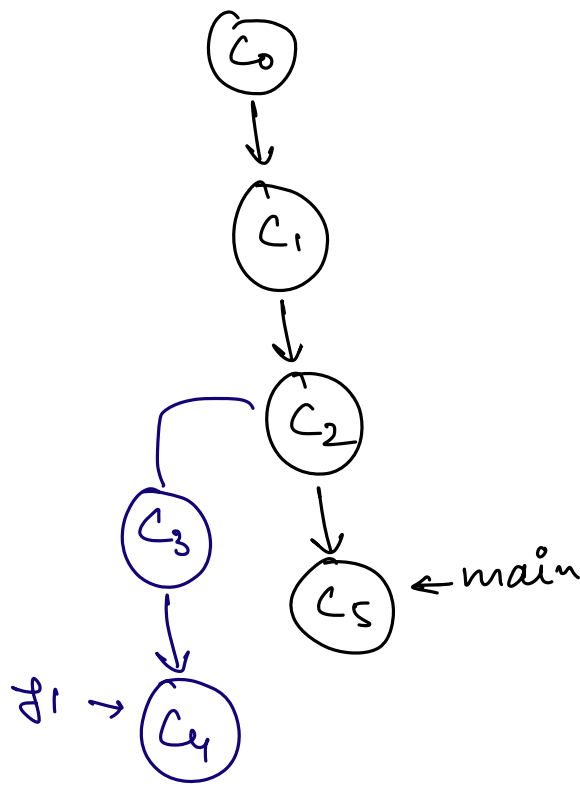
- ⇒ In the merge commit, history is getting overlapped from different branches.
- ⇒ Because of overlapping, debugging can be difficult.



Merge f1 into main.

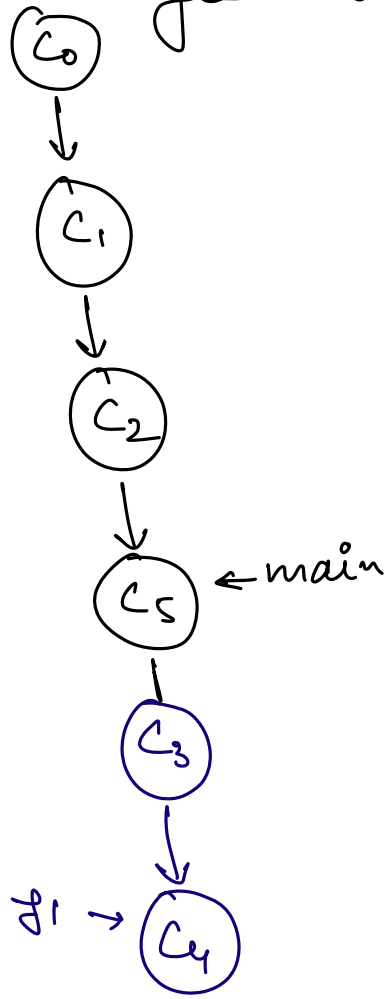
⇒ fast forward merge.

⇒ No merge commit will be created.



⇒ Rebase.

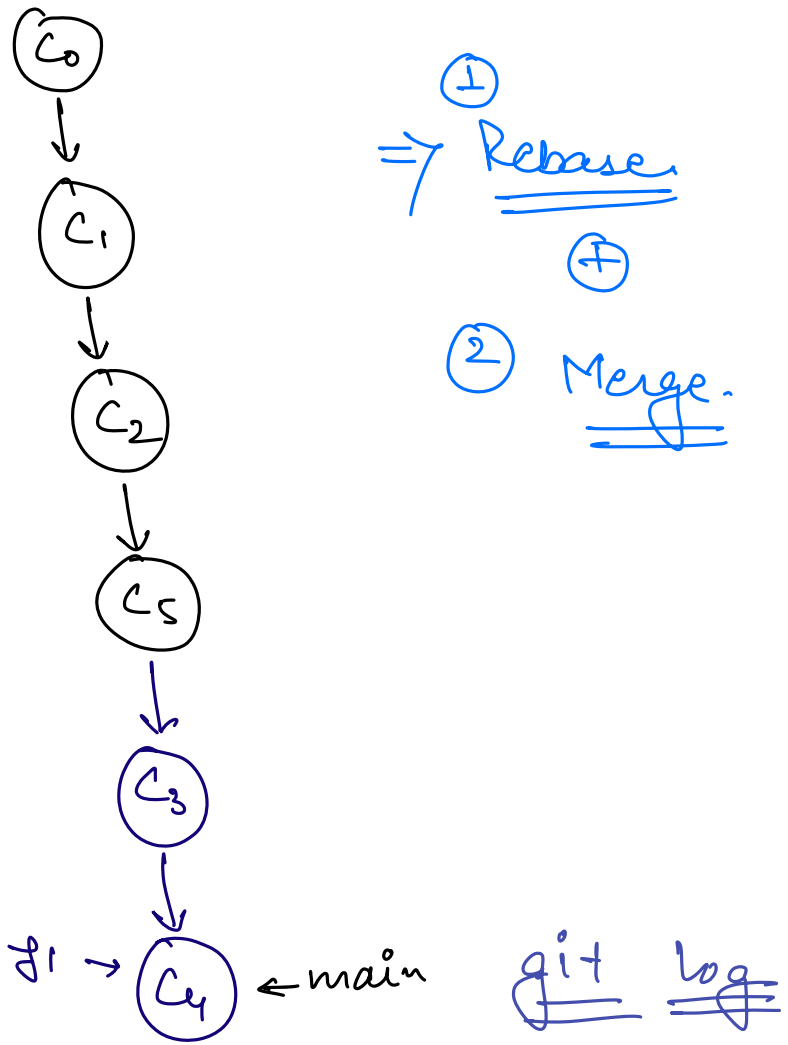
GIT Rebase.: Allows us to remap all the changes that we have done in a new branch below the latest changes in the other branch.



- 1) git checkout f1.
- 2) git rebase main

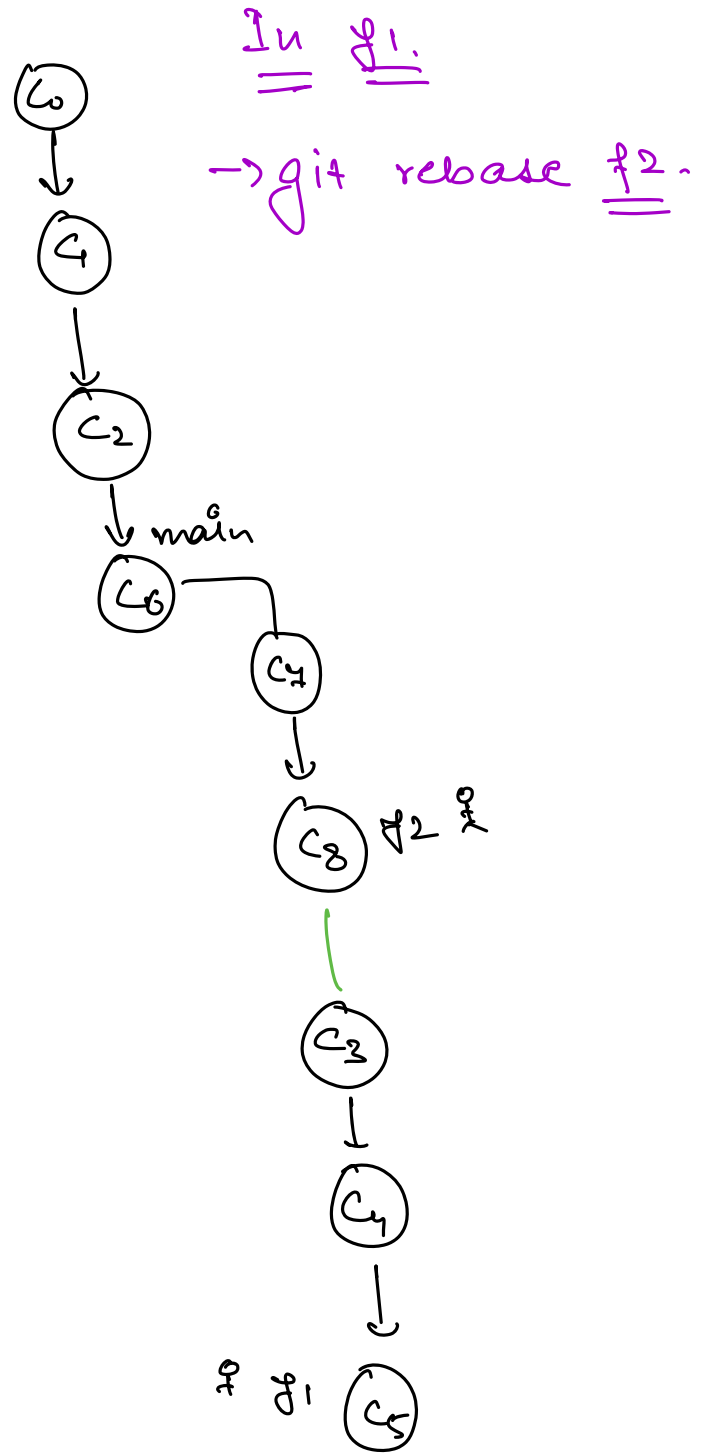
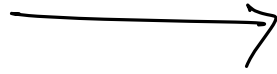
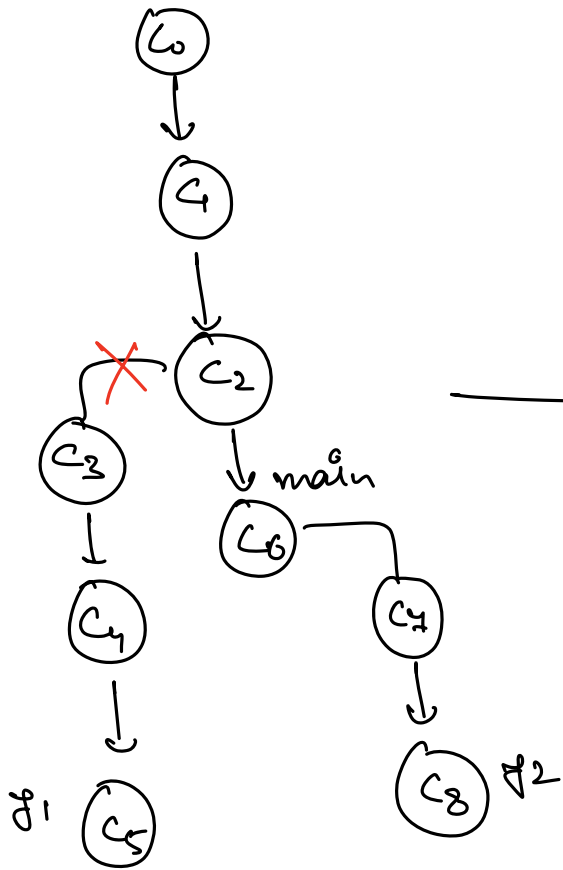
⇒ No change in the main branch.

#



Rebase + Merge \Rightarrow Cleaner History.

Merge (vs) Rebase + Merge



HEAD.

↳ Special pointer in GIT that always points to the latest commit.



Commit on which the
new commit will be
added.

Detached HEAD.

When HEAD pointer points to commit instead
of pointing to a branch