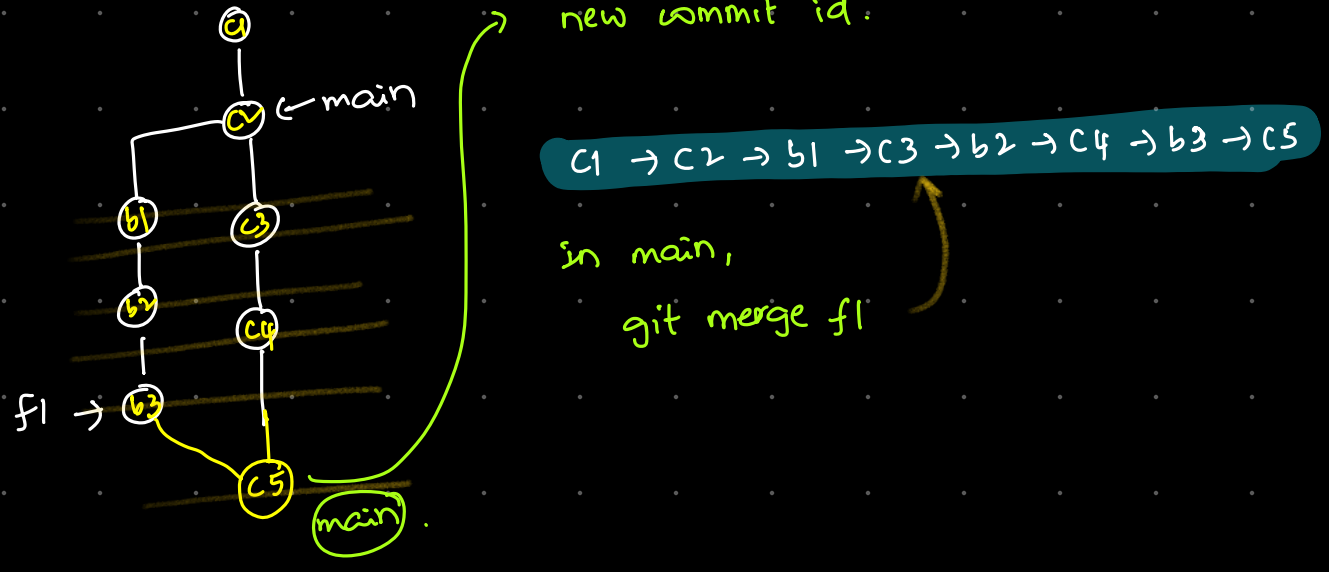
Recap.

- → commits
- → branching
- → merge
- → conflicts
- → simple git remotes.

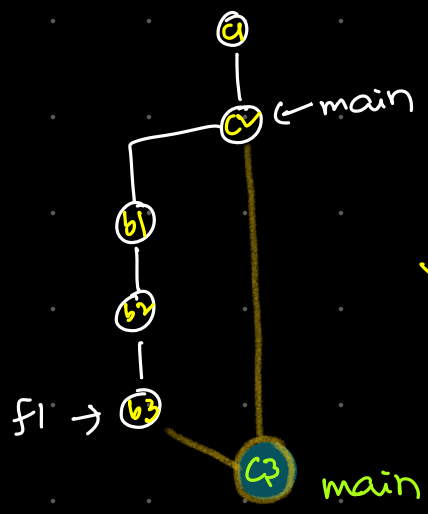
Today

- fast forward
- rebase
- git remote

Revisiting merge



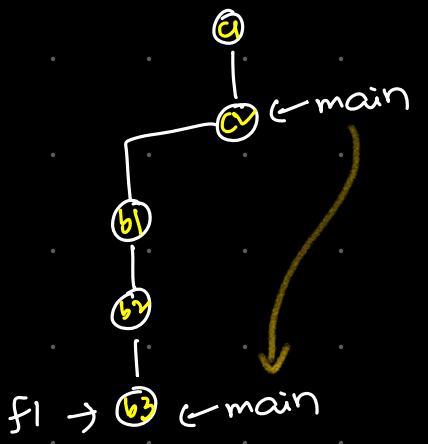
After you checked out f1, you never did any changes in main.



X

commit log

c1 → c2 → b1 → b2 → b3 → c3.



branch

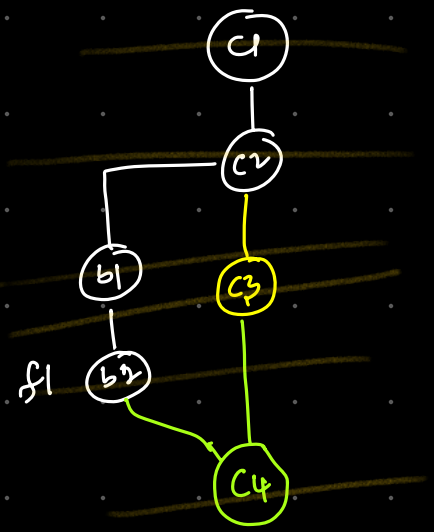
A pointer to the latest commit.

In-order to merge f1 in main.

c1 → c2 → b1 → b2 → b3

fast forward merge.

The problem with timestamp based commits history.



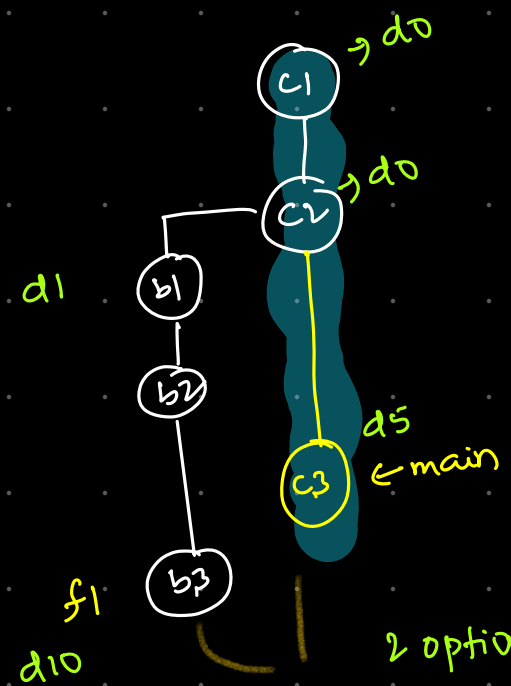
git log

c1 → c2 → b1 → c3 → b2 → c4.

dev1 → 3 commits

dev2 → 4 commits

dev3 → 5 commits

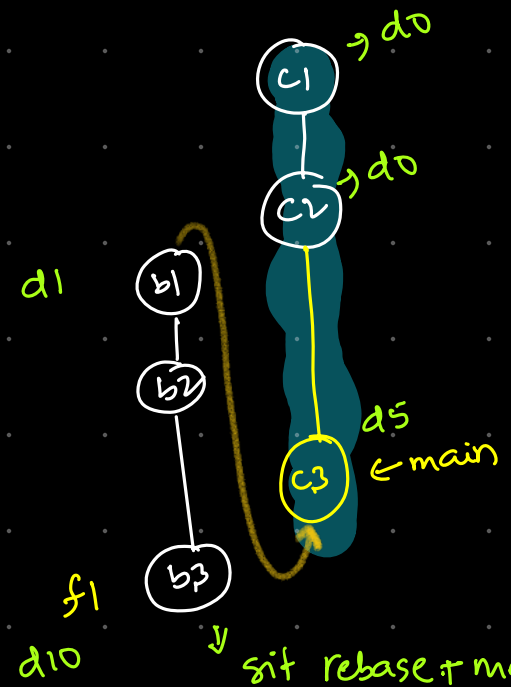


2 options.

① c1 → c2 → b1 → b2 → c3 → b3 → c10

② c1 → c2 → c3 → b1 → b2 → b3

rebase



git rebase + merge

In f1

rebase to main

In main

You merge f1

Git Remote

1. create a new proj in local
2. create a new repo in github ; remote
3. for the local repo set up the remote url
4. Push your changes to remote.

git pull origin main