Agenda

git init
git commit
git branch
git merge
git rebase

git init

Parent → "git init"

└ Child 1

└ Child 2

└ Subchild.txt

⋮ ⋮

└ .git

⇓

git repository

git commit

Commit ⇒ save

1. git add <file1> <file2> <file3> ....
   eg., git add add.java mult.java

2. git commit

git add. → Add all the files that are modified to the staging area

```
git commit -m <message>
```

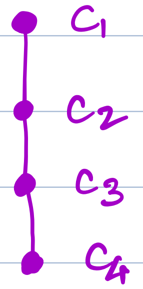
eg., `git commit -m "adding addition & multiplication"`

Properties of commit

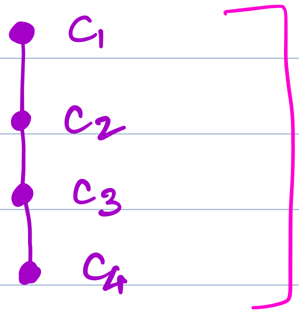
1. Commits are immutable (and permanent) → Can't be tampered with

→ Do not share confidential information in a commit

2.



Commits only store the deltas (differences)



If we want to construct the code at some commit, we have to start from the top

⇓

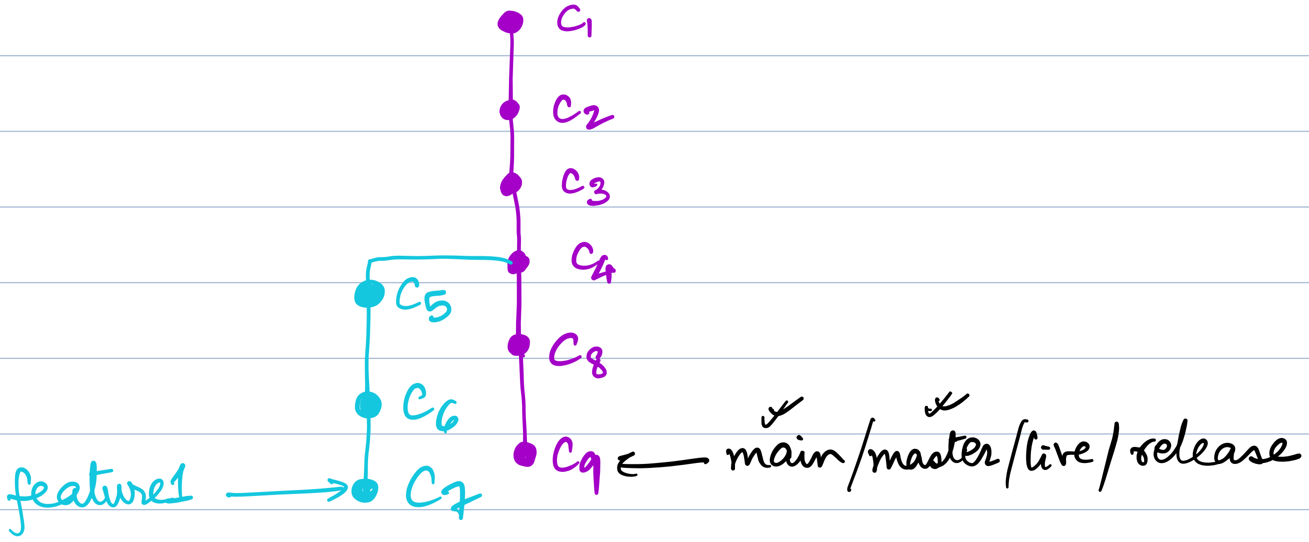
slower process

Details stored in commit

① Who made the commit

② When was the commit made
③ Deltas in the commit
④ Id of the commit

git branch

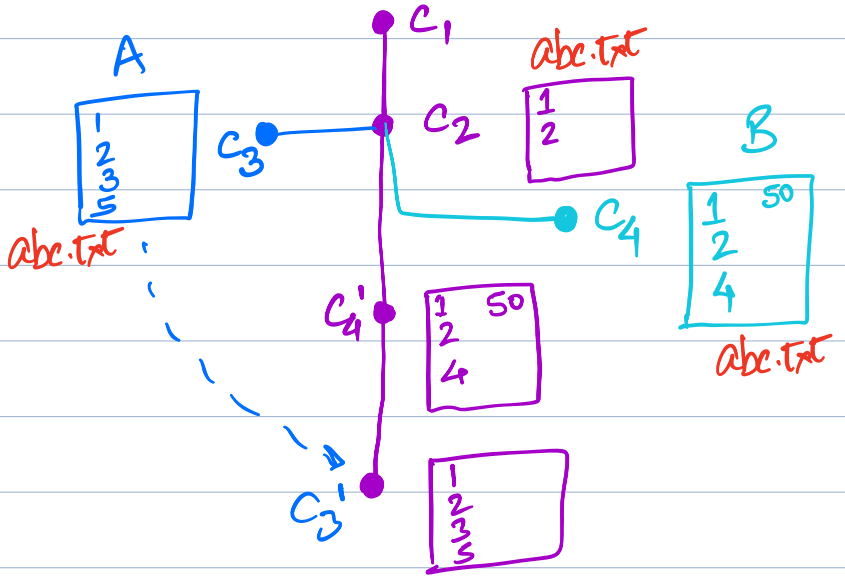


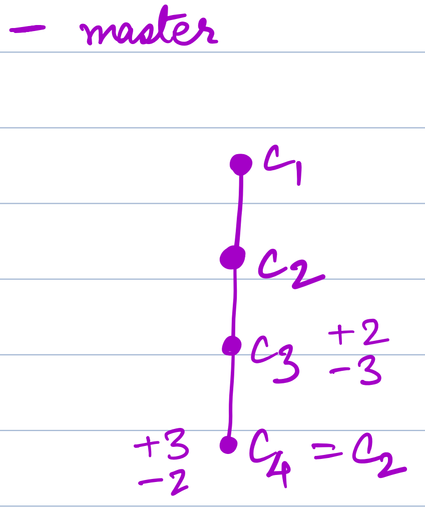
- Parallel line of production
- Name given to the latest commit in a series of commits

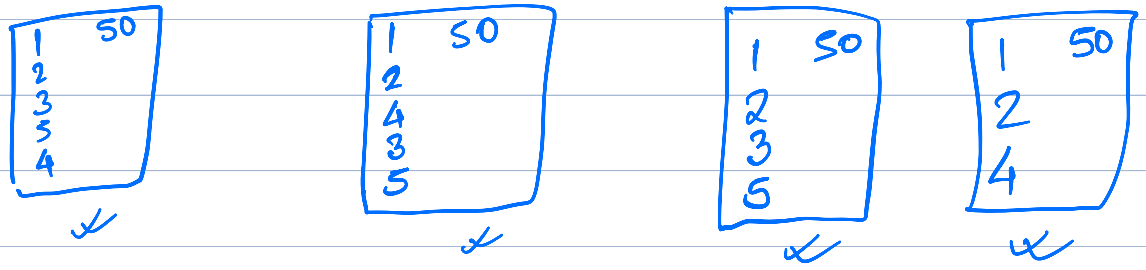
```
git branch <branch_name>   ⇒ Create a new branch
git checkout <branch_name>

→ git checkout -b <branch_name>
```

git merge

Bringing the changes of one branch to another
→ typically feature to master

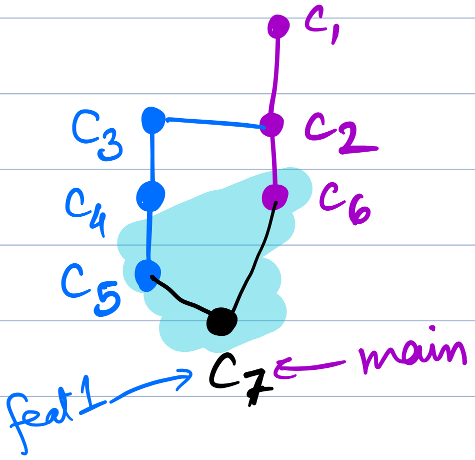






Types of Merge

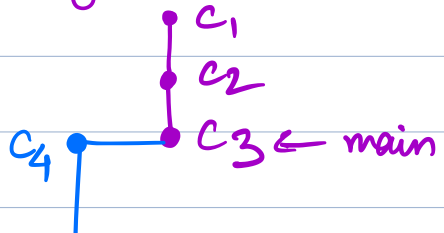
① 3-way Merge

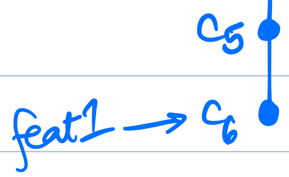


git checkout main

git merge feat1

② Fast-forward merge



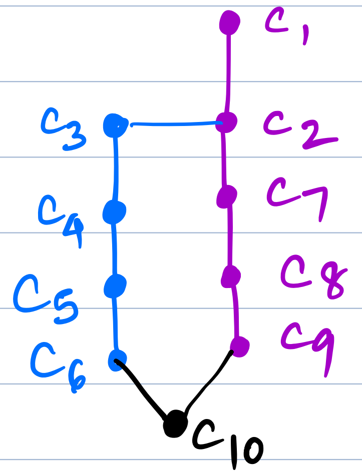


- Main should not have any commits from the time the feature branch was checked out to the time it was merged to main

- Main should be an ancestor of feature

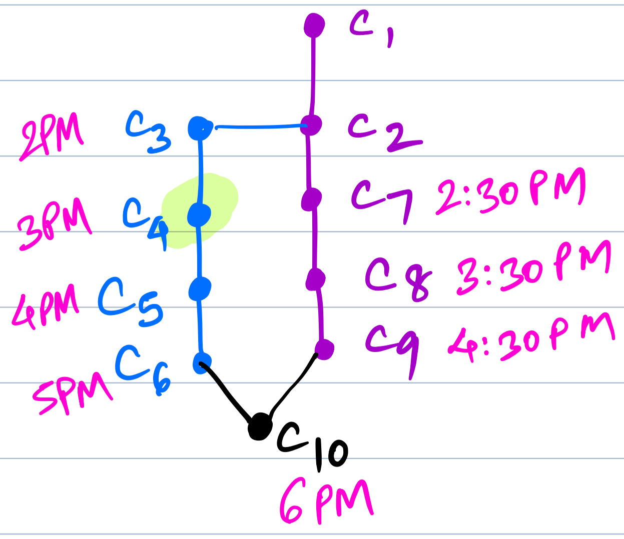
Problems with 3-way merge

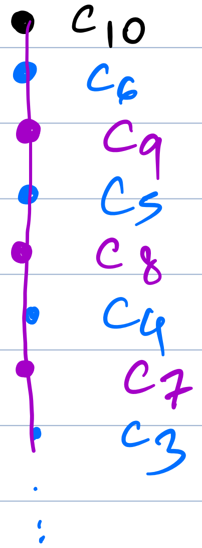
①



Many merge conflicts

②



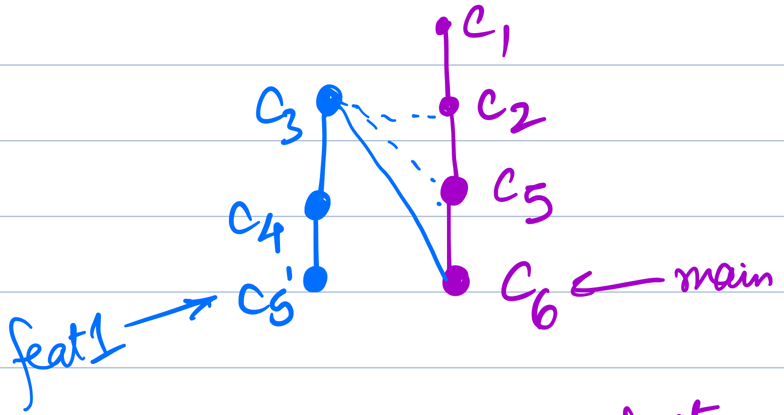


Jumbled up git history

Big Idea

1. FF Merges are ideal
2. Cleaner git history

git rebase



- Rebasing is changing the first parent of the feature branch ⇓ after it was checked out to the latest commit of main/master branch.

git checkout feat1

```
git rebase main
[git Checkout main
 git merge feat1]
```

→ you can do it after the feature work is complete

Advantages of rebase

1. Lesser merge conflicts
2. Cleaner git history

Tips to rebase

1. Rebase frequently, ideally twice a day
2. Rebase before merge