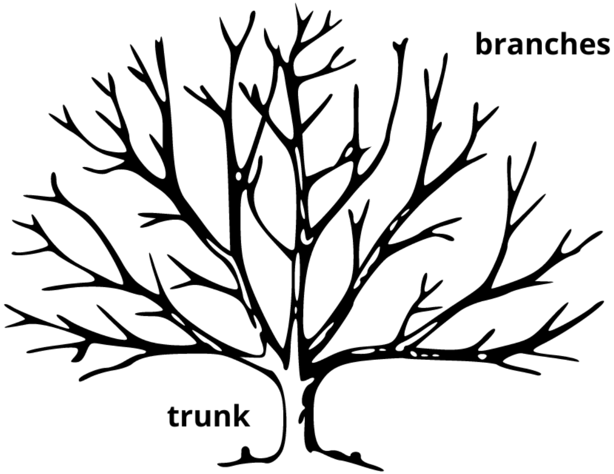
IT Club | uxcam

GIT WORKSHOP

Day 2: Branching

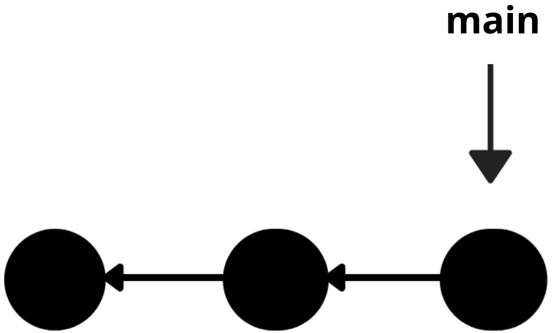
Introduction to Branching

- Similar to branch of a tree
- Creates copies of programs or objects in the development process
- Allows you to work in the parallel environment



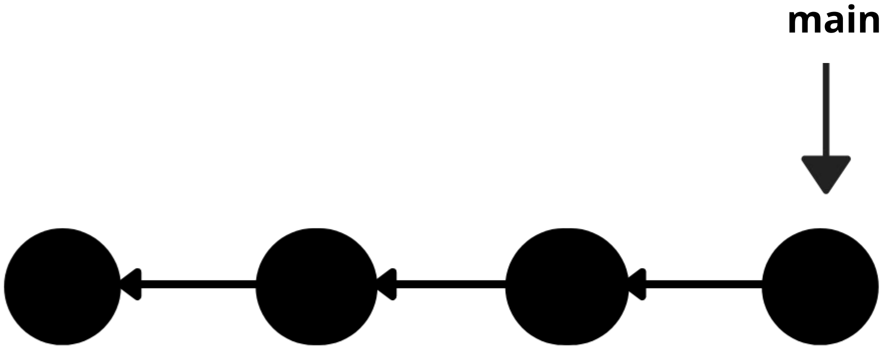
Main Branch

- Similar to the trunk of the tree
- Created as soon as you create your repository (default branch)
- Head points to the last commit you made
- Moves forward automatically



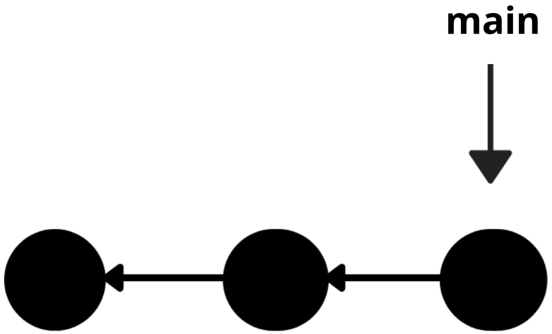
Main Branch

On adding another commit:



Why Branching?

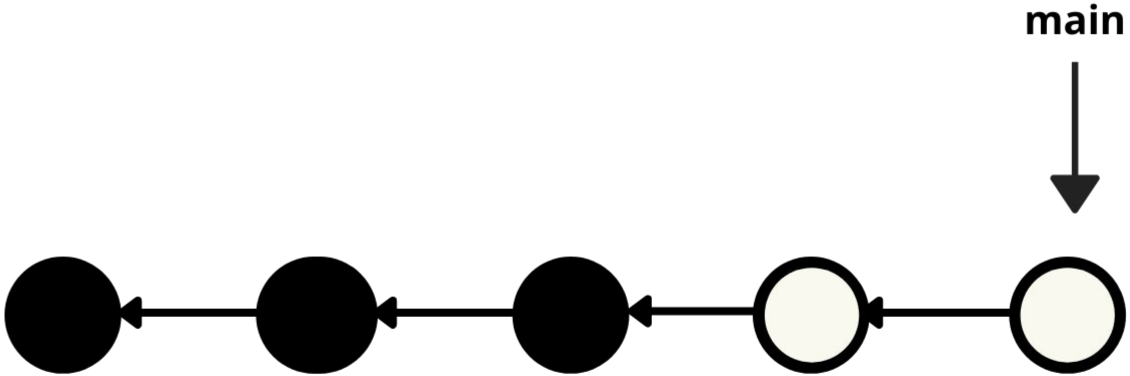
- Let's see an example



Initial Condition

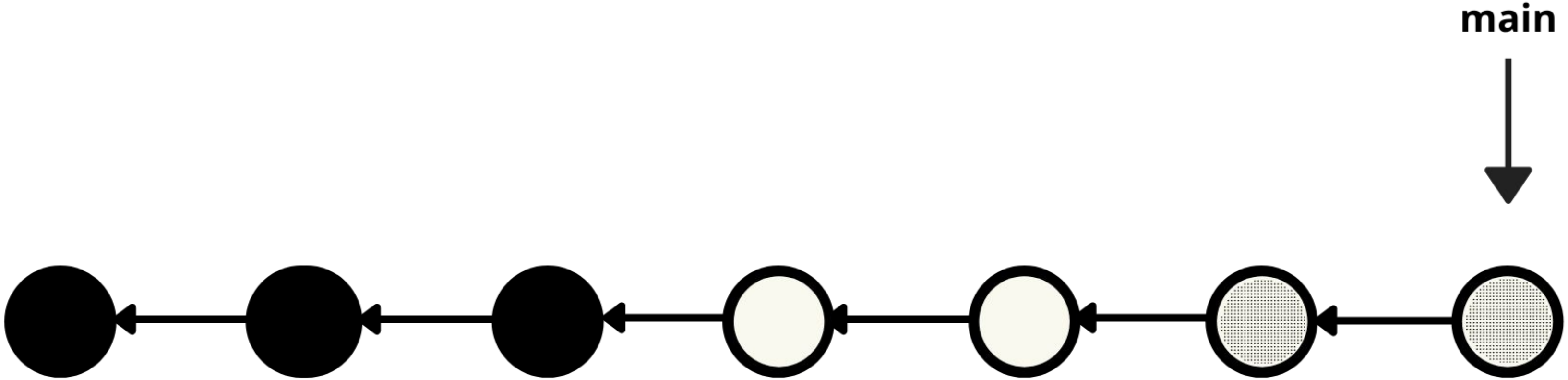
Why Branching?

❑ You want to add a feature



Why Branching?

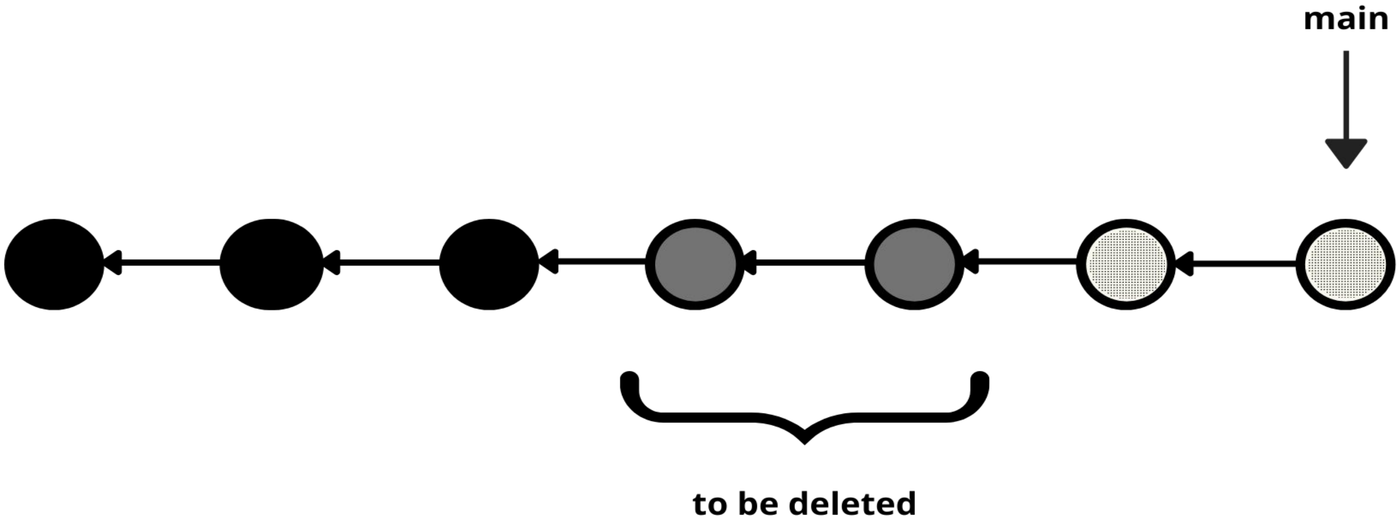
❑ Your project partner adds another feature



Why Branching?

❑ Later you realize, your feature was buggy and you have to delete it for the project to run

❑ Your only option is to delete your friend's feature as well

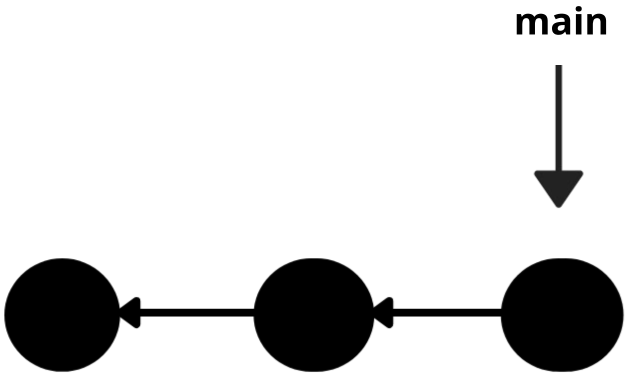


Why Branching?

What naming convention you use for git branches?
What's a branch? I only work in master.
imgflip.com

Why Branching?

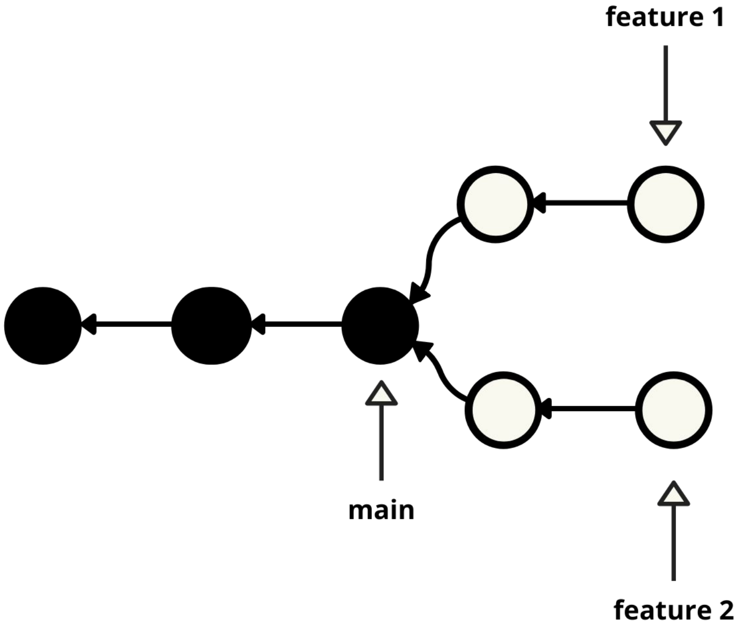
❑ If you had used branching,



Initial Condition

Why Branching?

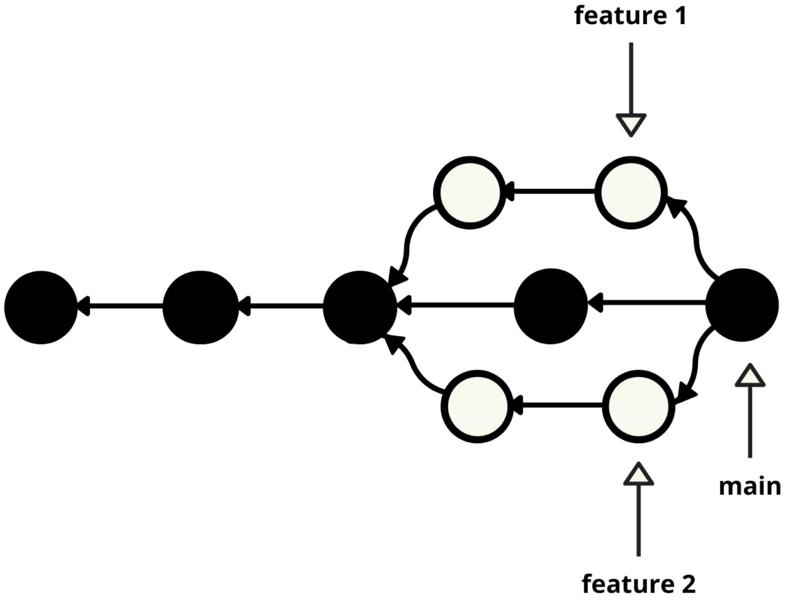
- Both you and your friend work on your respective feature on your respective branches



Why Branching?

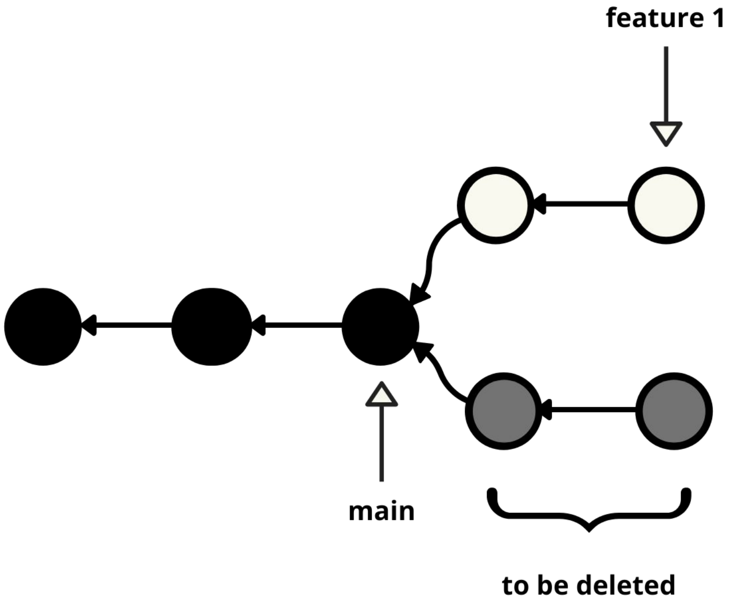
❑ After you are happy with your features, you can combine them

❑ Combining is technically called merging



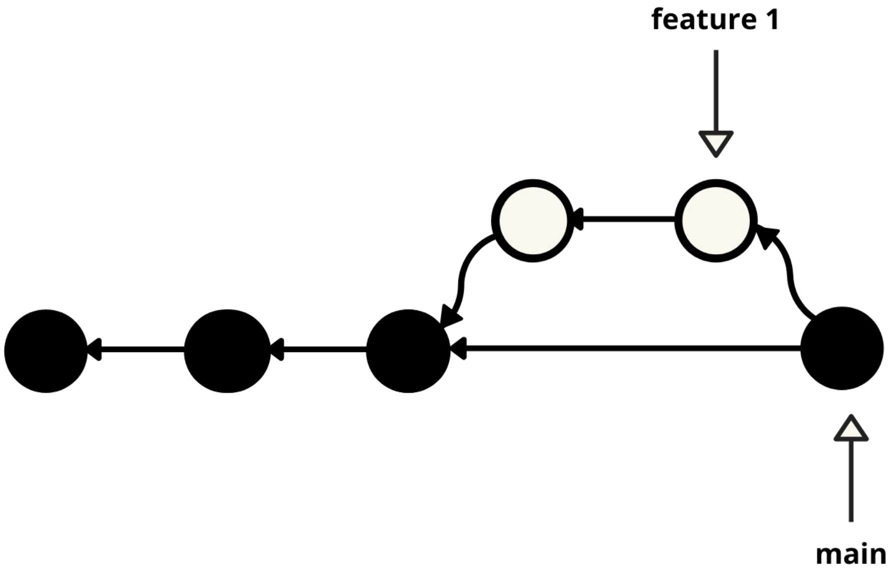
Why Branching?

❑ If you are not happy with your feature, you can delete it very simply



Why Branching?

- You can combine the remaining brach with your main branch



Why Branching?

Important Commands

git branch : List branches (the asterisk (*) denotes the current branch)

git branch [branch name] : Create a new branch

git branch -d [branch name] : Delete a branch

git checkout -b [branch name] : Create a new branch and switch to it

git checkout [branch name] : Switch to a branch

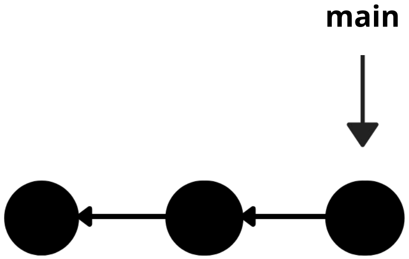
git branch -m [old branch name] [new branch name] : Rename a local branch

Time for some demonstrations!

Time for some demonstrations!

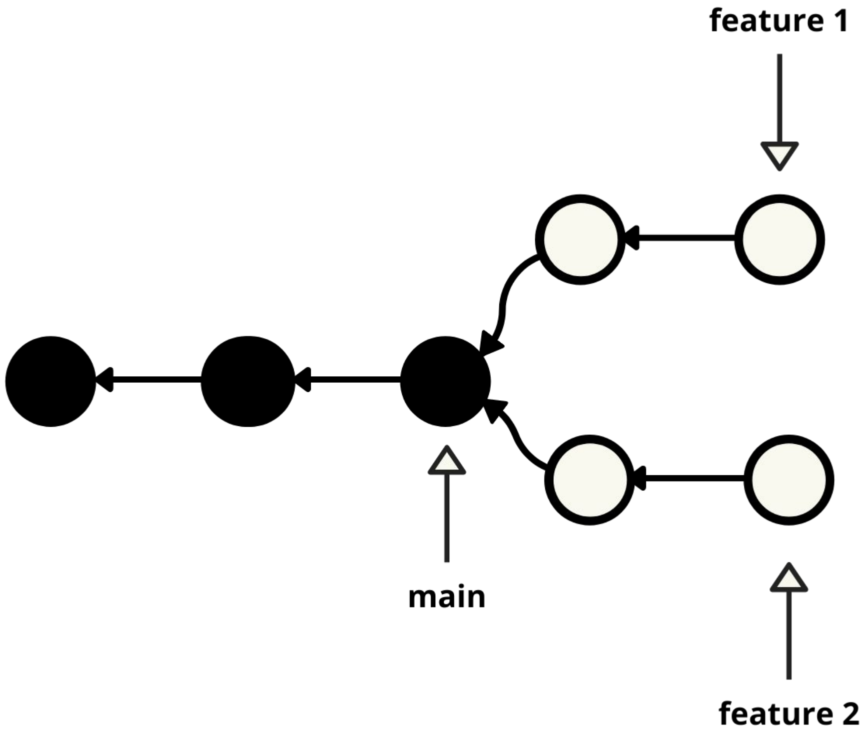
Merging

- ❑ Simply, merging means combining
- ❑ Different lines of code present in independent branches are integrated into a single branch (generally the main branch)
- ❑ Let us revisit our example

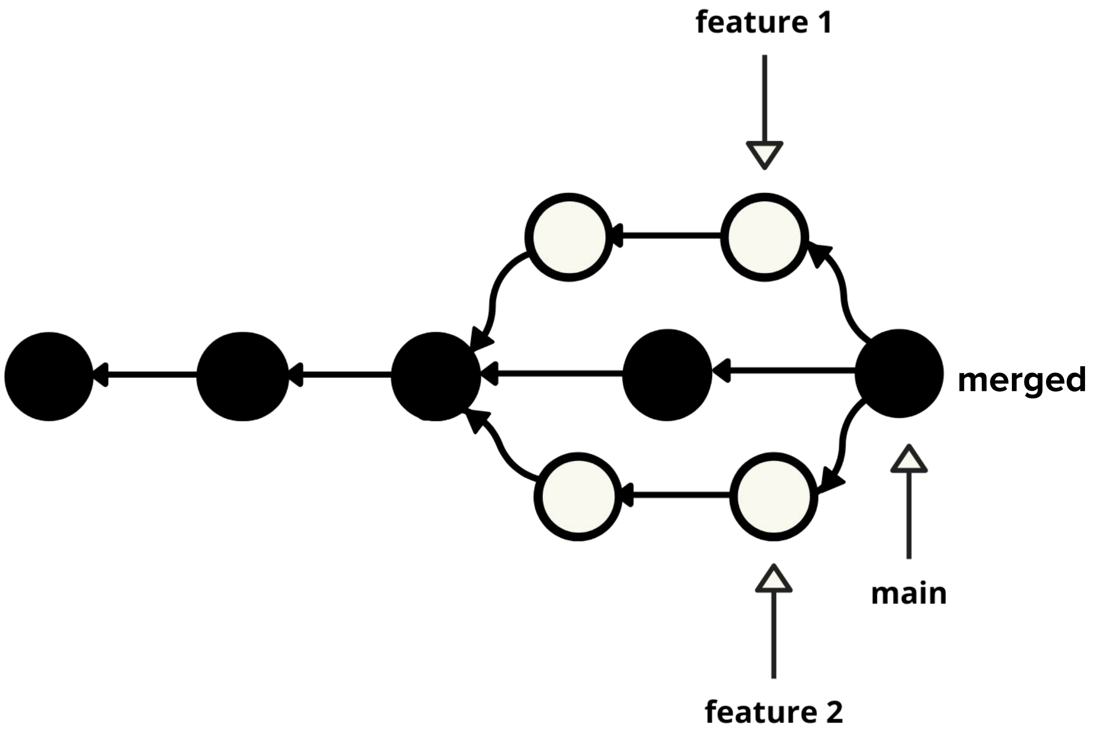


Initial Condition

Merging



Merging



Merging

Important Commands

git merge [branch name] : Merge a branch into the active branch

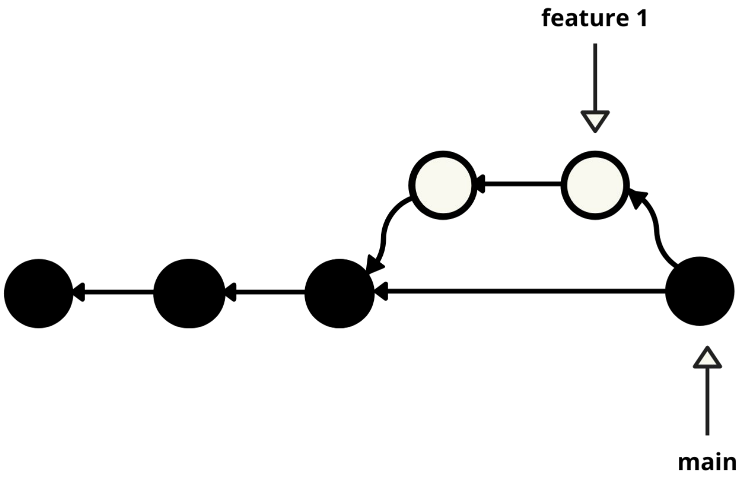
git merge [source branch] [target branch] : Merge a branch into a target branch

Time for some visualizations

Time for some demonstrations!

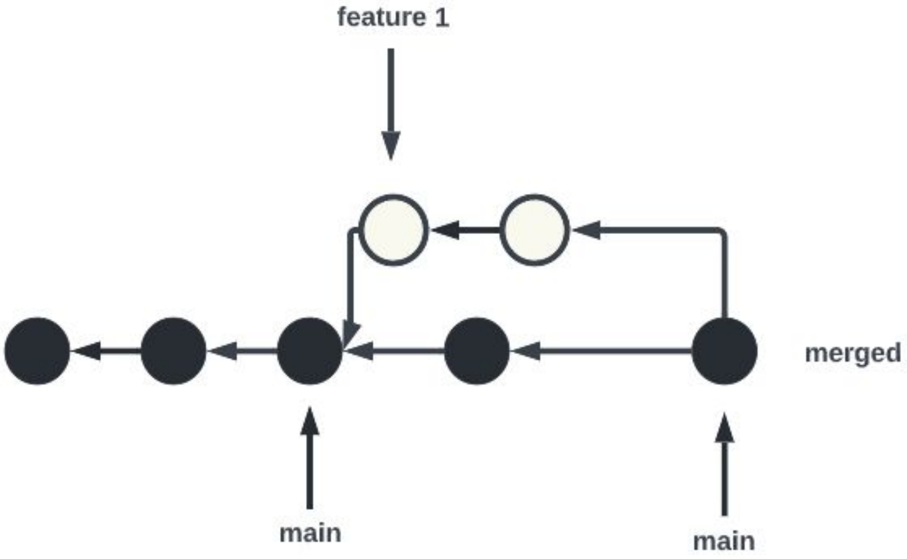
Fast Forwarding

- ❑ Merging that occurs when there is a linear path from the current branch tip to the target branch
- ❑ In fast forwarding, instead of merging, the histories are integrated



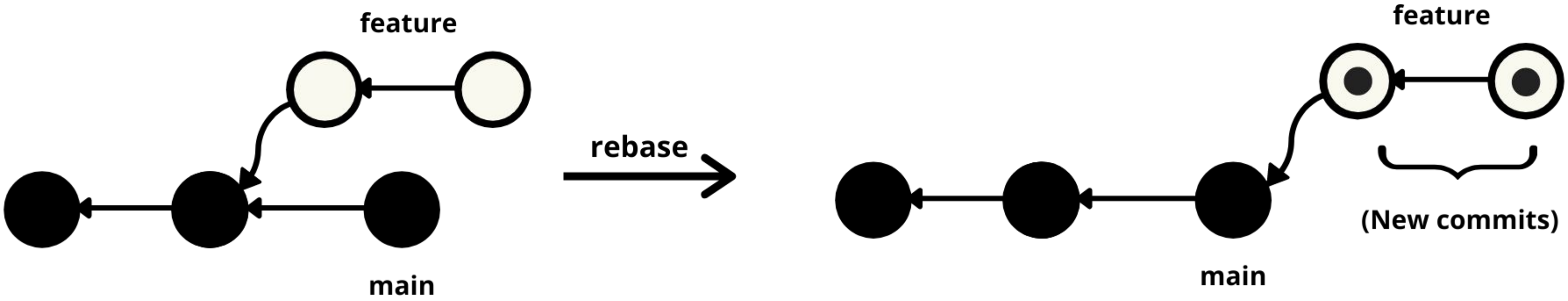
Merging by Recursive Strategy

❏ Default Merging

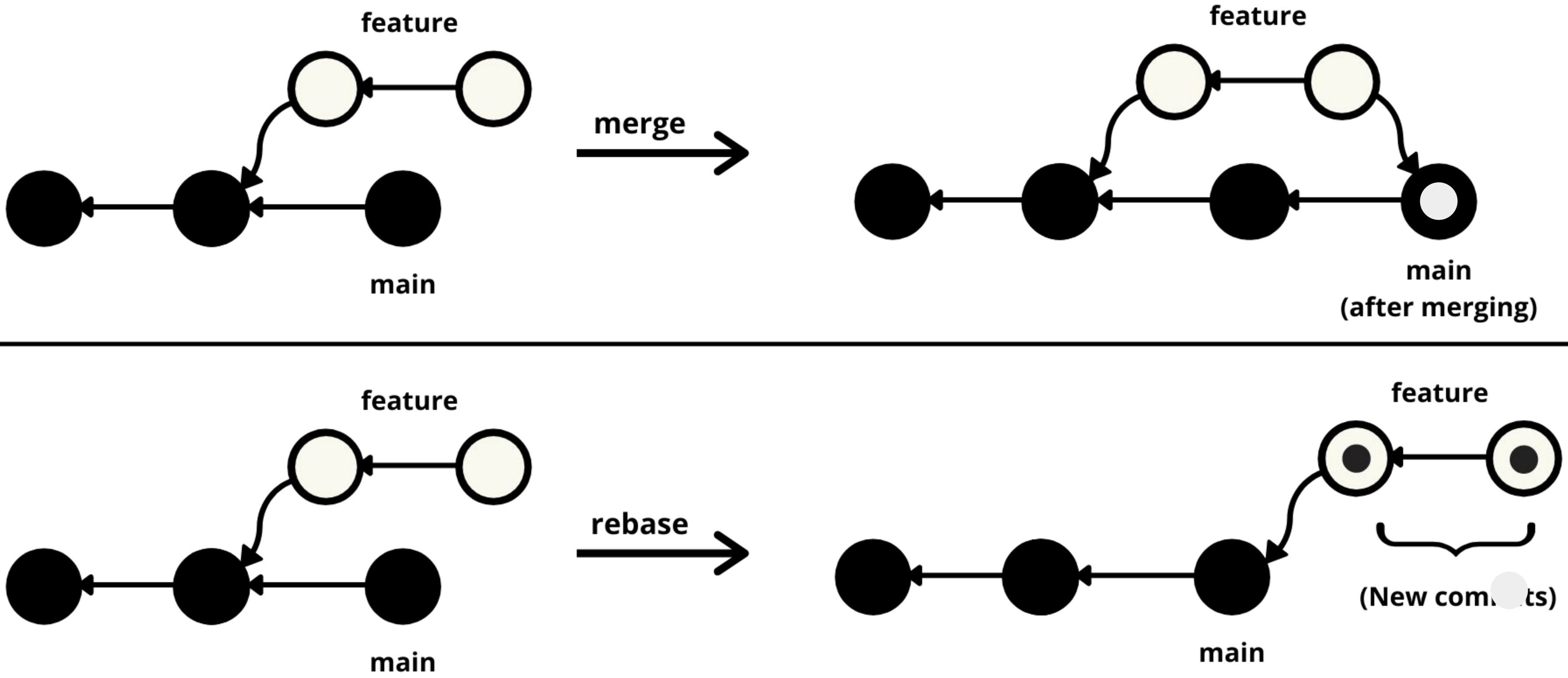


Rebasing

- Another way to integrate changes from one branch to another.
- Rewinds the head to replay your work on top of it.
- Rewrites history by creating new commits

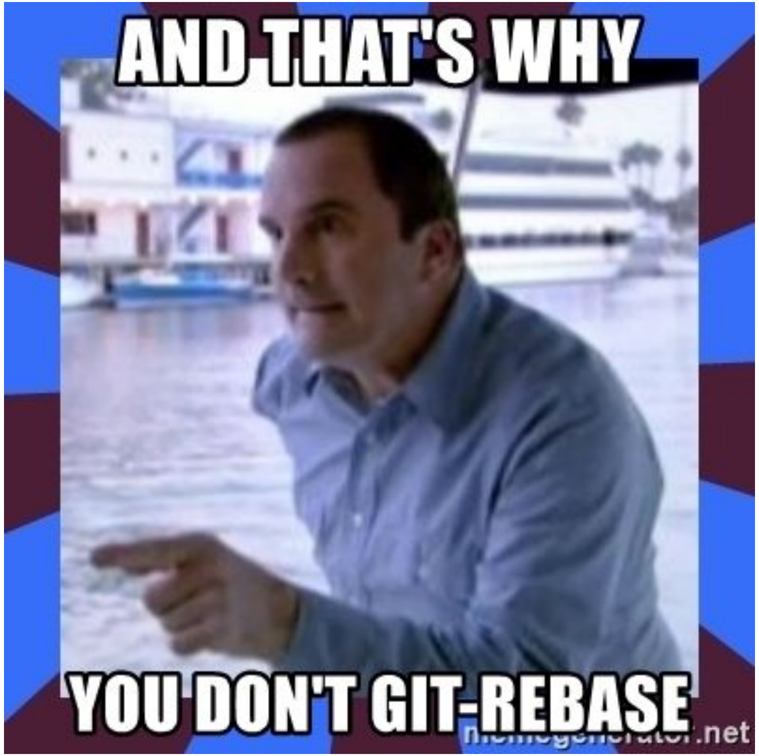


Merge vs. Rebase



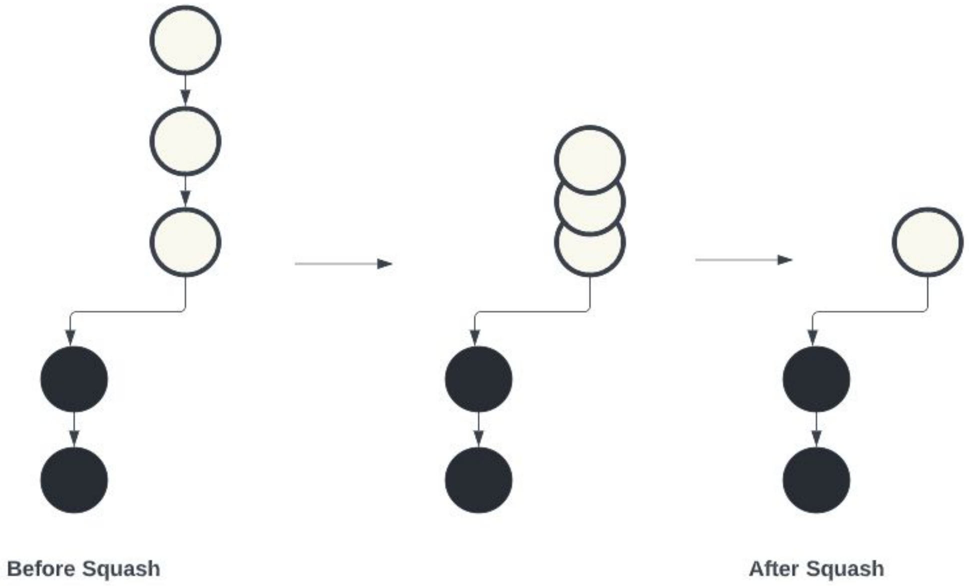
Time for some demonstrations!

Merge vs. Rebase



Squashing in Git

- ❑ Combine multiple commits into one
- ❑ To do some clean-up before merging



Time for some demonstrations!

Time for **the final** demonstrations!

BRACE YOURSELF
MERGE CONFLICTS ARE COMING
memegenerator.net

Questions?