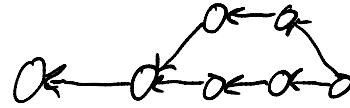


Remember Git's Model

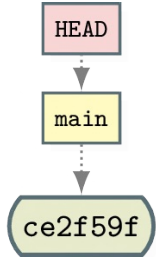


- A repository is a directed acyclic graph of commits
- Every commit except the initial commit has at least one parent
- Every commit is identified by a hash (a4b293...)
- A branch is a pointer to a commit (a stored hash)
- The current files that exist in an editable state in a repository form the working tree

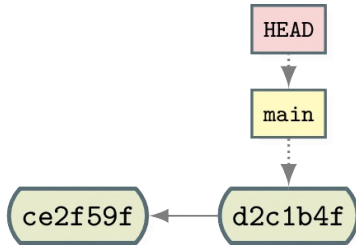
- Checking out a branch or commit changes the working tree
- `git add` puts changes to files in the working tree into the staging area
- `git commit` records staged changes as a new commit
- HEAD is a pointer to the currently checked out commit
- `git merge` merges two or more branches
- `git pull` fetches commits from a remote branch that don't exist locally and merges them into the current branch

- git pushes local commits to a remote and performs a merge on the remote

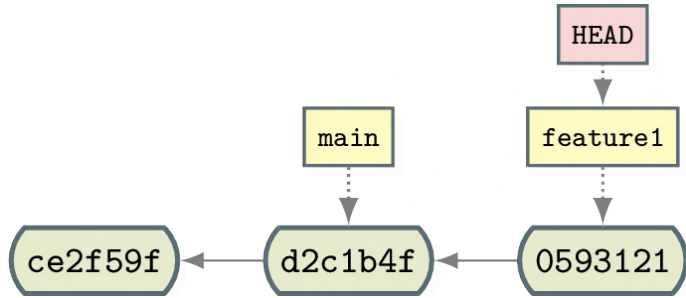
One commit, one branch, both main and HEAD point to the
one commit



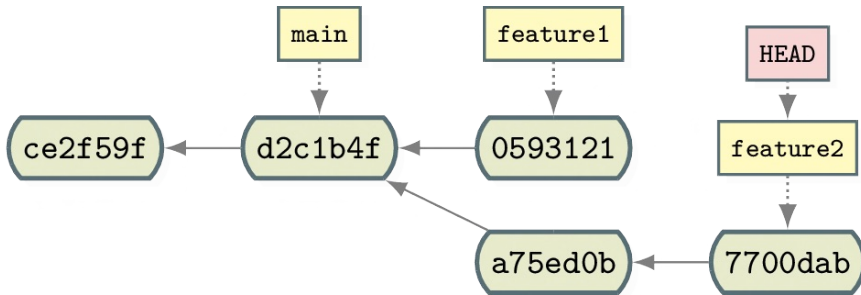
Added a second commit



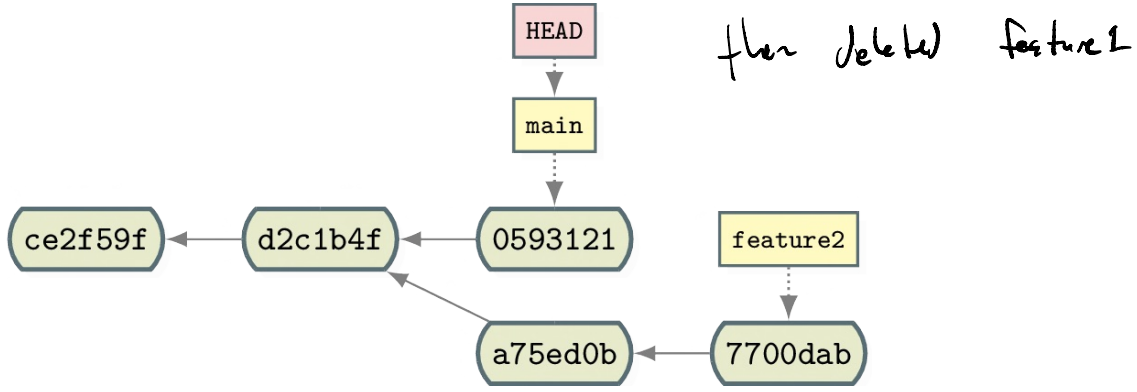
(checked out new branch feature1 and added a commit



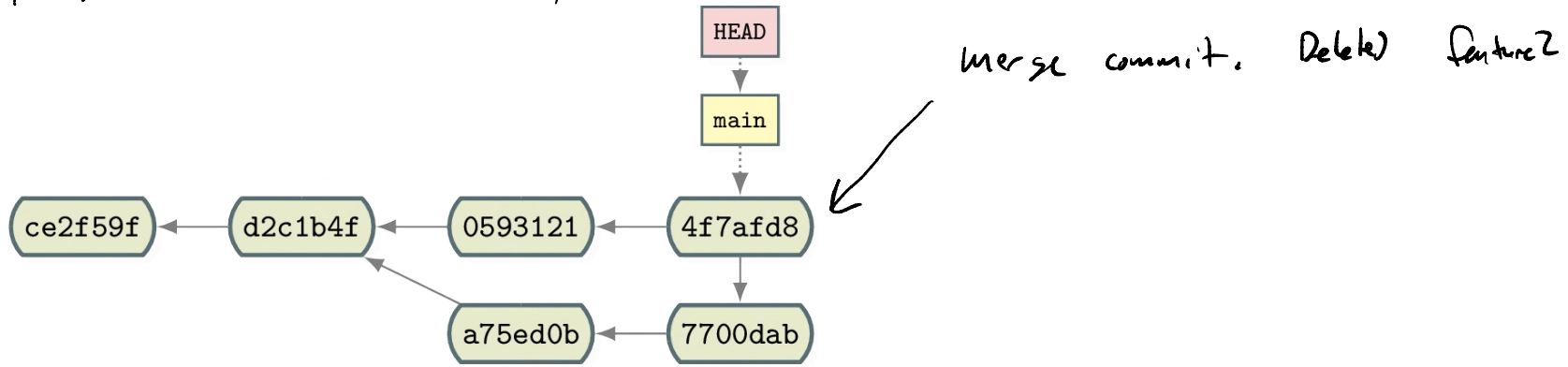
Made branch feature2 from main and added 2 commits



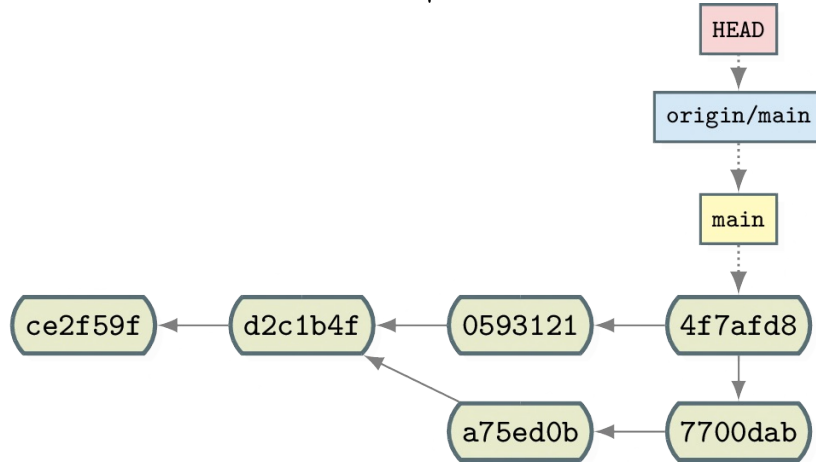
Merged feature1 into main (fast-forward merge, simply updated the main pointer)



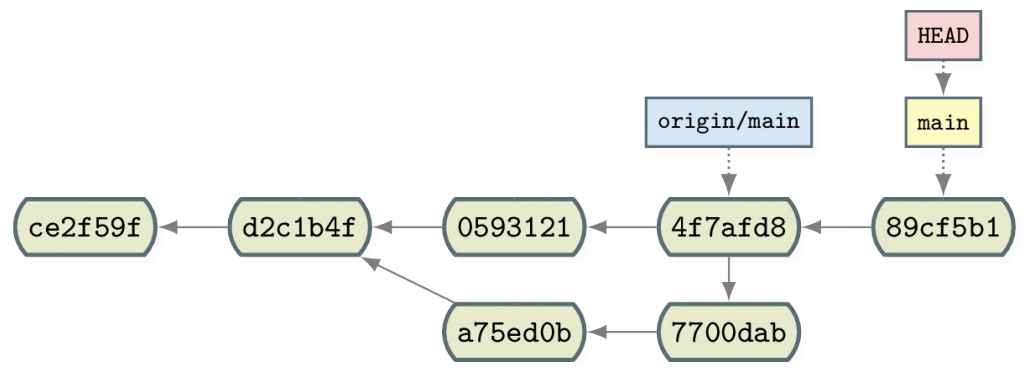
Merged feature2 into main, which required resolving conflicts resulting in a merge commit. Deleted feature2



Made a Github repo, added a remote that points to it named origin, and pushed to the main branch on origin

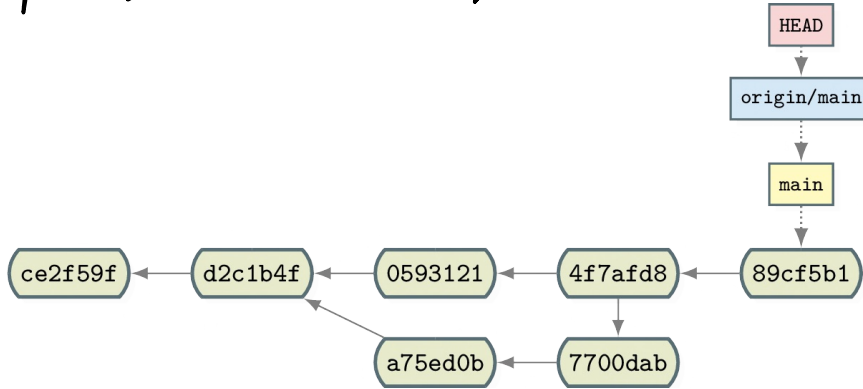


and pushed to the main branch on origin



Committed to main locally

Pushed to origin main



Forked on GitHub
Added remote "fork"
Created branch feature3
Pushed to fork feature3

