Intro to Git & Github

Part II: Branching and Merging

UChicago TechTeam

October 17, 2016

Review

Git Review

Local repository tracks changes to files on your computer

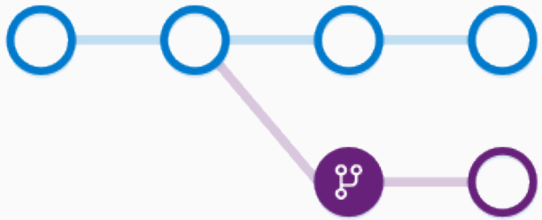
- `git add` *filename* adds the file to the staging area
- `git commit -m` "[*commit message*]" stores the changes as a new version in the repository

Remote repository is hosted on Github, allows collaboration

- `git push` to send files/changes from local repository to remote repository
- `git pull` to update local repository with files/changes from remote repository

Branching

Branching in Git



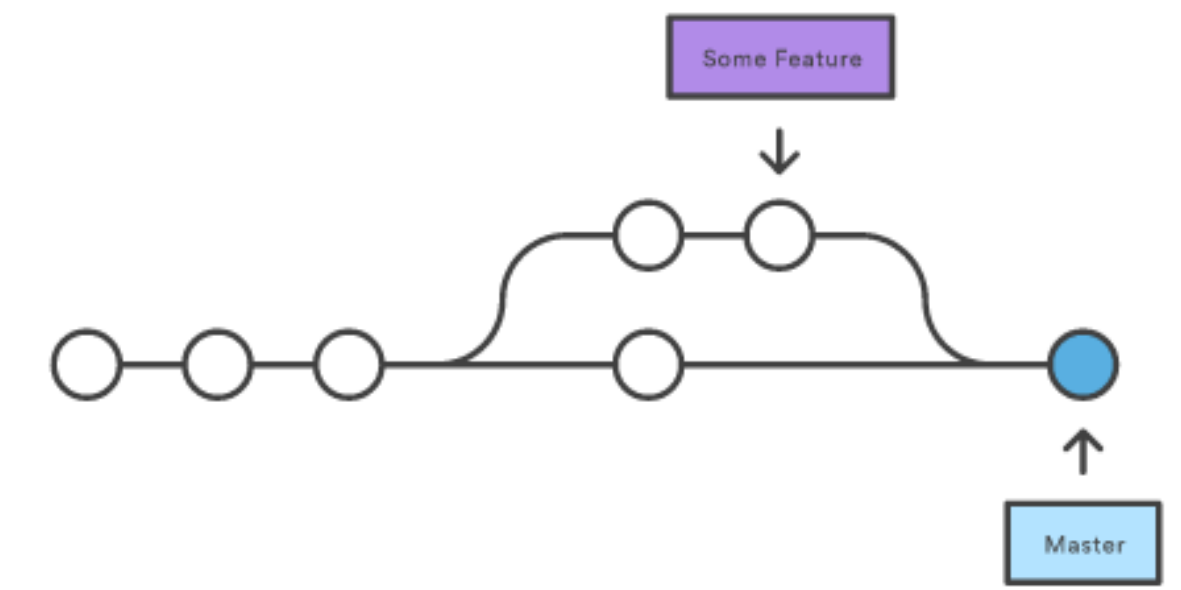
- Create a new branch to work on a particular feature of a project

  The original branch is known as `master`

- `git branch` *branch-name* to create a new branch, will create copies of all files
- `git checkout` *branch-name* to switch to that branch
- View all the branches in your repository with just `git branch`

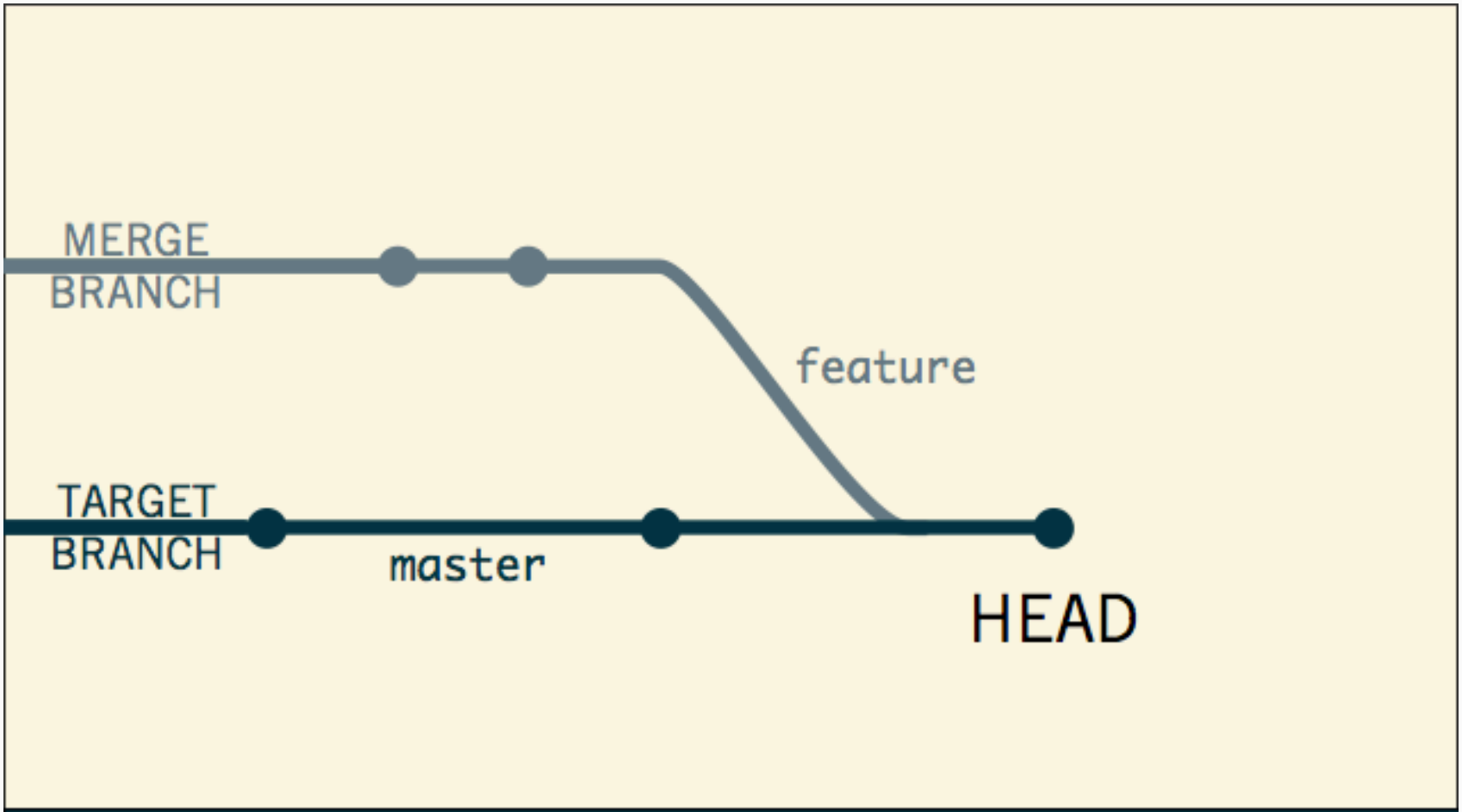
Merging

Merging branches



- The goal of branching is to eventually incorporate the features implemented in a branch back into the main branch
- Merge branches with the master branch
- `git merge` *branch-name* merges the specified branch into the current branch

Merging branches




```
(feature) $  git checkout master
 (master) $  git merge feature
```

Merge conflicts :(

- Auto-merging sometimes fails — if both branches have edits on the same part of a file, Git will not know which version to use
  - Open up the file(s) with conflicts and manually resolve
  - Add files and commit changes
- After successfully merging, the non-target branch will still exist

  Safe to remove the branch with `git branch -d` *branch-name*

Questions?