CS3110 Level Up!
Session 3: Advanced Git

Anna Fang

What we learned last time:

git pull: pull files from remote repository into local repository

git add: add file(s) and changes to the staging area

git commit: take snapshot of staging area and update remote repository

git push: upload local repository content to remote repository

git reset: unstage file(s)

git reset HEAD~<num>: undo <num> commits earlier (pretend like it never happened)

git revert: undo an earlier commit, but keep record of it

git checkout: change a file to the version on remote repository

git status: show state of working directory and staging area

git log: show commit history

What we're learning today:

(How to prevent and fix) Merge Conflicts

Branching
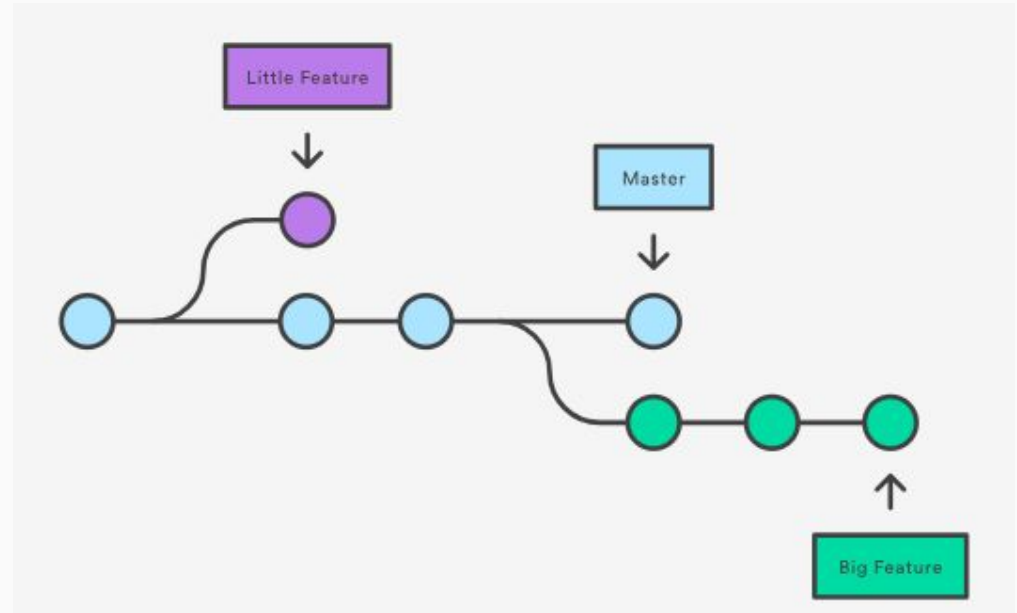
Cherry Picking

Rebasing

Pull Requests

Branching

Branching

Branch:

- an independent piece of development
- a new pointer to a commit
- like a tree branch or a fork in the road

Merge Conflicts

Merge Conflicts

Your changes conflict with another member's changes

A merge conflict will look like:

```
$ git merge iss53
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.
```

```
<<<<<<< HEAD
current branch code
=======
your merging branch code
>>>>>>> new_branch_to_merge_later
```

Directly fix the conflicts, then continue merging!

Merge conflict example

git branch
- list all branches in the repository

git branch newBranch
- creates new branch called newBranch

git merge newBranch
- merge newBranch into receiving branch

git branch -d newBranch
- deletes a branch whose changes have been merged

git branch -D newBranch
- deletes a branch regardless of if it has been merged

git checkout targetBranch
- switches to targetBranch
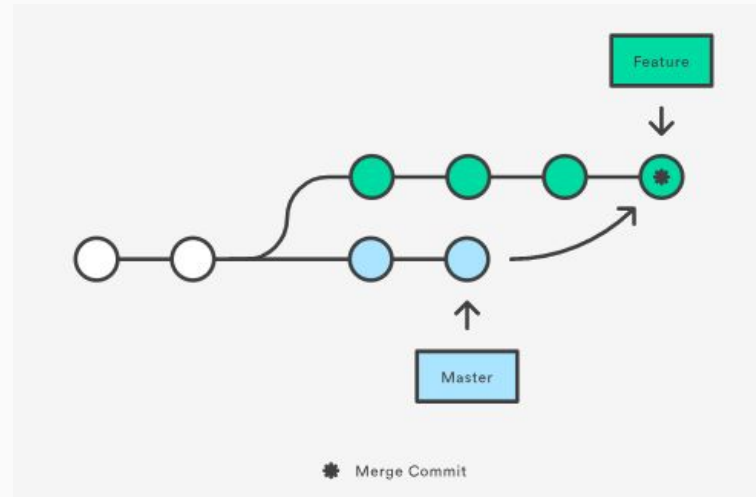- i.e. switch between your lines of development

Merging

git merge

Choose a branch and integrate into another

Combines commits into one history

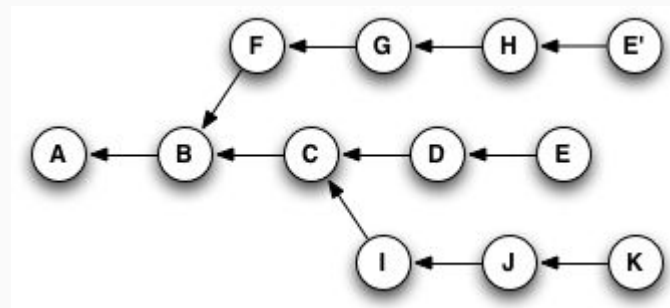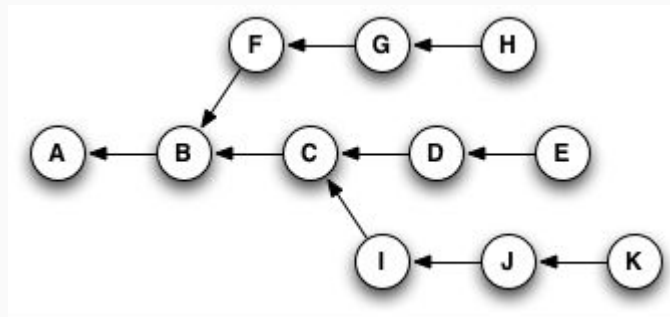

Feature

Master

✹ Merge Commit

Cherry Picking

git cherry-pick

Choose a commit from one branch
Apply it to the branch you're on

i.e. you want to push a commit somewhere in your
log to master

example:

git checkout master
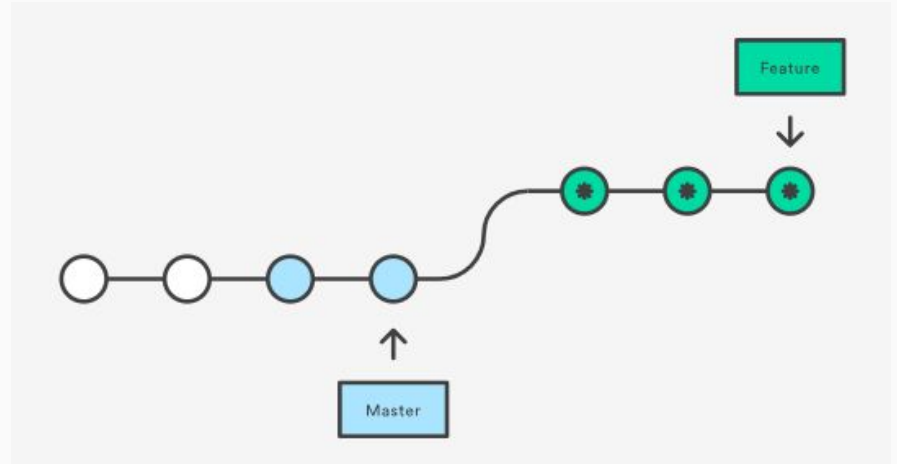git cherry-pick <commit-hash>

Rebasing

git rebase

Effectively the same as git merge, but destructive
Results in a cleaner project history
Creates a linear commit history

```
git checkout feature
git rebase master
```
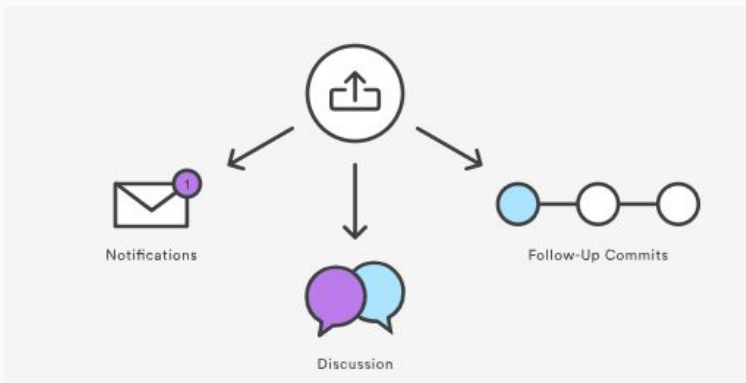
Pull Requests

Pull Requests

i.e. @ my members, please pull my changes!

Allows discussion and comments on your code changes

Notifications

Discussion

Follow-Up Commits

Next week:
Bash Programming