Lecture 5
Remotes

Sign in on the attendance sheet!
Announcements

• Midterm in-class next week (2/21)
  • Git commands we have learned: init, clone, log, show, diff, checkout, reset, branch, merge, fetch, pull, push, remote
  • Mental model for how git works: the working directory, staging area (index), commit log, diffs, branches (local and remote)

• Homework submissions will be through Autolab

• Thanks for filling out the survey!
  • More practical questions on the homeworks
  • More live examples during class
  • Keep doing the group activities at the end of class
Today

- Remotes
- `git remote`
- `git fetch`
- `git pull`
- `git push`
- Github
Last Time: Branches

- 8f4c26: Alice: begin work on feature A
- 6f96cf3: Alice: more work on feature A
- 8fc42c6: Alice: begin work on feature A
- b5f3729: Alice: even more work on feature A
- db82ca7: Merge branch ‘featureA’ into master
- e167179: Bob: more work on feature B
- 8277e09: Bob: even more work on feature B
- 8b7d883: Bob: begin work on feature B
- b4e2c29: initial commit
- master, HEAD

branch

- featureA
- featureB
Scenario: Alice, Bob, and Charlie collaborate on a project, but each on their own computer.

- Alice: 8fc42c6: heapchecker no longer infloops
  - b4e2c29: initial commit

- Bob: 8fc42c6: heapchecker no longer infloops
  - b4e2c29: initial commit

- Charlie: 8fc42c6: heapchecker no longer infloops
  - b4e2c29: initial commit
Scenario: Alice, Bob, and Charlie collaborate on a project, but each on their own computer.
Scenario: Alice, Bob, and Charlie collaborate on a project, but each on their own computer

How does Alice get commit 6f96cf3 to Bob and Charlie?
Scenario: Alice, Bob, and Charlie collaborate on a project, but each on their own computer

How does Alice get commit 6f96cf3 to Bob and Charlie?

Idea 1: Bob and Charlie each **fetch** the commit from Alice.
Idea 2: Alice pushes the commit to Bob and Charlie.
Fetching

1. Tell git to set up Alice’s repository as a “remote repository” or a “remote”. This only happens once.
2. Tell git to download the commits from the remote repository.
git remote add <remotename> <remoteurl>

Example use:
git remote add origin aperley@unix.andrew.cmu.edu:~/mygitrepo

• Adds a remote repository called “origin” located at aperley@unix.andrew.cmu.edu:~/mygitrepo (through SSH)
• “origin” is the default name for a remote, since often times the first remote you have is the one you clone from
• If you created the repository using git clone (rather than git init), the repository you cloned from is called “origin”
git remote -v

• Verbosely lists all your remotes:

```
$ git remote -v
origin https://github.com/autolab/Autolab.git (fetch)
origin https://github.com/autolab/Autolab.git (push)
```

• Each remote has a fetch and push url. They are almost always the same.
Git Remote URLs

• 4 choices: HTTPS, SSH, local, git

• HTTPS: git objects and commits are transferred over web protocols. Pretty easy to set up, unless you need authentication.

• SSH: git objects and commits are transferred over the SSH protocol. Requires a ssh daemon to be running, allows authentication.
git fetch <remotename>

Example use:

```
git fetch origin
```

- **Downloads** and **updates** all branches published by the remote
- Stores these branches as `<remotename>/<branchname>`
- Does NOT affect your own branches, like `master`!
Listing Remote Branches

- `git branch -r` (only remote) or `git branch -a` (all branches)
How do you actually bring in the remote changes?

Alice

- b4e2c29: initial commit
- 8fc42c6: heapchecker no longer infloops
- 6f96cf3: don’t create headers in free blocks

Bob

- b4e2c29: initial commit
- 8fc42c6: heapchecker no longer infloops
- 6f96cf3: don’t create headers in free blocks
How do you actually *merge* in the remote changes?

Scenario: Working on the Autolab Project, want to update my master branch to origin/master.
How do you actually *merge* in the remote changes?

Step 1: `git fetch origin`
How do you actually *merge* in the remote changes?

Git knows that master and origin/master are associated (more on this later), and that master can be fast-forwarded up to origin/master.
How do you actually *merge* in the remote changes?

$ git log --graph --oneline \ --decorate --branches \ --remotes
How do you actually *merge* in the remote changes?

Step 2:
```
git merge origin/master
```
Before Merge:

- 564bece (Merge/updated Mac installation) Update README.md
- 317b0be Update README.md
- d42cf70 Create start.sh
- 301d241a Update README.md
- 4bb0e66 Update README.md
- 6392d9f Update README.md
- 7e627a8 Update README.md

- 7e627a8 Rename files to include assignment name on download. Fixes #693. (labels: issue: #693)
- 3d32385 Fix issue #757: CUD creation now explains problem (#769)
- 891a4e6 Issue #757: Removed unnecessary comments.
- 4342d5b Issue #757: CUD creation now explains problem.
- e6b9312 Fixed gem dependencies in setup-osx script.
- 1fe4564 Finished OS shell setup script.
- 940f5 Command to go to the DB template setup for sqlite.
- c4f0287 Added primitive OS setup script.

- 4617679 (original/annotations_page_instructions) Updated message to properly inform how.
- 4a490c5 Attachments#show should use an absolute path (#751)
- 7157693 Non exist course 404 error (#768)
- 7157693 Rename files to include assignment name on download. Fixes #693. (labels: issue: #693)
- 3d32385 Fix issue #757: CUD creation now explains problem (#769)

- e4342d5b Issue #757: CUD creation now explains problem.
- 3d32385 Fix issue #757: CUD creation now explains problem.
- e6b9312 Fixed gem dependencies in setup-osx script.
- 1fe4564 Finished OS shell setup script.
- 940f5 Command to go to the DB template setup for sqlite.
- c4f0287 Added primitive OS setup script.

- 4617679 (original/annotations_page_instructions) Updated message to properly inform how.
- 4a490c5 Attachments#show should use an absolute path (#751)
- 7157693 Non exist course 404 error (#768)

After Merge:

- 564bece (Merge/updated Mac installation) Update README.md
- 317b0be Update README.md
- d42cf70 Create start.sh
- 301d241a Update README.md
- 4bb0e66 Update README.md
- 6392d9f Update README.md
- 7e627a8 Update README.md

- 7e627a8 Rename files to include assignment name on download. Fixes #693. (labels: issue: #693)
- 3d32385 Fix issue #757: CUD creation now explains problem (#769)
- 891a4e6 Issue #757: Removed unnecessary comments.
- 4342d5b Issue #757: CUD creation now explains problem.
- e6b9312 Fixed gem dependencies in setup-osx script.
- 1fe4564 Finished OS shell setup script.
- 940f5 Command to go to the DB template setup for sqlite.
- c4f0287 Added primitive OS setup script.

- 4617679 (original/annotations_page_instructions) Updated message to properly inform how.
- 4a490c5 Attachments#show should use an absolute path (#751)
- 7157693 Non exist course 404 error (#768)

- e4342d5b Issue #757: CUD creation now explains problem.
- 3d32385 Fix issue #757: CUD creation now explains problem.
- e6b9312 Fixed gem dependencies in setup-osx script.
- 1fe4564 Finished OS shell setup script.
- 940f5 Command to go to the DB template setup for sqlite.
- c4f0287 Added primitive OS setup script.

- 4617679 (original/annotations_page_instructions) Updated message to properly inform how.
- 4a490c5 Attachments#show should use an absolute path (#751)
git pull <remotename>

Example use:
git pull origin

• Runs git fetch <remotename>, then git merge <remotename>/<currentbranch>

• Ex: runs git fetch origin, then git merge origin/master
How do you checkout a branch on a remote repository?

Scenario: Working on the Autolab Project, want to take a look at a feature someone is developing on a separate branch.
How do you checkout a branch on a remote repository?

Step 1: Figure out the branch name

```bash
$ git branch --list -r *gradebook*
origin/gradebook_changes
origin/gradebook_tooltip_fix
```
How do you checkout a branch on a remote repository?

Step 2: Create a local branch pointing to the same place as the remote branch

```
$ git branch gradebook_changes origin/gradebook_changes
Branch gradebook_changes set up to track remote branch gradebook_changes from origin.
```

(Aaron@HELIOS ~)/Dropbox/Dropbox Documents/98174/Autolab (master)
How do you checkout a branch on a remote repository?

Step 3: Checkout the local branch you just created
git push <remotename> <branchname>

Example use:
git push origin master

- Updates the remote-tracking branch (<remotename>/<branchname>) and uploads the necessary commits for that branch to origin
- You must be fully up-to-date with the remote in order to push! Otherwise your push will be rejected.
What happens if you run `git push`?

- pushes current branch to origin (default remote) **git2.0 default**
- pushes all branches to origin (default remote) **pre-git2.0 default**

`git config --global push.default simple`
Summary

• Configuring remotes:
  • git remote [-v] – lists remotes [verbosely]
  • git remote add <remotename> <remoteurl> - configure a new remote
  • git branch –r or –a – lists branches including remote tracking

• Fetching:
  • git fetch <remotename> - downloads updates to all remote-tracking branches to match the remote
  • git pull <remotename> - runs `git fetch`, then merges in updates to the current branch

• Pushing:
  • git push <remotename> <branchname> - uploads changes in your branches to the remote
Activity!

Clone:
YOUR_ANDREW_ID@unix.andrew.cmu.edu:/afs/andrew.cmu.edu/course/98/174/public/lecture5-practice

Create a branch named YOUR_ANDREW_ID
And make a commit to it

Push the branch up to origin