I. Git Introduction

Codeschool Notes

II. Commits and Resets

III. Cloning And Branching

A. Collaborating Workflow

- 1. Greg pushes his local repo to github
- 2. Jane finds the repo url on github
- 3. then she clones it

B. How to Clone

- 1. **git clone https://github.com/username/repositoryname.git**
  - a) this clones the repository to **repositoryname.git**
- 2. **git clone https://github.com/username/repostioryname.git some-other-name**
  - a) this gives the repository a name of **some-other-name** when you download it
- 3. what does clone do?
  - a) downloads the entire repository into a new directory
  - b) adds the '**origin**' remote, pointing it to the clone URL
    - (1) (remember **git remote -v** lists all the remotes we have)
  - c) check out the initial branch (usually **master**)

C. Branching

- 1. **git branch**
  - a) lists all branches that currently exist
  - b) also tells you what branch you are on using a * in front of the name
- 2. **git branch some-name**
  - a) creates a branch called **some-name** that is a copy of THE BRANCH YOU ARE ON
  - b) does NOT automatically put you on the new branch
- 3. to get to the new branch use **git checkout some-name**
  - a) this switches you to the new branch

D. Merging

- 1. you've done work on your branch, but now you want to merge it back into the main
- 2. **git checkout master** (or whatever branch you want to merge TO)

- 3. **git merge some-name**
  - a) this merges **some-name** INTO **master**
  - b) you merge INTO the branch you are CURRENTLY ON
- 4. FAST FORWARD MERGE
  - a) if you did nothing at all from the time you created the branch
    - (1) no problems
    - (2) this is called a FAST FORWARD merge (REMEMBER THIS)
    - (3) **git branch -d some-name**
      - i) deletes branch
- 5. **RECURSIVE** MERGE
  - a) how does this happen?
    - (1) you create a new branch ie, **experimental**
    - (2) then you do some work on the branch, you change/add files etc
    - (3) then you switch to the master and do work on that
    - (4) then you switch back to experimental and do more work
    - (5) THEN you merge, and suddenly you are in a text editor
  - b) if everything is cool
    - (1) that is, nothing you wrote in the **experimental** branch conflicts with the master
    - (2) then all you have to do is sign off on an edit message
    - (3) the message will say “please enter a message saying why you are merging blah blah”
  - c) the log will then say “at this point, the master and the experimental became one branch”

IV. Fixing Upstream Merges