Version Control with Git

This assignment assumes you know how to use git on your computer locally

- Go to http://github.com and sign up for an account
- Create a new github repository using the + in the upper righthand corner
- Select a name (it doesn't matter for the assignment, but usually you will pick something informative)
- In your Terminal (Mac / Linux) or git-bash (Windows) go to your local repository
- Connect your repository to your "remote" github repository (conventionally termed origin)

  `git remote add origin https://github.com/rachelss/curly-enigma.git`
- Check that "origin" indicates the remote repo

  `git remote -v`
- Push your local "master" repository to github (i.e. duplicate the entire repository including all the commit history to your remote "origin" site)

  `git push origin master`
- Make a change to your repository, save it, and commit the change

Work in pairs for the rest of the assignment

- Assign one partner to be person #1 and one as #2
- (1) in your github repository click the settings button and add #2 as a collaborator
- (2) got to github notifications or check your email to accept the collaboration
- (2) go to the collaborative repository, click on the clone/download button, click the clipboard to copy the repo link
- (2) copy the repo to your computer

  `git clone https://github.com/rachelss/curly-enigma.git`
- (2) make a change to the repo and push it
- (1) get #2's change

  `git pull origin master`

Both collaborators now have identical copies of #1's repo

Challenge 1

Using #1's repo collaboratively

- (2) make a change to the first line of a file and push it

- (1) make a change to the last line
- (1) pull the repo (the changes should be merged automatically)
- (1) push the repo
- (2) pull the repo

Both collaborators now have identical copies of #1's repo

Challenge 2

Using #1's repo collaboratively

- (1) make a change to the first line of the file
- (1) push the repo
- (2) make a change to the first line of the file
- (2) push the repo (this should result in a conflict)
- (2) pull the repo
- (2) manually fix the conflicts (both versions of the conflicting lines are indicated in the file)
- (2) push the repo
- (1) pull the repo

Both collaborators now have identical copies of #1's repo

Challenge 3

Using #2's repo NOT collaboratively (so you don't have to argue about how you resolve merge conflicts)

- (1) go to #2's repo
- (1) click Fork - this gives you your own copy of #2s repo on github
- (1) clone your fork of the repo
- (1) make a change and push the repo
- (1) submit a pull request to get #2 to add your changes to their version of the repo
- (2) review and accept the pull request
- (2) pull your repo
- (2) make a change to the repo and push it
- (1) get changes to the original repo locally and in github

```
git remote add upstream https://github.com/rachelss/curly-enigma.git
git pull upstream master
git push origin master
```

Both collaborators now have identical copies of #2's repo locally and on github