How to Safely Clone (people)

Kyle Helmick - kyle.helmick@colorado.edu - Spring 2018

The goal is to be organized without letting your classmates copy all of your answers. Sometimes with GitHub repositories things can get rough when forking a repo and then trying to make it private (you can't do that). However I'm writing this to tell you that you can, and it *is* possible. What we'll aim to do is create a private repository that has two "master" branches, a public one (Dan's in this case) and a private one (the head of your own private repository). This will allow you to pull new assignments and content from Dan's GitHub repository, and push edited files and completed assignments to your private repository. This is also done in a way that Dan isn't notified (or asked for permission) for every push, pull, and commit you make.

The Tutorial

First, create a private repo on GitHub (this is given that you have signed up for git with your student email and got all the goodies like free private repos). For demonstration purposes we will name this repo **csci3022-Lastname**.

Now you'll want to open a terminal (or something that does git) and clone the *public* repository. As a reminder this repository is: https://github.com/dblarremore/csci3022

```
git clone https://github.com/dblarremore/csci3022
```

Next you'll want to change directories into the newly created folder **csci3022** and execute the command:

```
git push --mirror https://github.com/yourusername/csci3022-Lastname
```

At this point you've created a private repo and *mirrored* Dan's public repo into it. You can then remove the clone of Dan's repository because you don't really need his public version of the repository on your computer.

```
cd .. && rm -rf csci3022
```

This time you'll want to clone the *private* repository. As a reminder this repository is: https://github.com/yourusername/csci3022-Lastname

```
git clone https://github.com/yourusername/csci3022-Lastname
```

To make syncing with Dan easy you'll also have to run:

```
git remote add public https://GitHub.com/dblarremore/csci3022
```

This is kind of an alias for his "branch" so that you can pull his stuff with the keyword **public**.

The Part Where You Make Changes

Right here you want to make some changes to the git repo, you could be doing your homework, editing the syllabus, photoshopping Dan's face onto corgis... Anything you want!

Saving Your Changes

First, it's good to know which files you changed while you were frantically working on the practicum minutes before it's due. You can see all "staged" files (files that are part of the current commit) and all changed files with:

```
git status
```

Just like a normal repo you'll want to add files:

```
git add dancorgimix.jpeg chrispoodlemashup.png
```

or every changed file:

```
git add .
```

to "the stage" and then commit:

```
git commit -m "We are not Knaves"
```

the files with a message to log your changes. You will want to make sure to

```
git push
```

to save your edits on GitHub's servers.

Getting Dan's Changes

So now Dan has posted homework/Ralphie stories and you really want them in your private repo. These are the steps to take to get those sweet sweet problems sets:

1. Make sure to save any current private changes with:

   ```
   git commit -m "He is a Knave"
   ```

   and

   ```
   git push
   ```

2. Copy Dan's new public files with:

   ```
   git pull public master
   ```

3. Then

   ```
   git push
   ```

   Dan's new public content into your private repository

One of the quirks about this, is that every time you pull from Dan's repository you will create what's called a *merge commit* so it's recommended to push Dan's new public content (which includes the *merge commit*) to your private repository before editing things.

I would also recommend more descriptive commit messages because they are more useful than you think!

Bonus: which commit message is a knight and which is a knave?

The End.