NE 24 - Homework #2
Chris Lamb
13 February, 2017

Section 1: Creating a Repository

1. Dracula starts a new project, *moons*, related to his *planets* project. Despite Wolfwomans concerns, he enters the following sequence of commands to create one Git repository inside another:

```
cd # return to home directory
mkdir planets # make a new directory planets
cd planets # go into planets
git init # make the planets directory a Git repository
mkdir moons # make a sub-directory planets/moons
cd moons # go into planets/moons
git init # make the moons sub-directory a Git repository
```

Why is it a bad idea to do this? How can Dracula "undo" his last *git init*?

Answer: Its a bad idea to do this because when the repositories are nested inside of each other, they mess with each other and cause errors. To fix this error, you can remove the .git repository with rm rf moons/.git

Section 2: Tracking Changes

1. Which command(s) below would save the changes of *myfile.txt* to my local Git repository?

```
$ git commit -m "my recent changes"
```

```
$ git init myfile.txt
$ git commit -m "my recent changes"
```

```
$ git add myfile.txt
$ git commit -m "my recent changes"
```

```
$ git commit -m myfile.txt "my recent changes"
```

Answer: $ git add myfile.txt
$ git commit -m "my recent changes"

2. Create a new Git repository on your computer called *bio*. Write a three-line biography for yourself in a file called *me.txt*, commit your changes, then modify one line, add a fourth line, and display the differences between its updated state and its original state.

Answer: Here's a screenshot of the answer: http://i.imgur.com/xvGHjaE.png

Section 3: Exploring History

1. Jennifer has made changes to the Python script that she has been working on for weeks, and the modifications she made this morning broke the script and it no longer runs. She has spent ∼1hr trying to fix it, with no luck Luckily, she has been keeping track of her projects versions using Git! Which commands below will let her recover the last committed version of her Python script called *data_cruncher.py*?

   1. `$ git checkout HEAD`

   2. `$ git checkout HEAD data_cruncher.py`

   3. `$ git checkout HEAD~1 data_cruncher.py`

   4. `$ git checkout <unique ID of last commit> data_cruncher.py`

   5. `Both 2 & 4`

Answer: Both 2 & 4