Making Friends With the Terminal & Git

Stack Overflow

- A question and answer site for professional and enthusiast programmers.
- https://stackoverflow.com/

Goals

- Be able to navigate through folders from the command line.
- Exposure to common bash commands.
- Git and the most popular commands.
- Clone first repository.
- Push to first repository.

Command Line

Command-Line Interface

- Runs Commands
- Outputs Results

Flavors

- sh (Bourne shell)
- csh (C shell)
- zsh (Z shell - popular with web developers)
- bash (Bourne-again shell)

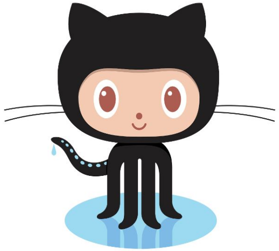
Frequently used Commands

- mkdir = create new folder.
    - mkdir example_folder.
- rm -r <foldername> = delete folder.
- ls <foldername> = list files in folder.
- ls = list files in folder.
- https://annawilliford.github.io/2016-04-02-UTA/workshop/Linux/bash_cheat_sheet.pdf

- cd <foldername> = change directory
- cd / = go to root
- cd .. = go up one folder
- cd ~=home directory
- pwd = print working directory

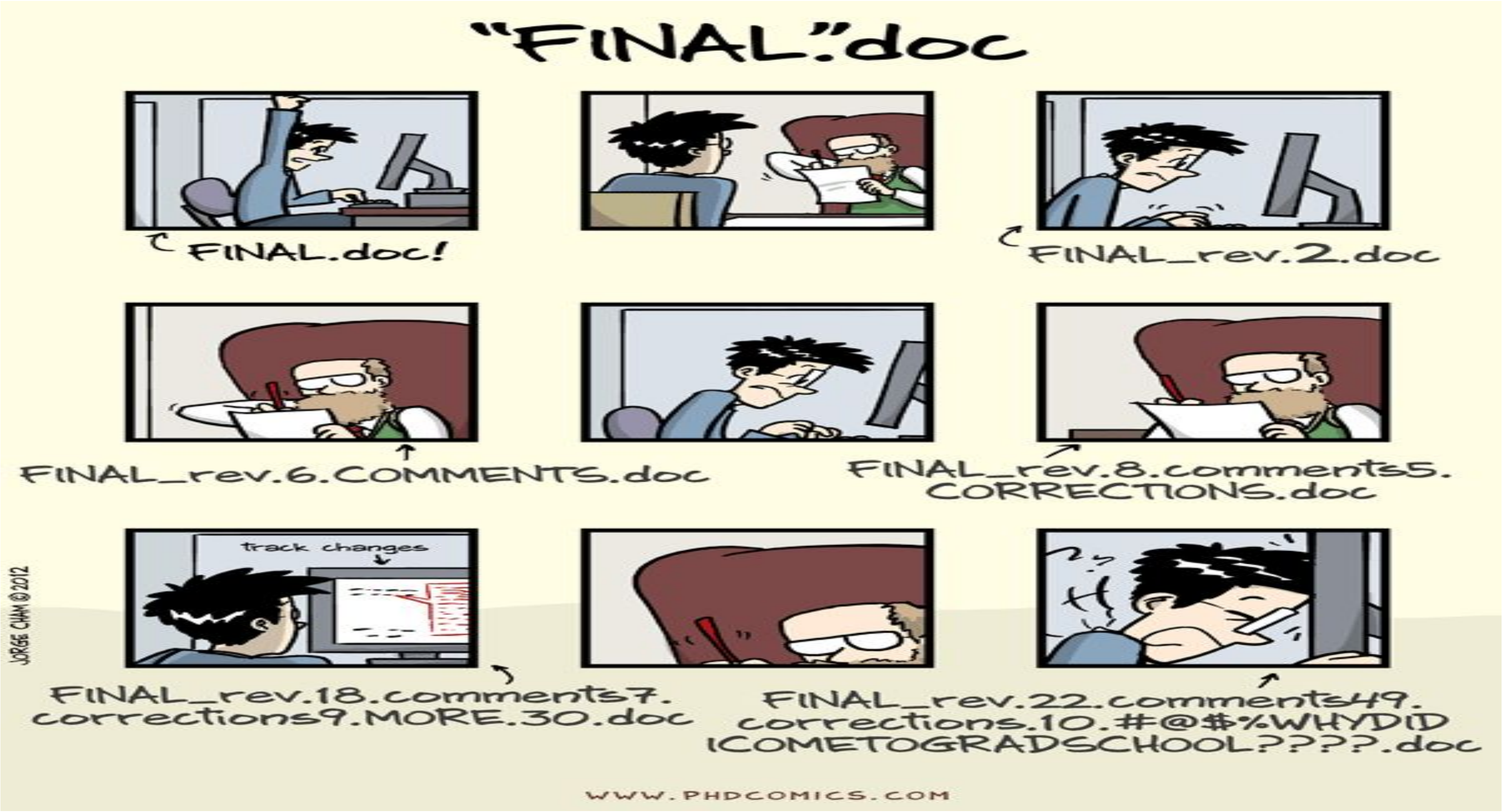
Why Learn GIT?

What is GIT?



GITHUB?

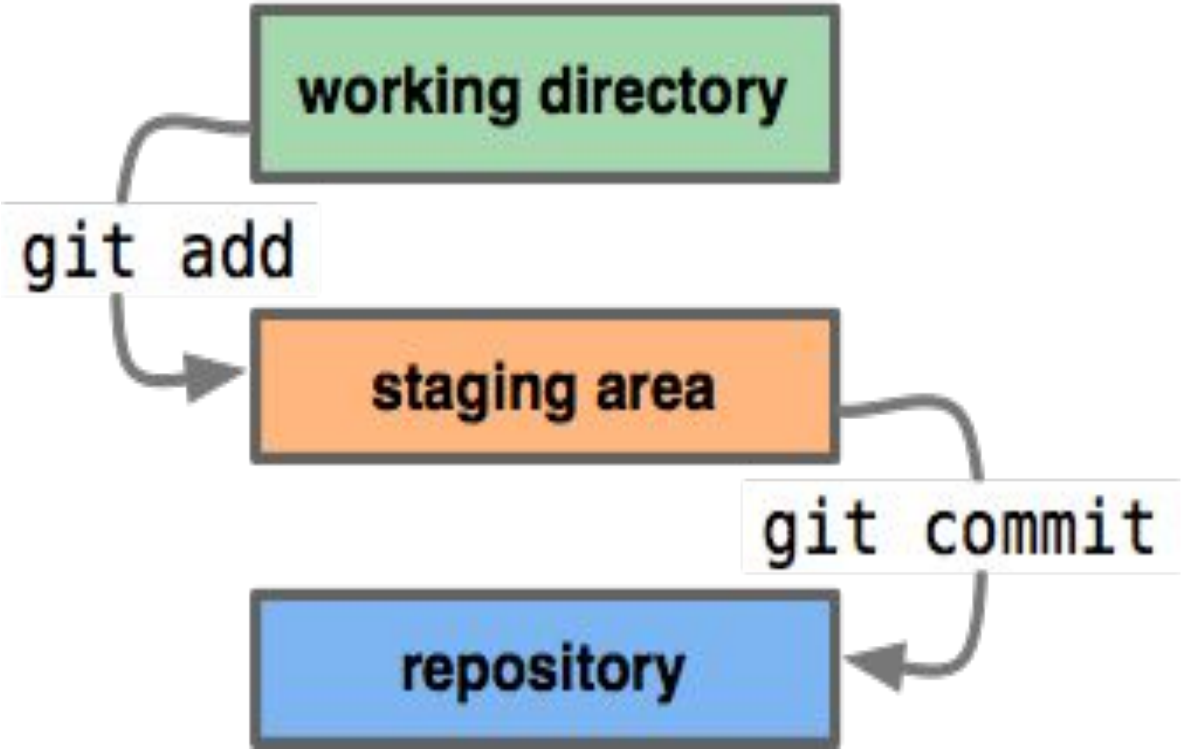
Why Github


"FINAL".doc
FINAL.doc!
FINAL_rev.2.doc
FINAL_rev.6.COMMENTS.doc
FINAL_rev.8.comments5.
CORRECTIONS.doc
track changes
JORGE CHAM © 2012
FINAL_rev.18.comments7.
corrections9.MORE.30.doc
FINAL_rev.22.comments49.
corrections.10.#@$%WHYDID
ICOMETOGRADSCHOOL????.doc
WWW.PHDCOMICS.COM

Git Terminology

- **Repository (repo):** A directory that contains your project work, as well as a few files used to communicate with Git. Either local or remote
- **Working Directory:** The files that you see in your computer's file system. When you open your project files up on a code editor, you're working with files in the Working Directory.
- **Staging Area:** A file in the Git directory that stores information about what will go into your next commit. You can think of the staging area as a prep table where Git will take the next commit.
- **Fork:** a repository from GitHub to create your own local copy.

Workflow



working directory

git add

staging area

git commit

repository

Git Commands

- Use **git clone** to clone a repository to your local computer.
- Use **git status** to see the status of your locally cloned git repository.
- Use **git add** to add your local changes to be committed.
- Use **git commit -m "Commit Message"** to commit changes that have been added with a message.
- Use **git push** to upload your local changes to GitHub.

Go to link below

- https://github.com/learn-co-students/ds-atlanta-09_16_19
- Fork a copy to your own Github account.
- Do not clone directly from my copy!

Clone from the copy you forked.

- Should be in the folder where you plan to keep your code or just cd ~.
- In your terminal type after you have copied from Github.
- **git clone** https://github.com/learn-co-students/ds-atlanta-09_16_19.git
- Type **ls** should see folder **ds-atlanta-09_16_19.**
- **cd** into **ds-atlanta-09_16_19.**

Let’s Explore More Commands

- Type **git status.**

```
On branch master

Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```

Creating Folder for update

- Type **mkdir** **module_1.**
- **cd** into **module_1.**

Let’s Create Something

- Type **touch emptyfile.txt**

- **echo** "I'm working with Git" **>> emptyfile.txt**

- **cat emptyfile.txt**

Let’s Check the status

- **git status**

  nothing added to commit but untracked files present (use "git add" to track)

Let’s Add

- git add emptyfile.txt

- git commit -m "this is a commit"

Let’s push to github

- git push

Cheat Sheets

- https://annawilliford.github.io/2016-04-02-UTA/workshop/Linux/bash_cheat_sheet.pdf
- http://try.github.io/
- https://github.github.com/training-kit/downloads/github-git-cheat-sheet.pdf

Class Exercise

- Fork to your own github a copy of a repository you find useful.
- Examine the files.