Getting Started with Version Control

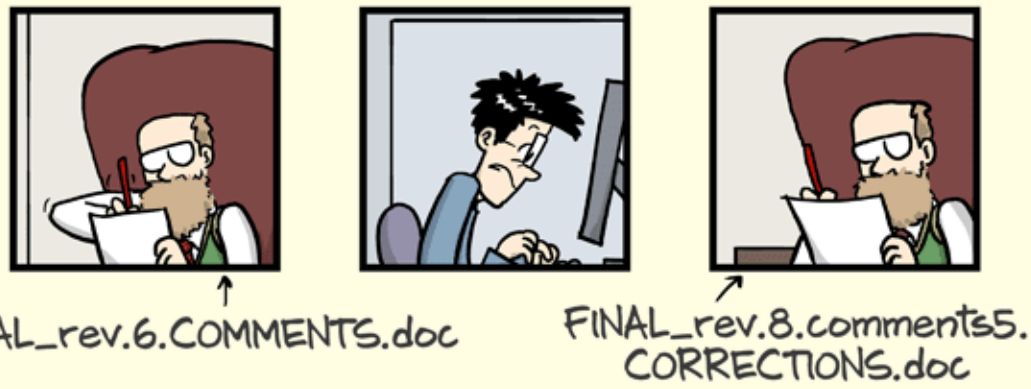
Lindsay Bassman,
Aravind Krishnamoorthy, Hiroyuki Kumazoe
University of Southern California

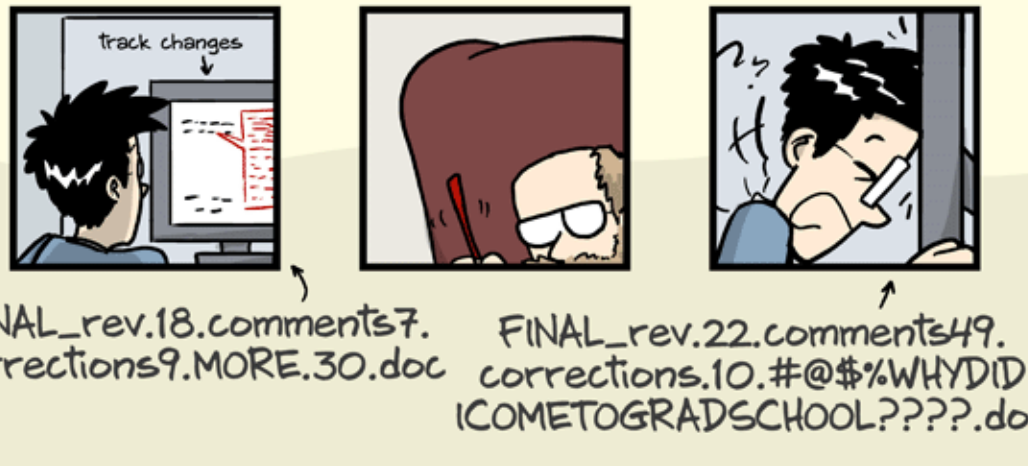
CS 699
Extreme-Scale Quantum Simulations
January 17, 2018
Presentation adapted from
http://swcarpentry.github.io/git-novice/

What is Version Control?

- Version control systems start with a base version of the document and then save just the changes you make at each step of the way.
- Allows multiple people to work on a document in parallel.
  - Automatically notifies users whenever there's a conflict between one person's work and another's.
  - Provides a record of who made what changes when.
  - Avoids having to email revised copies back and forth.
- Nothing that is committed to version control is ever lost.
- Acts as the ultimate 'undo' option, as you can always revert back to a previous version.
- It isn't just for software: books, papers, small data sets, and anything that changes over time or needs to be shared can and should be stored in a version control system.

"FINAL".doc





What You Need

- Bash shell
  - If your Windows machine does not natively have one, try http://gitforwindows.org/
- Git
  - Try typing ‘git’ into your shell and see if it needs to be downloaded
- GitHub account
  - You may create one for free at github.com
- Text Editor
  - nano is a very basic text editor but you may use whichever you like

This page nicely summarizes how to obtain all of the above:
https://swcarpentry.github.io/workshop-template/#git

Setting Up Git

When we use Git on a new computer for the first time, we need to configure a few things

```
$ git config --global user.name “Lindsay Bassman”
$ git config --global user.email “bassman@usc.edu”
```

*Use the email address with which you made your GitHub account/

If you ever need help you can type either:

```
$ git config -h
$ git config --help
```

Creating a Repository

First, create a directory for our work and then move into that directory:

```
$ mkdir CS699
$ cd CS699
```

Then we tell Git to make CS699 a repository—a place where Git can store versions of our files:

```
$ git init
```

Try
```
$ ls
```

Then try:
```
$ ls –a
```

Git stores information about the project in this special sub-directory '.git'. If we ever delete it, we will lose the project's history.

Tracking Changes

First, ensure you are in the ‘CS699’ repository:
$ pwd

We will create our first file using the nano editor (or editor of your choice):
$ nano helloworld.txt

Type “Hello World!” then save and exit.
Now check that your file has been saved:
$ ls

Check the status of your repository:
$ git status

The “untracked files” message means that there’s a file in the directory that Git isn’t keeping track of. We can tell Git to track a file using git add.
$ git add helloworld.txt

Check the status again: $ git status

Tracking Changes

Git now knows that it's supposed to keep track of helloworld.txt, but it hasn't recorded these changes as a commit yet. To get it to do that, we need to run one more command:

$ git commit -m "Greeting to the world added"

When we run git commit, Git takes everything we have told it to save by using git add and stores a copy permanently inside the special .gitdirectory. This permanent copy is called a commit (or revision).

We use the -m flag (for "message") to record a short, descriptive, and specific comment that will help us remember later on what we did and why. If we just run git commit without the -m option, Git will launch nano (or whatever other editor we configured as core.editor) so that we can write a longer message.

Try checking the status again:

$ git status

Tracking Changes

Now edit the text file helloworld.txt (add a line, change a word, etc.), save and exit

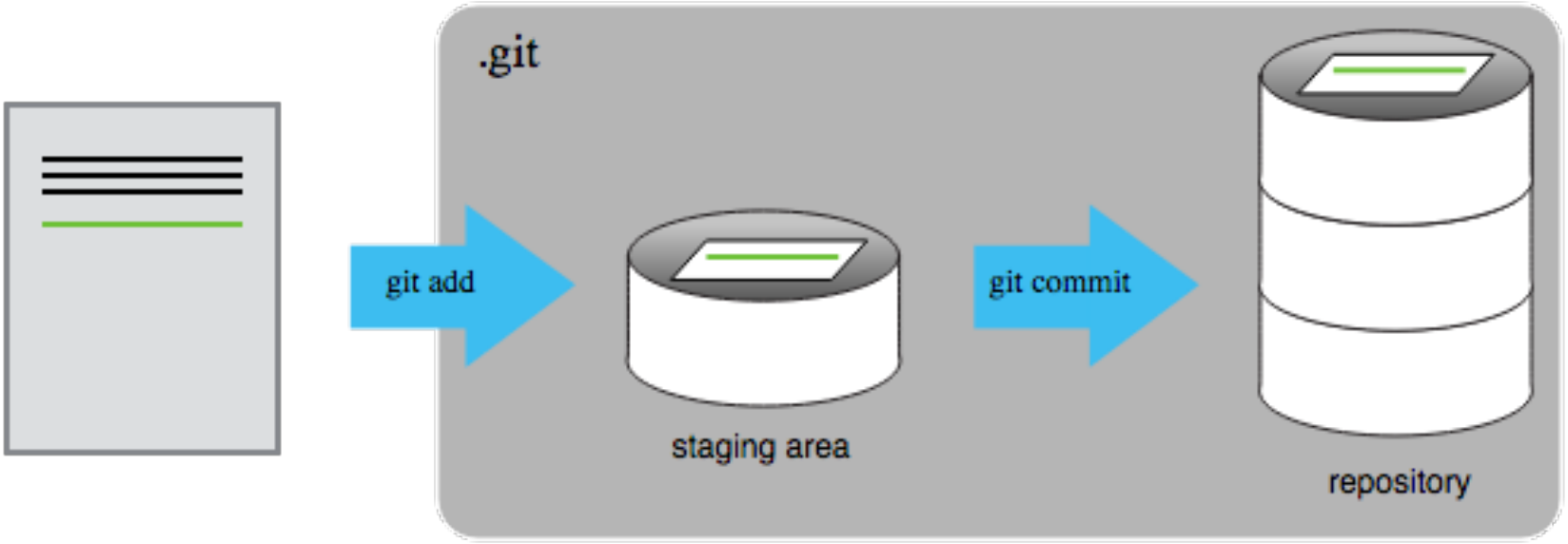
Check the status again:
$ git status

We have changed this file, but we haven't told Git we will want to save those changes (**git add**) nor have we saved them (**git commit**).

It is good practice to always review our changes before saving them:
$ git diff

After reviewing our change, it's time to add and commit it:
$ git add helloworld.txt
$ git commit -m "Edited my greeting to the world"

Tracking Changes



Remotes in GitHub

To clone this course's repository:
$ git clone https://github.com/USCCACS/QXMD_Course.git
$ git pull origin master

To learn more about making your own repository see:
http://swcarpentry.github.io/git-novice/07-github/