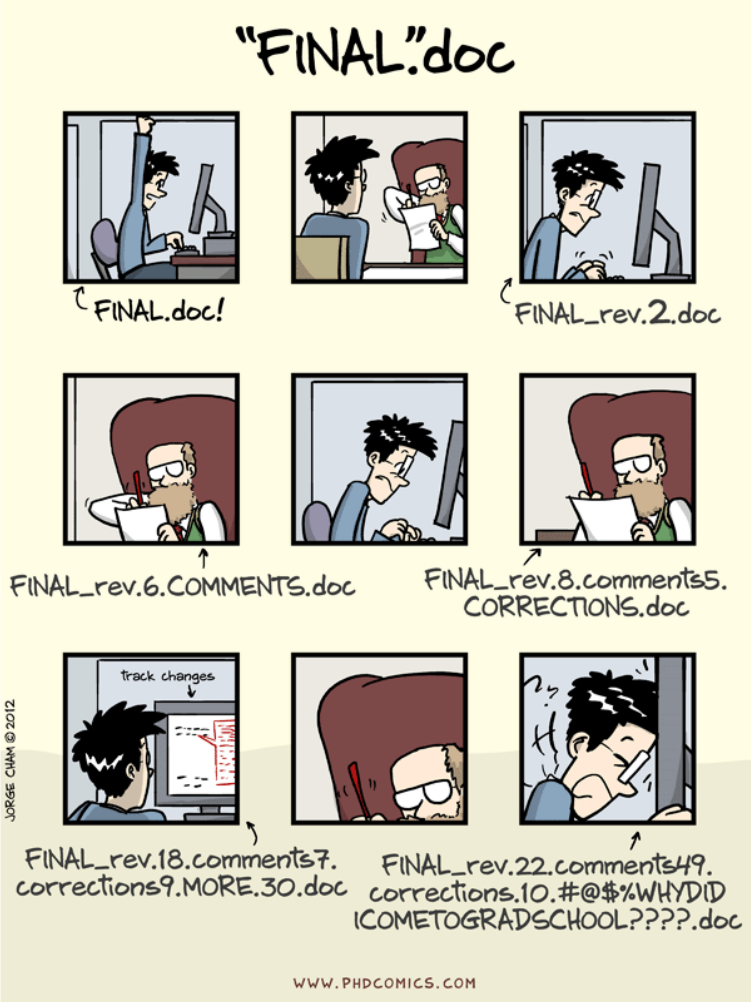
Introduction to Scientific Computing
Meeting 9
Version Control with git



Jeremiah Lant, Hydrologist
USGS Kentucky Water Science Center
jlant@usgs.gov

Today's Objective

1. **Understand what version control is.**
2. **Learn the basic git commands; git init, git add, git commit**
3. **Create your first Git repository on your computer**

- **Why?**
  - Good basis for learning how to program.
  - Becoming more comfortable on command line
  - Becoming more efficient

Review from last meeting: bash scripts

$ HELLO=“Hello World”
$ echo $HELLO

- Sample script that reads command line arguments and user input
  - readinput.sh

```
# read command line arguments from command line
ARG0=$0
ARG1=$1
ARG2=$2

echo "Argument 0 is: " $ARG0
echo "Argument 1 is: " $ARG1
echo "Argument 2 is: " $ARG2

# read user input from command line
echo -n "Enter your name and press [ENTER]: "
read name
echo $name
```

Review from last meeting: For Loop

- **for loop** - A loop that is executed once for each value in some kind of set, list, or range.

```
$ for filename in *.txt
> do
>         echo $filename
> done
```

- Note, the above is the same as the following:

```
$ for filename in *.txt; do echo $filename; done
```

Review from last meeting: For Loop

```
$ for filename in *.txt
> do
>         echo $filename
>         mv $filename process-$filename
> done
```

- Note, the above is the same as the following:

```
$ for filename in *.txt; do echo $filename; mv $filename process-$filename; done
```

Review from last meeting: peak.sh

- peak2.sh

```
for filename in *dv.txt
do
        echo $filename
        grep -v -e “#” -e “agency_cd” -e “5s” $filename | cut -f 4 | sort - n | uniq | tail -1
done
```

OR

- peak2.sh

```
for filename in *dv.txt
do
        echo $filename
        grep ^USGS $filename | cut -f 4 | sort - n | uniq | tail -1
done
```

Introduction to Version Control and Git

- Version Control – used **to keep track (a history)** of what you have done and **to collaborate** with others.
- From Software Carpentry:
  - **Version Control** - A tool for managing changes to a set of files. Each set of changes creates a new revision of the files; the version control system allows users to recover old revisions reliably, and helps manage conflicting changes made by different users.

- Git
  - Book – Pro Git by Scott Chacon
    - http://git-scm.com/book
  - Some Introductory Videos
    - http://git-scm.com/videos

Try out commands

- Create a Git repository called “myhobbies” and write 2 of your hobbies in a file called “hobbies.txt”. Commit the changes, then add another hobby to “hobbies.txt”, and display the differences between the updated and original file.

```
$ mkdir myhobbies
$ cd myhobbies
$ git init
$ ls -a
$ start notepad++ hobbies.txt
$ git status                                  # view current status of repo
$ git add hobbies.txt
$ git status
$ git commit –m “a list of my hobbies”
$ git status
$ git log                                     # view all commits
$ start notepad++ hobbies.txt
$ git diff
```

Try out commands

- Add a few more hobbies and commit changes

- Explore the history of hobbies.txt using **HEAD~** and **unique 40 character identifier**

$ git diff HEAD~1 hobbies.txt

$ git diff HEAD~2 hobbies.txt

$ git log

$ git diff *your-unique-40-character-identifier* hobbies.txt

- Recover old versions using **checkout**

$ git status

$ git log

$ git checkout *some-unique-40-character-identifier* hobbies.txt

$ cat hobbies.txt

$ git status

$ git add hobbies.txt

$ git commit –m “back to older version”

Summary of basic git commands

- **git init, git add, git commit**

$ git init *some-project-name* # initializes a project to be version controlled

$ cd *some-project-name*

OR

$ mkdir some-project-name

$ cd *some-project-name*

$ git init

$ git add . # add everything to “staging area”

$ git commit –m “*some-comment*” # commit (version) everything in “staging area”

Next meeting

- More with version control with Git
  - Collaborating
  - Conflicts