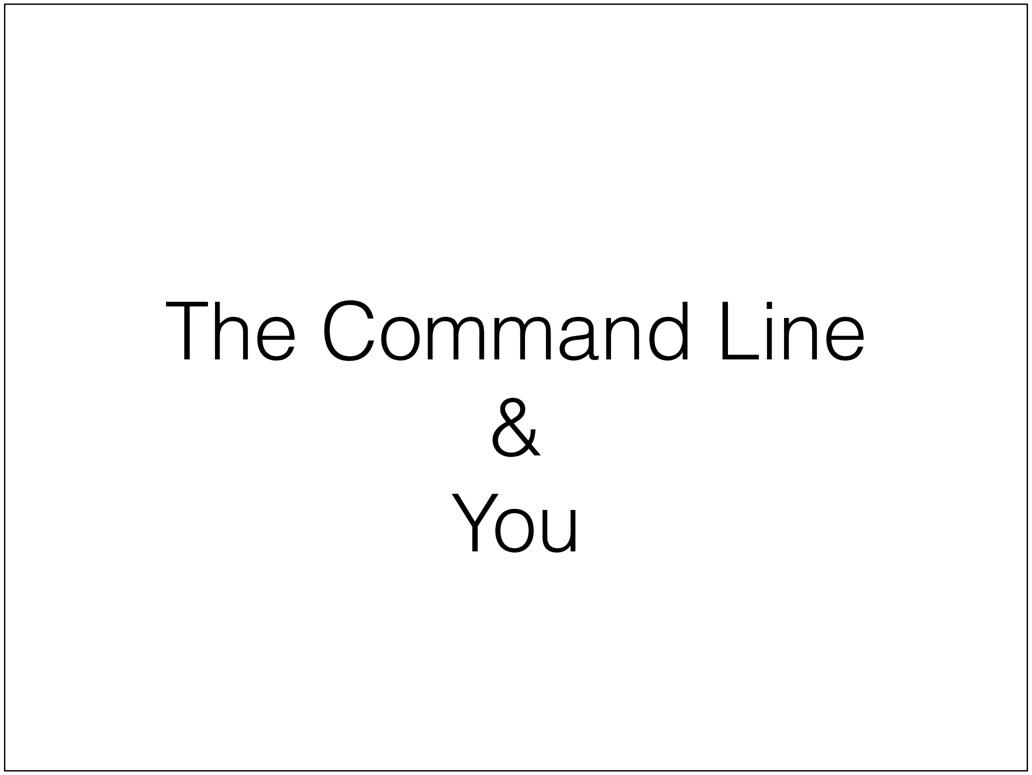
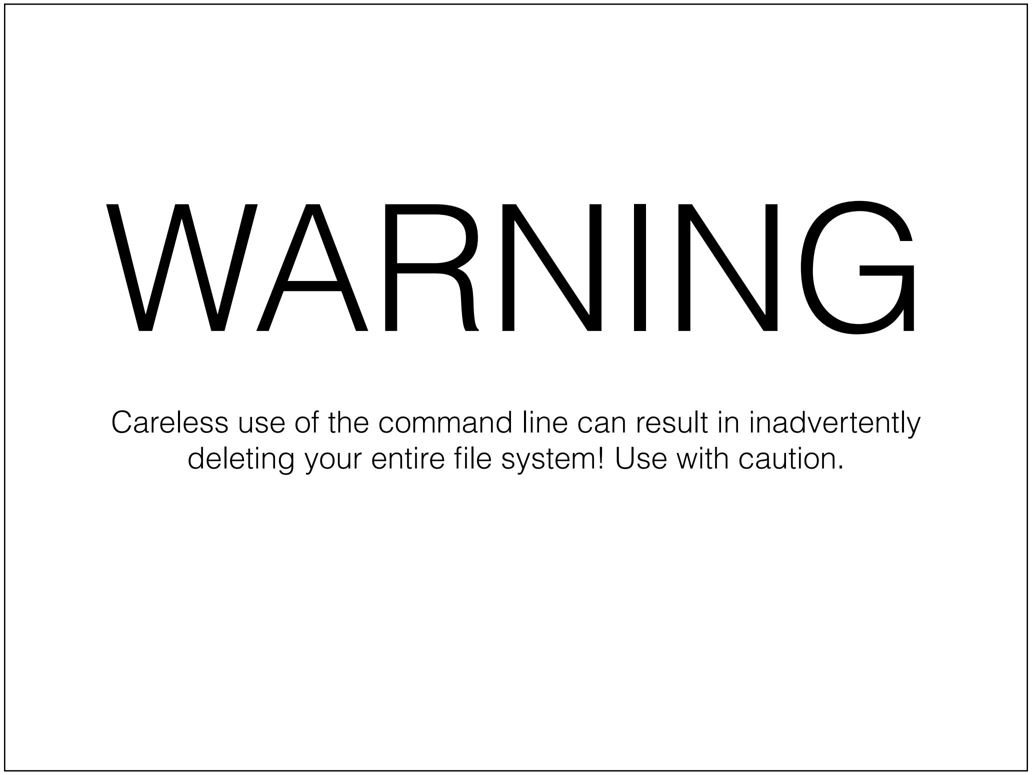


The Command Line
&
You

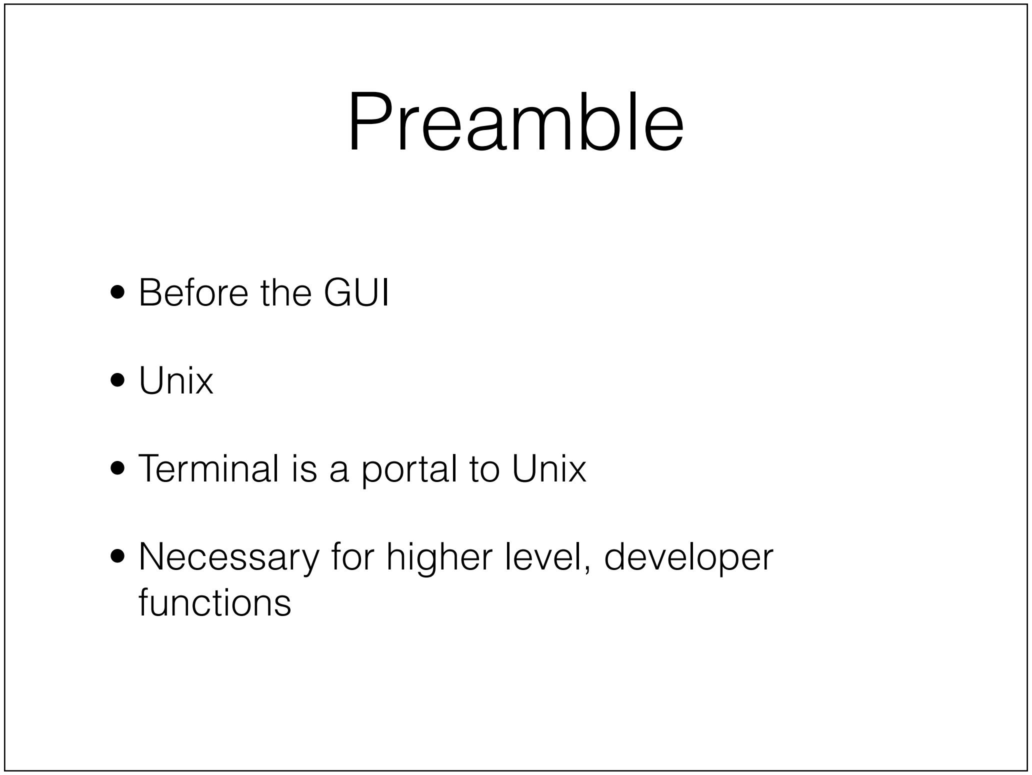
Or "how I learned to look like a developer in most coffee shops"



WARNING

Careless use of the command line can result in inadvertently deleting your entire file system! Use with caution.

Be careful!
Luckily these are school machines
Irreversible unlike regular delete



Preamble

- Before the GUI
- Unix
- Terminal is a portal to Unix
- Necessary for higher level, developer functions

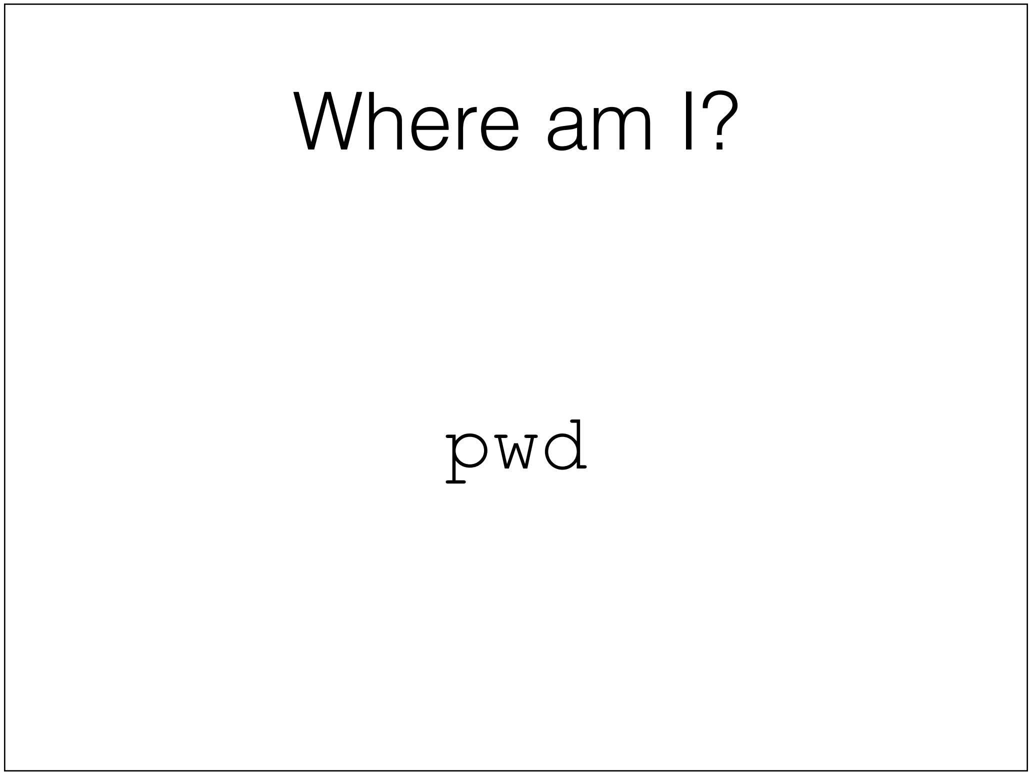
GUI = Graphical user interface (thank you xerox And Steve jobs)
Before, everything was done with the command line
Imagine if this was your entire computer experience!
Unix is the low level system Mac runs on, and Linux
Cmd is a portal to unix
Needed for precise control and access

Open "Terminal"

For easy access:
Press Command + Space, then type in terminal. Press enter.

Entering Commands

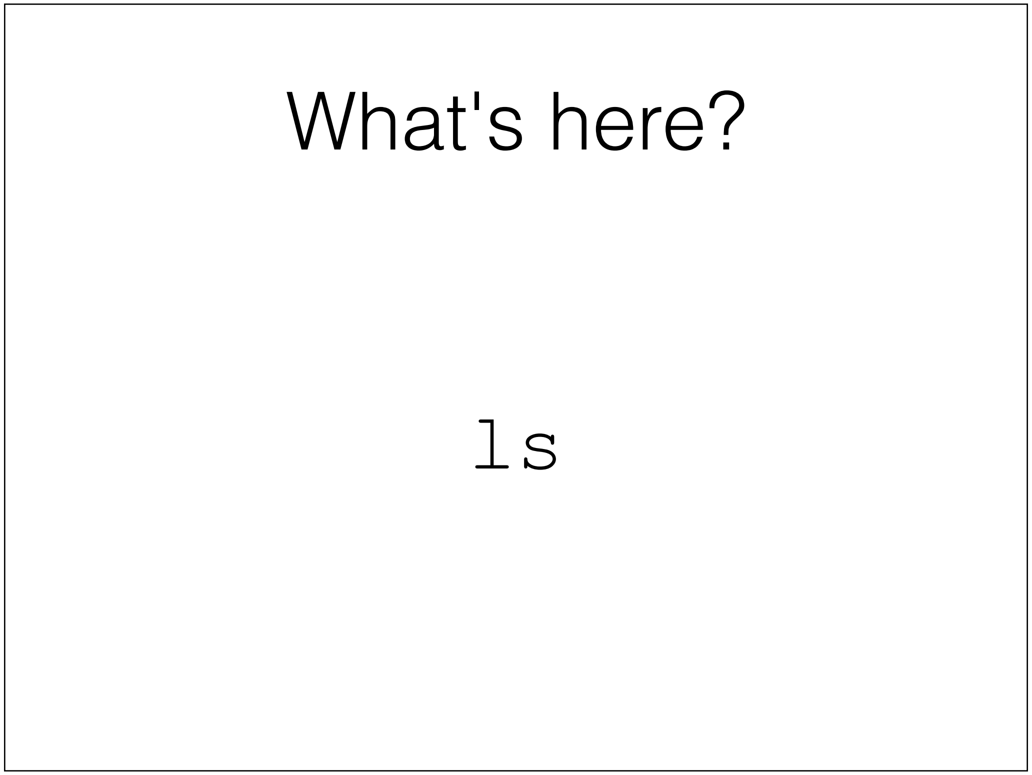
- Make sure the window is in focus
- Type your desired command
- Press enter



Where am I?

```
pwd
```

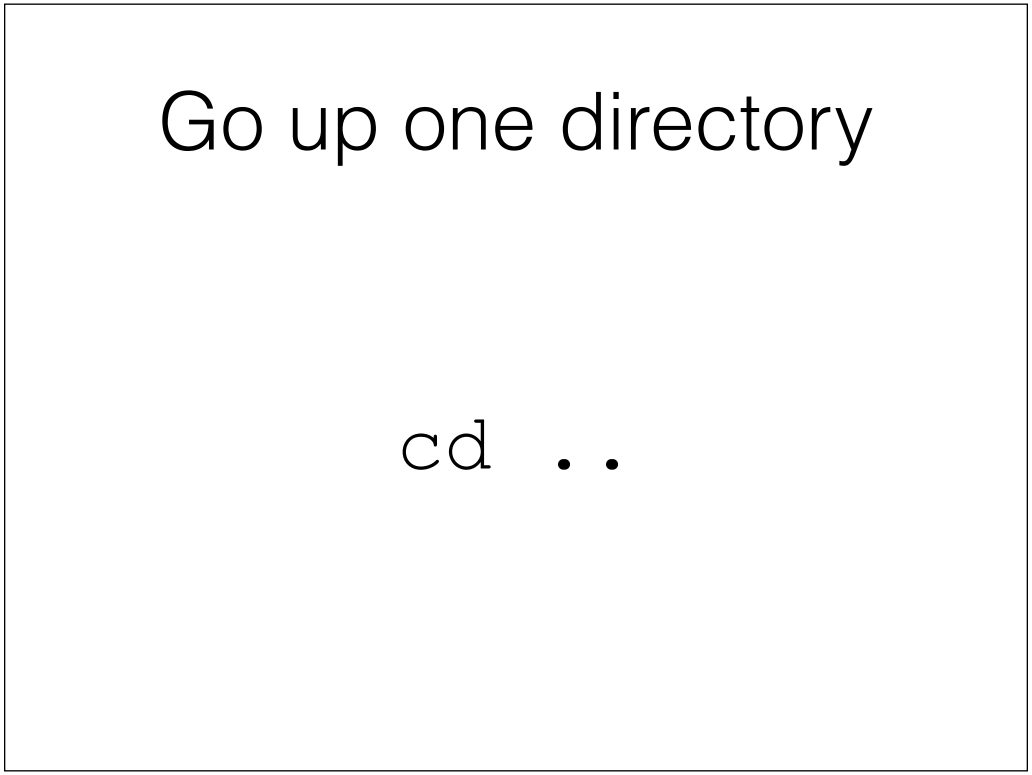
pwd = path with directory
Absolute path!
North Star of the file system



What's here?

```
ls
```

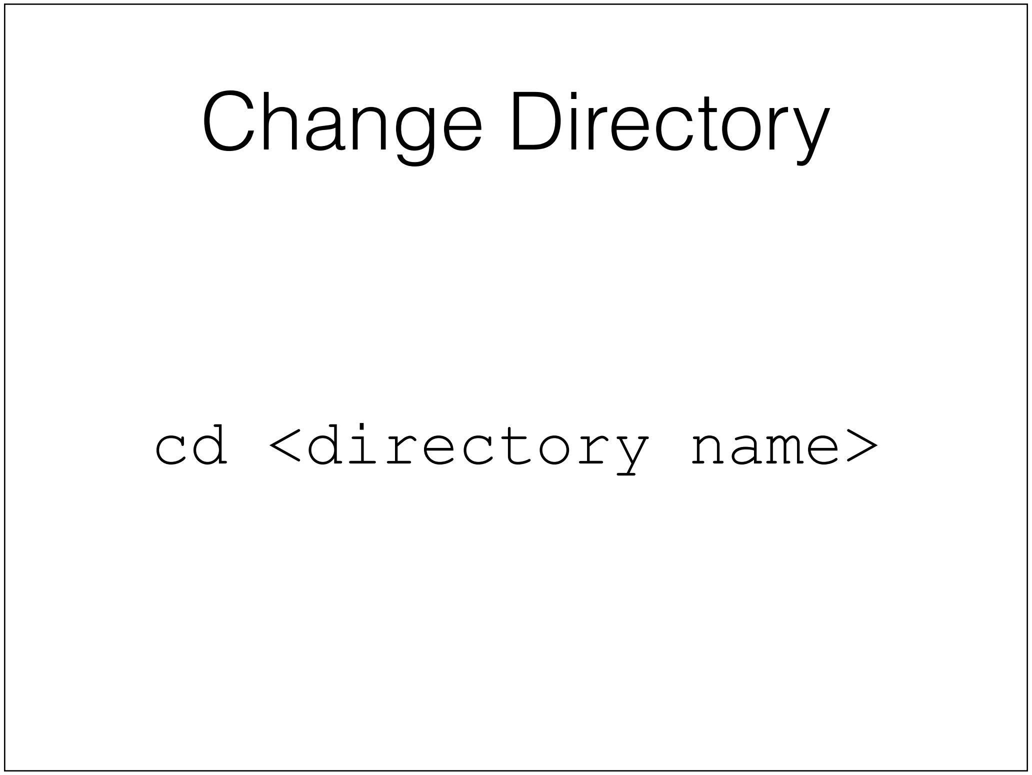
Ls = list
Use -l flag to show permissions and a nice looking list



Go up one directory

```
cd ..
```

Ls = list
Use -l flag to show permissions and a nice looking list



Change Directory

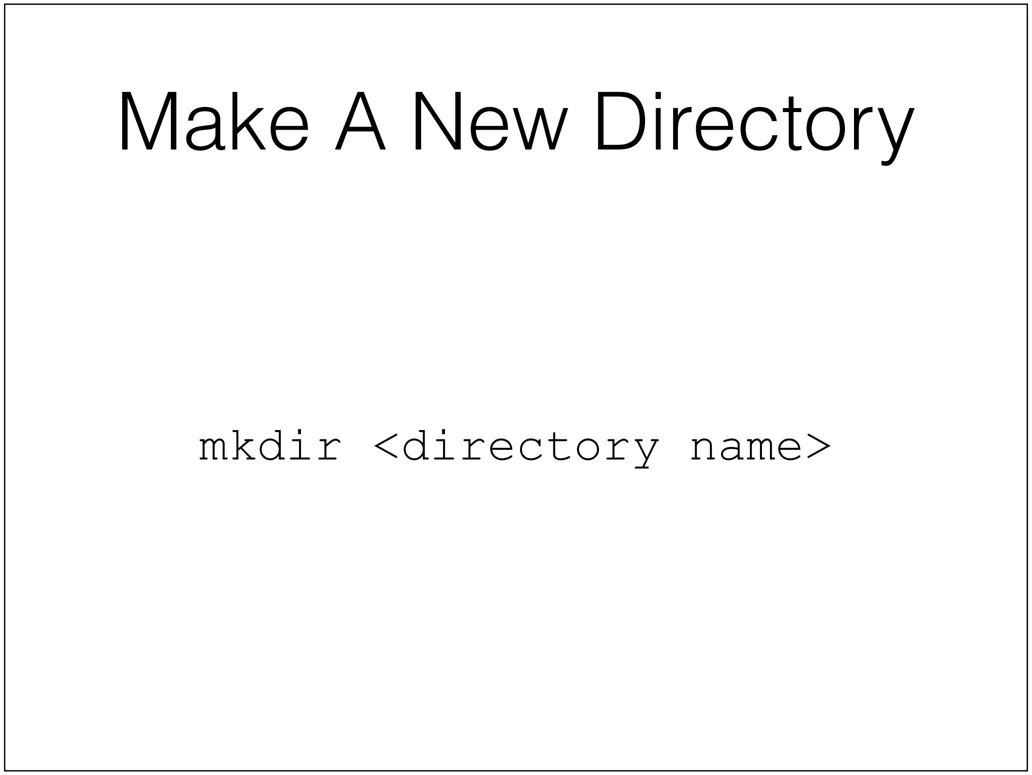
```
cd <directory name>
```

Ls = list
Use -l flag to show permissions and a nice looking list

Tab is Your Buddy!

- Tab will first try to complete what you are typing

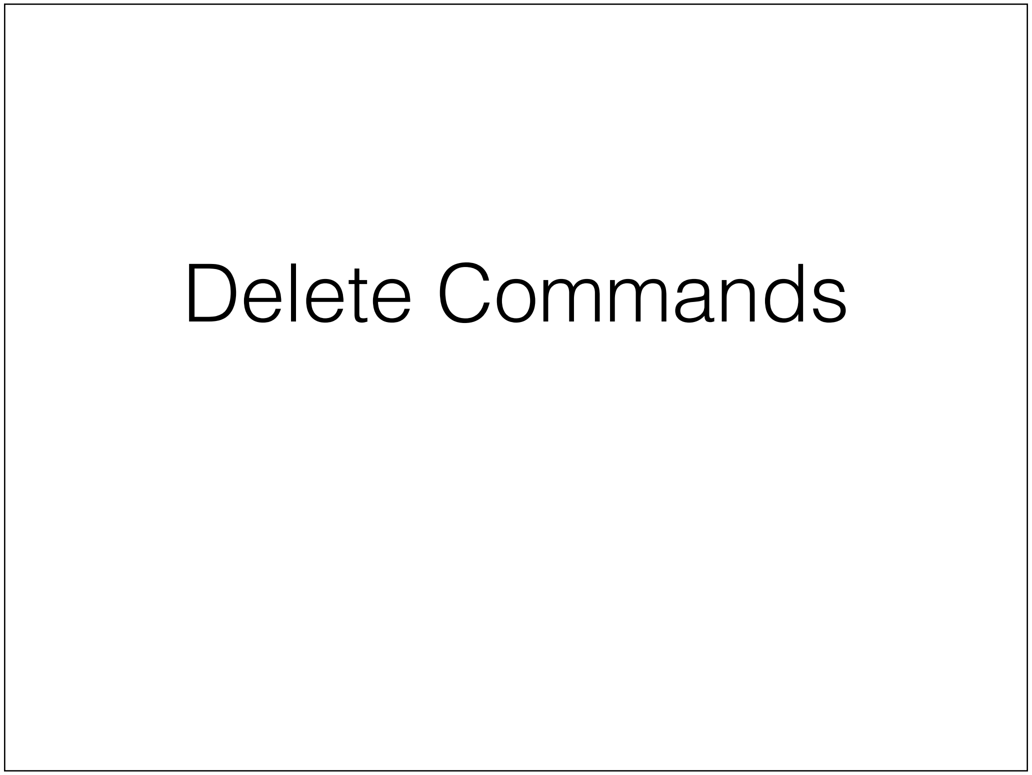
- If you press it twice, it will list the options you have



Make A New Directory

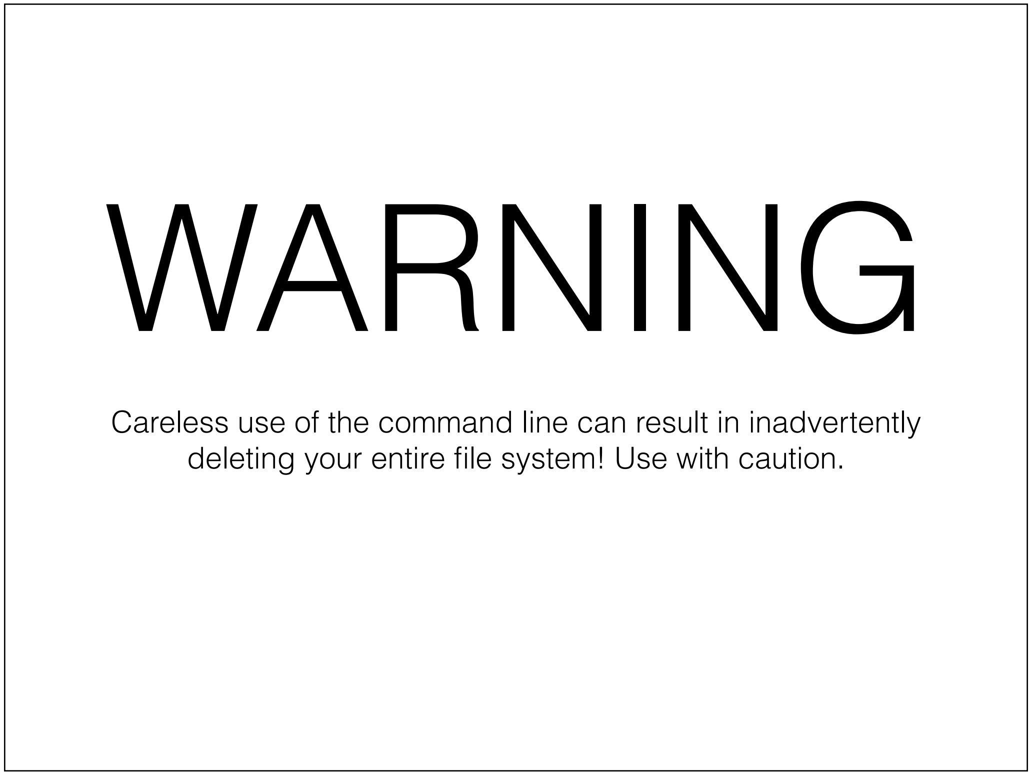
```
mkdir <directory name>
```

mkdir = make directory



Delete Commands

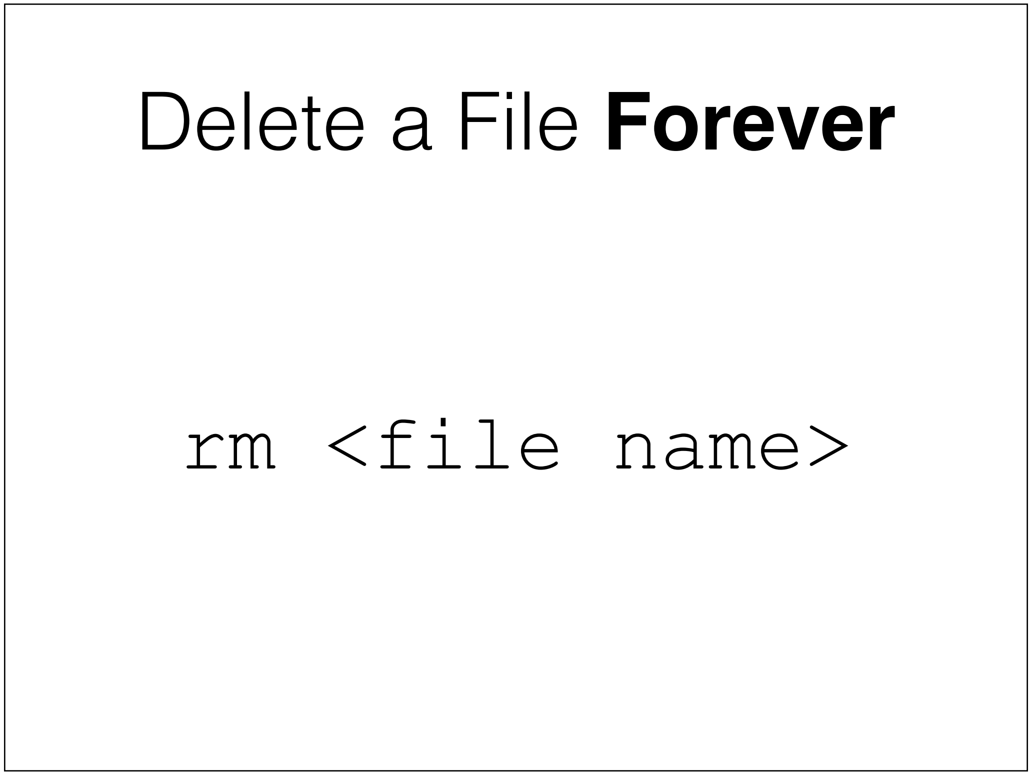
mkdir = make directory



WARNING

Careless use of the command line can result in inadvertently deleting your entire file system! Use with caution.

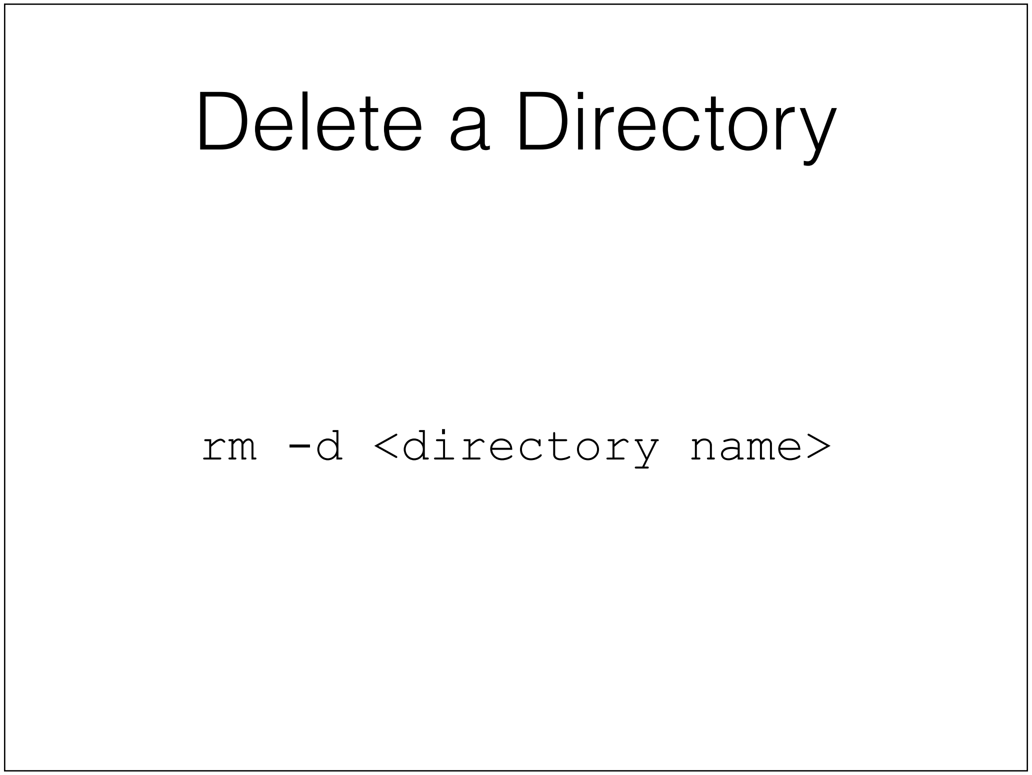
Be careful!
Luckily these are school machines
Irreversible unlike regular delete



Delete a File **Forever**

```
rm <file name>
```

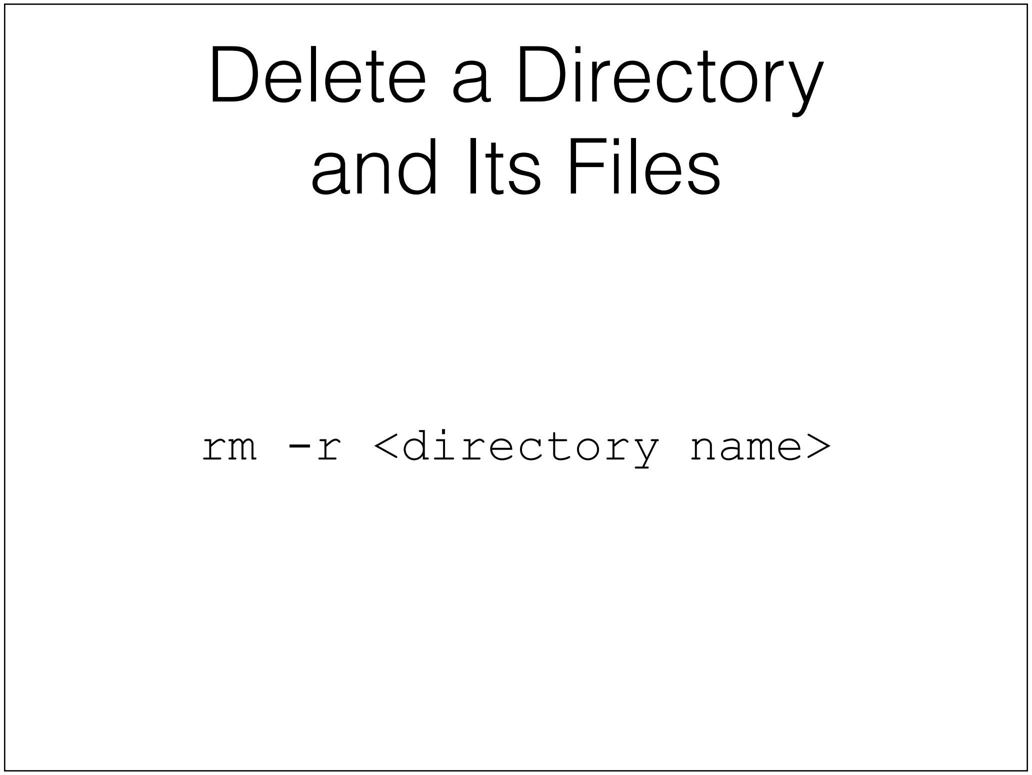
mkdir = make directory



Delete a Directory

```
rm -d <directory name>
```

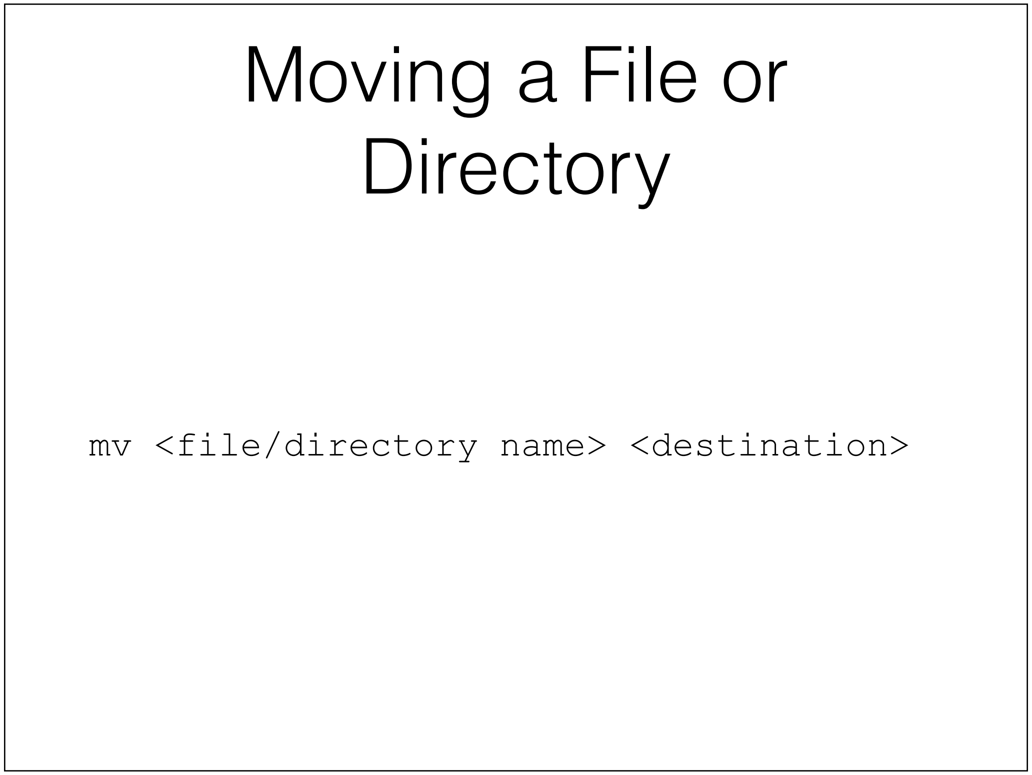
mkdir = make directory



Delete a Directory and Its Files

```
rm -r <directory name>
```

the -r flag is for recursive
a flag allows you to specify options with a command



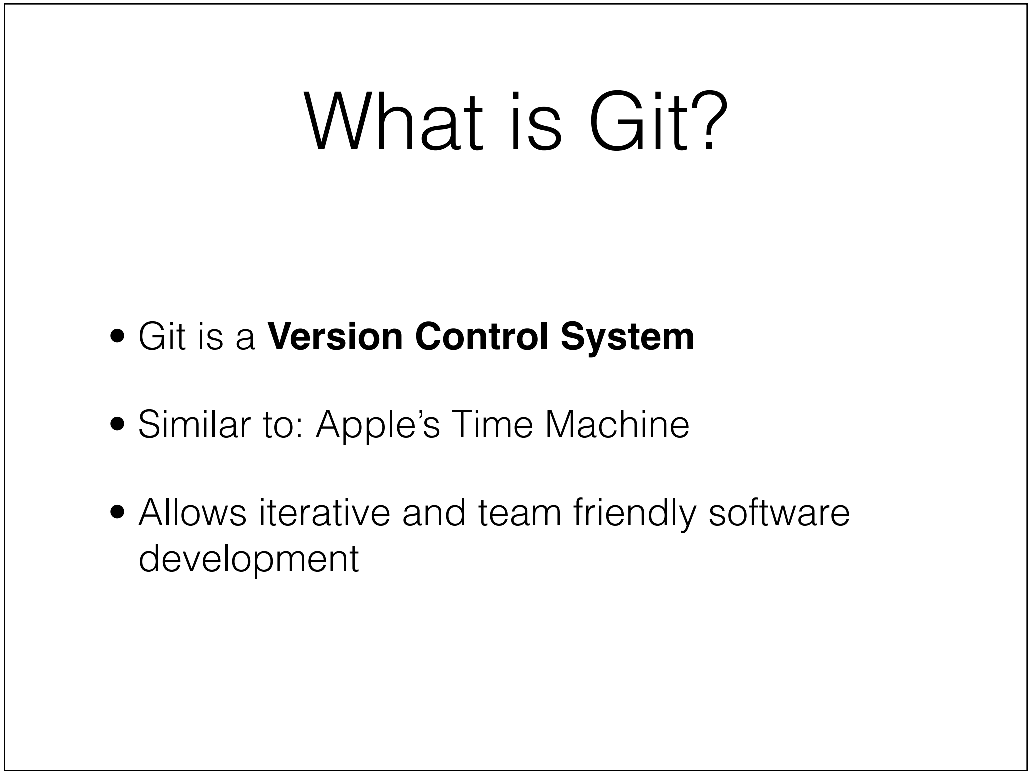
Moving a File or Directory

```
mv <file/directory name> <destination>
```

the -r flag is for recursive
a flag allows you to specify options with a command

Git

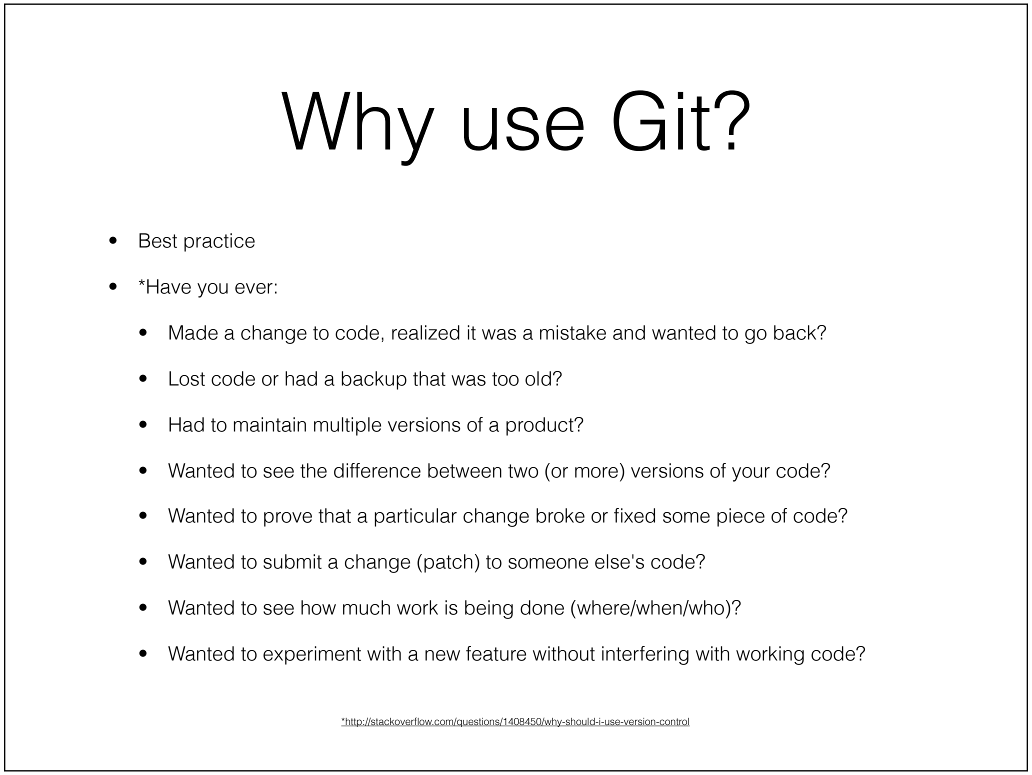
Easy version control from the command line



What is Git?

- Git is a **Version Control System**
- Similar to: Apple's Time Machine
- Allows iterative and team friendly software development

-primarily run from command line
-similar to time machine:
	-allows versions of code all the way back to the first commit
	-you can rewind your work if you mess up to go back to what was working
	-if your code breaks, you can go back and see what was different in the code that made it work before

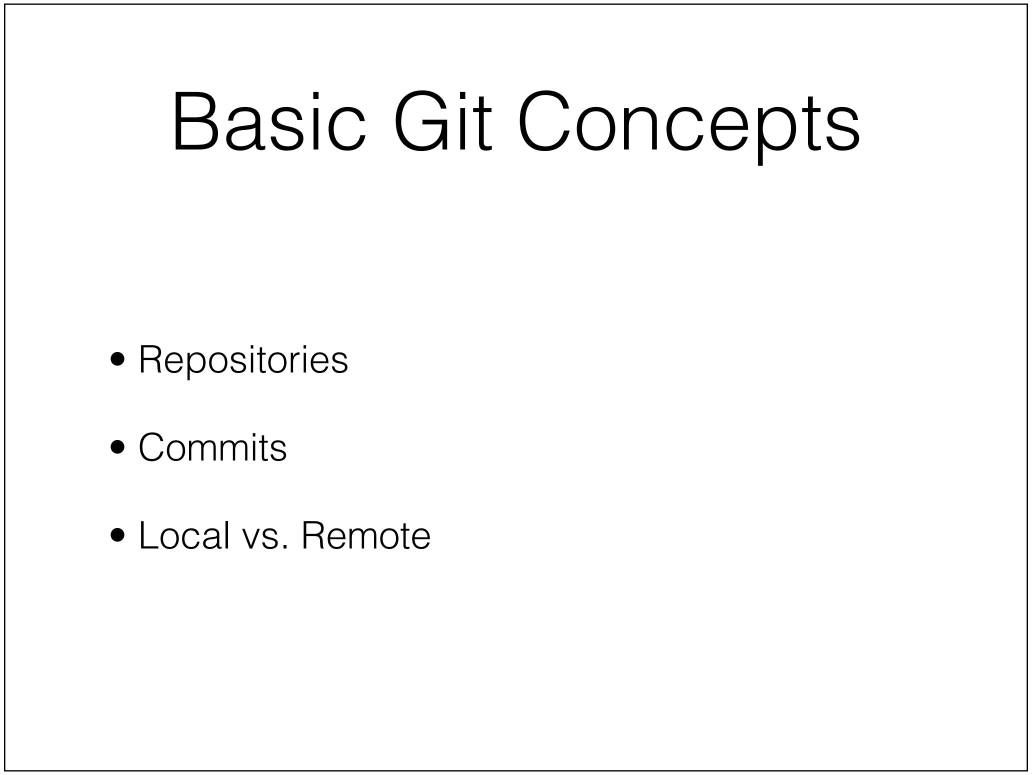


Why use Git?

- Best practice
- *Have you ever:
    - Made a change to code, realized it was a mistake and wanted to go back?
    - Lost code or had a backup that was too old?
    - Had to maintain multiple versions of a product?
    - Wanted to see the difference between two (or more) versions of your code?
    - Wanted to prove that a particular change broke or fixed some piece of code?
    - Wanted to submit a change (patch) to someone else's code?
    - Wanted to see how much work is being done (where/when/who)?
    - Wanted to experiment with a new feature without interfering with working code?

*http://stackoverflow.com/questions/1408450/why-should-i-use-version-control

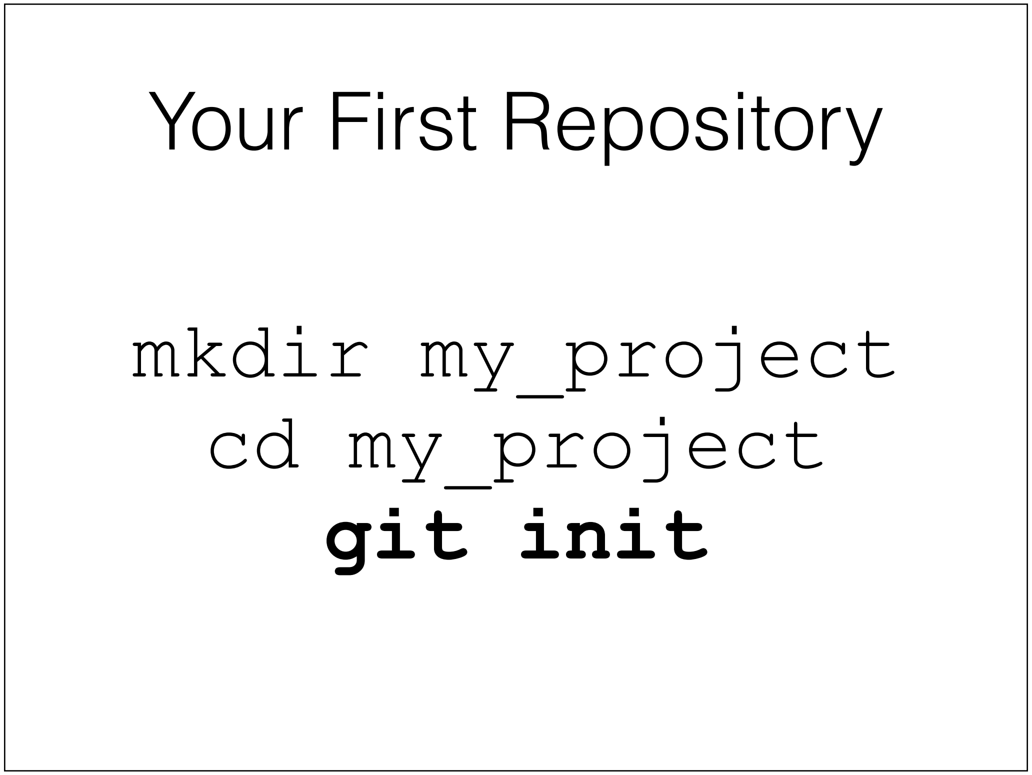
-Source control is used at almost all major programming companies today, whether SVN or Git
-At this point, it is basically a best practice



Basic Git Concepts

- Repositories
- Commits
- Local vs. Remote

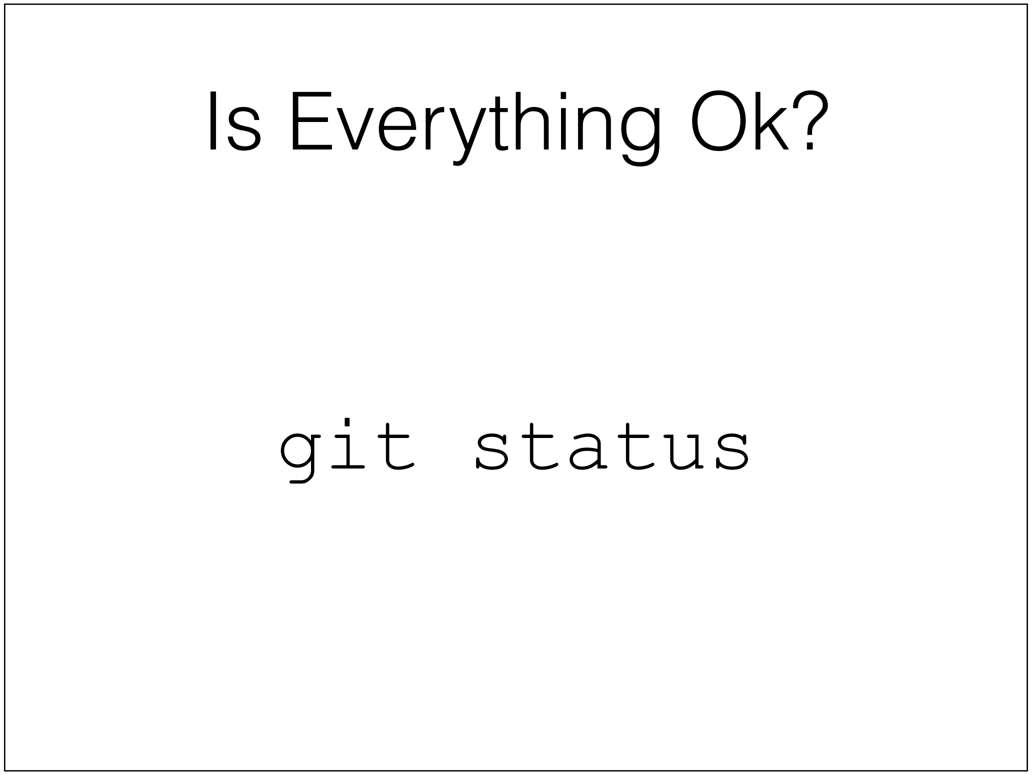
-Source control is used at almost all major programming companies today, whether SVN or Git
-At this point, it is basically a best practice



Your First Repository

```
mkdir my_project
cd my_project
git init
```

-once git is installed on your machine, the only command you need to start a new repository is git init!



Is Everything Ok?

```
git status
```

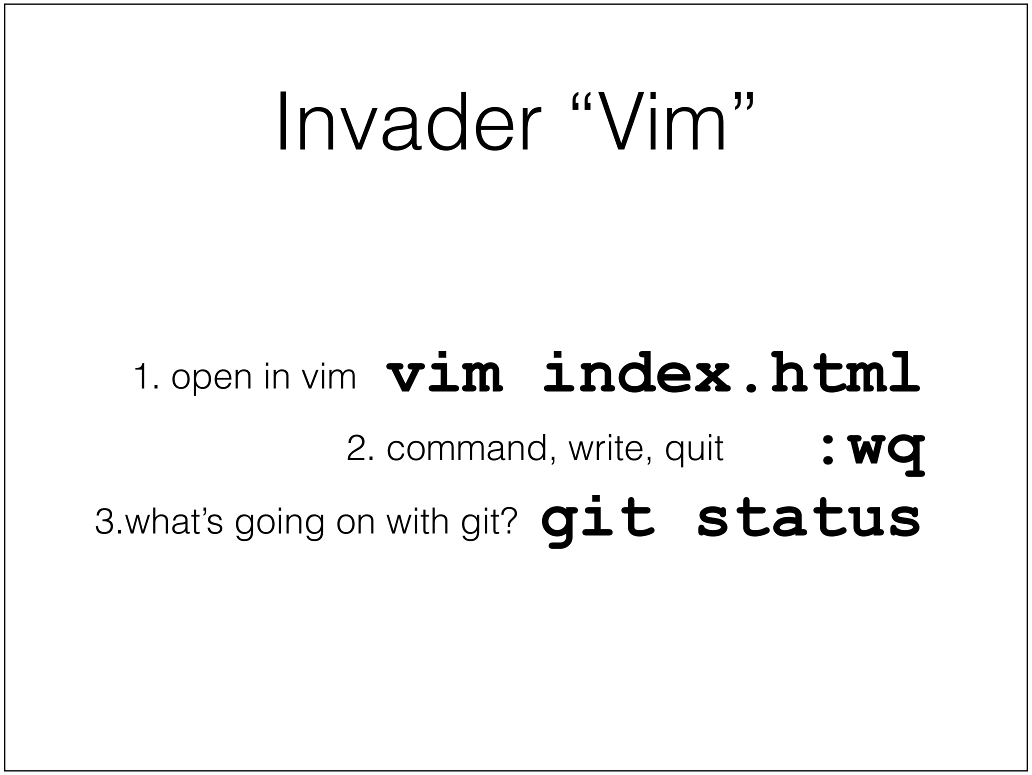
–You should see “On branch master – initial commit – nothing to commit”

Vim

- Simple, command line text editor
- Easy way to create a new file without leaving command line

*http://stackoverflow.com/questions/1408450/why-should-i-use-version-control

-You should see “On branch master - initial commit - nothing to commit”



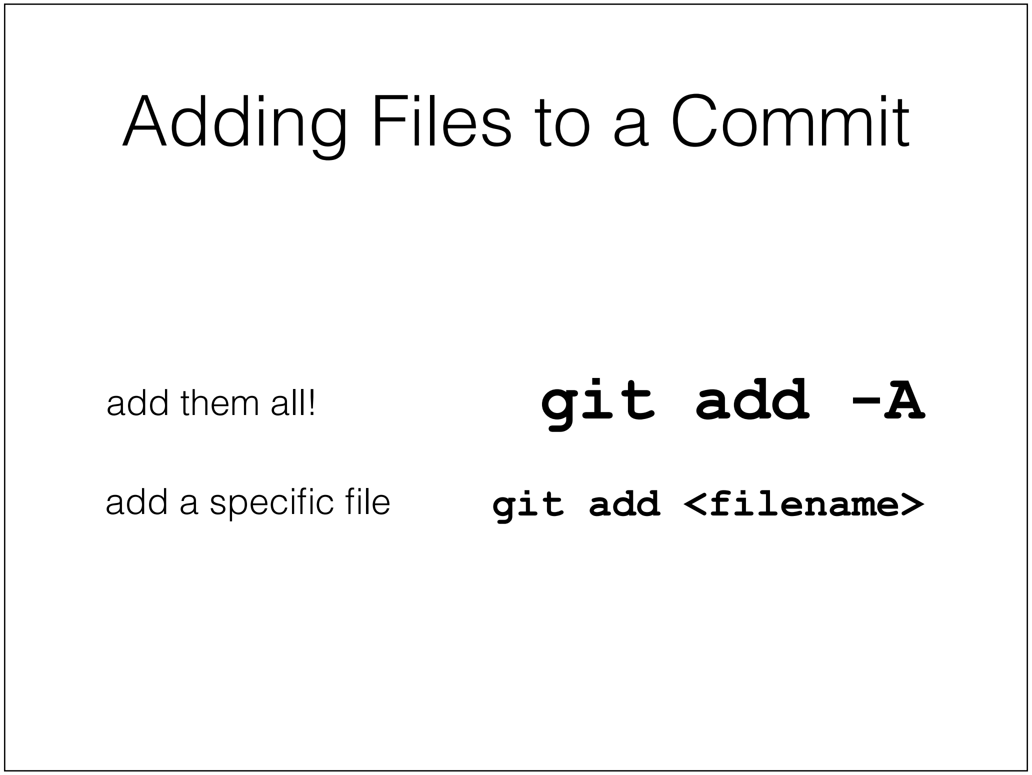
Invader “Vim”

1. open in vim `vim index.html`

2. command, write, quit `:wq`

3.what’s going on with git? `git status`

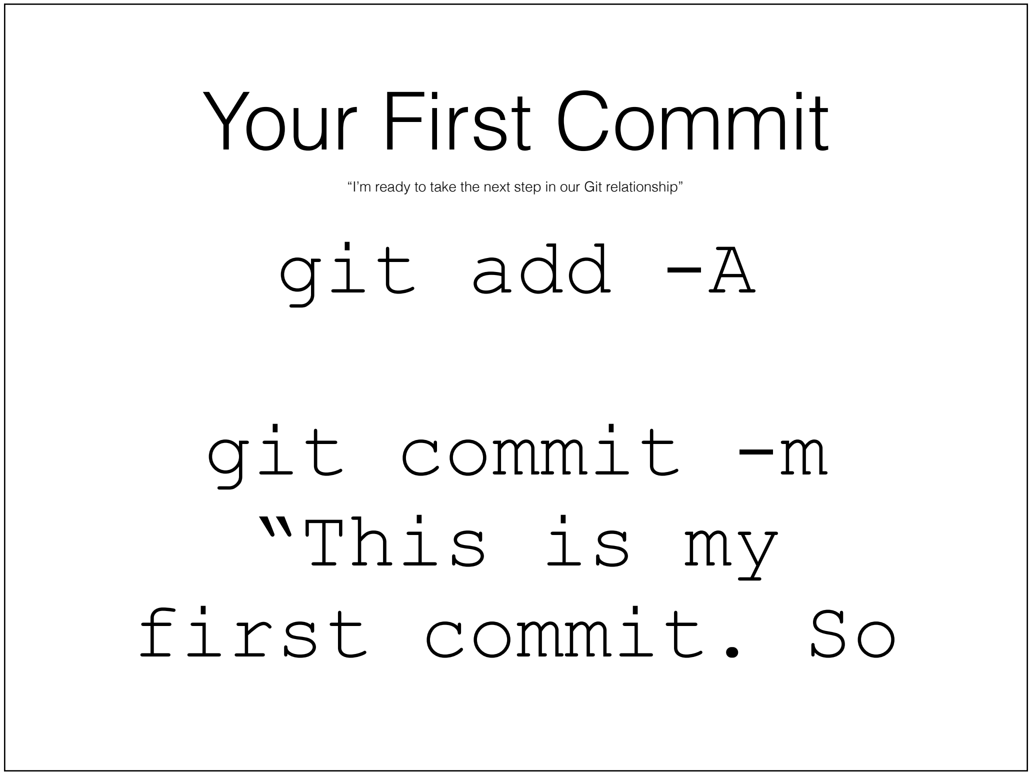
-to create a simple file, type vim <filename>
-to type a command in vim, use colon. w means write. q means quit.
-git status to see your changes reflected.



Adding Files to a Commit

add them all!	`git add -A`
add a specific file	`git add <filename>`

-A is another example of a flag!
You can also add directories instead of files.
Tab is incredibly useful here!



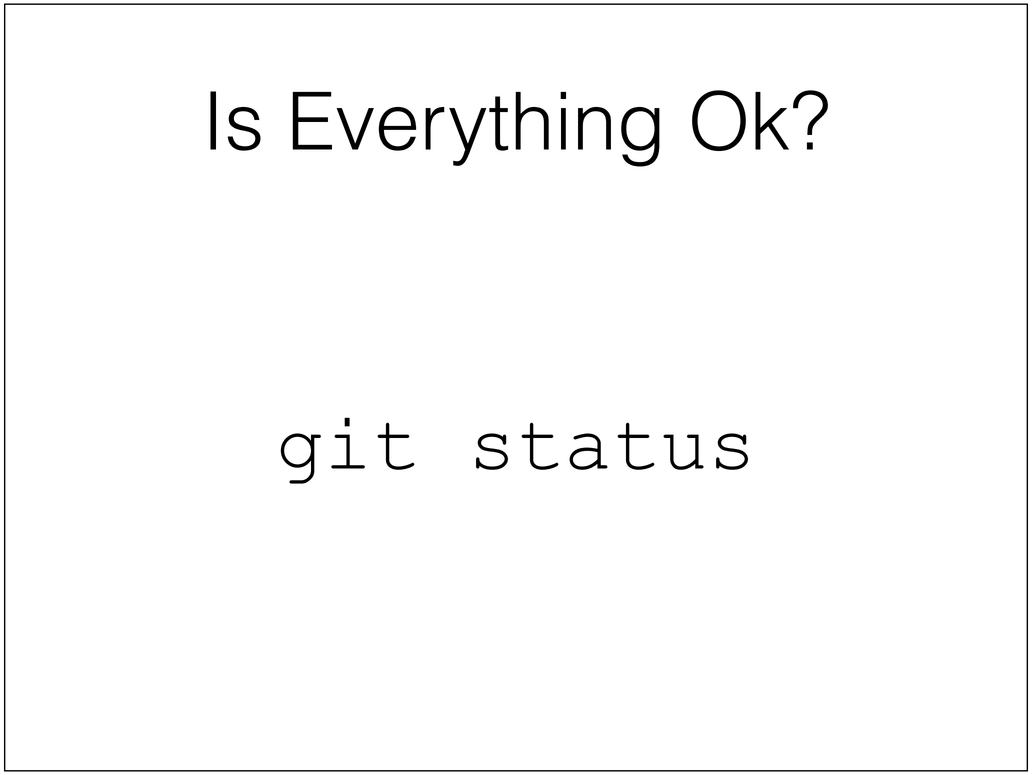
Your First Commit

"I'm ready to take the next step in our Git relationship"

```
git add -A

git commit -m
"This is my
first commit. So
```

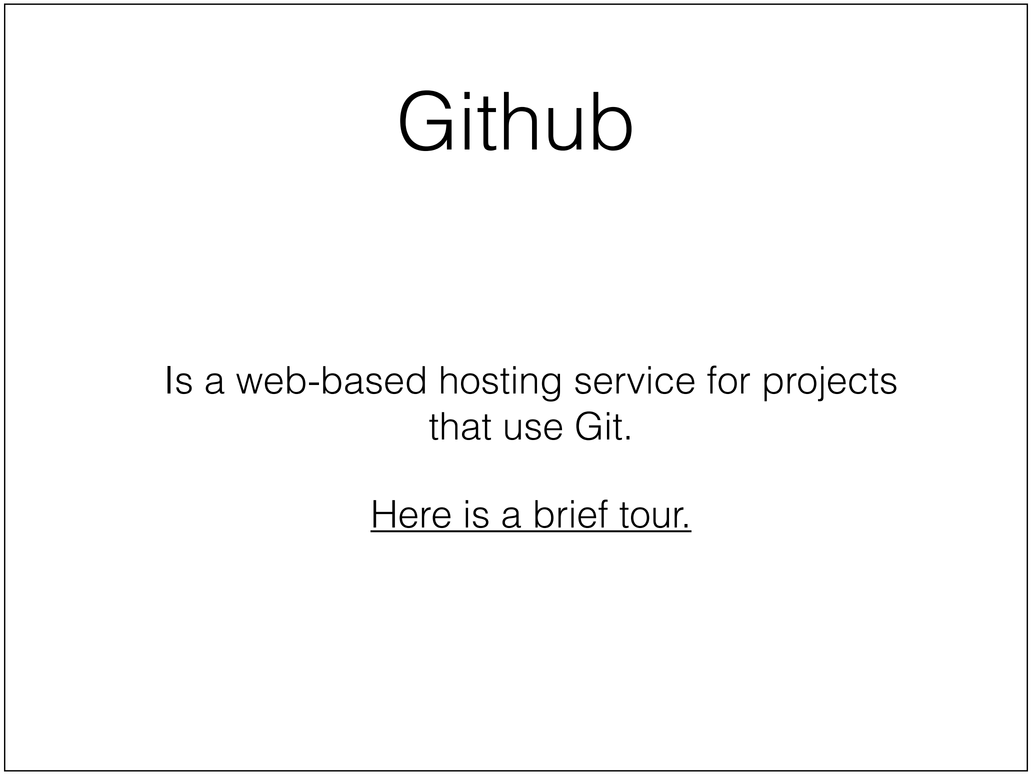
-You should see "On branch master - initial commit - nothing to commit"



Is Everything Ok?

```
git status
```

-You should see "on branch master, working directory clean."



Github

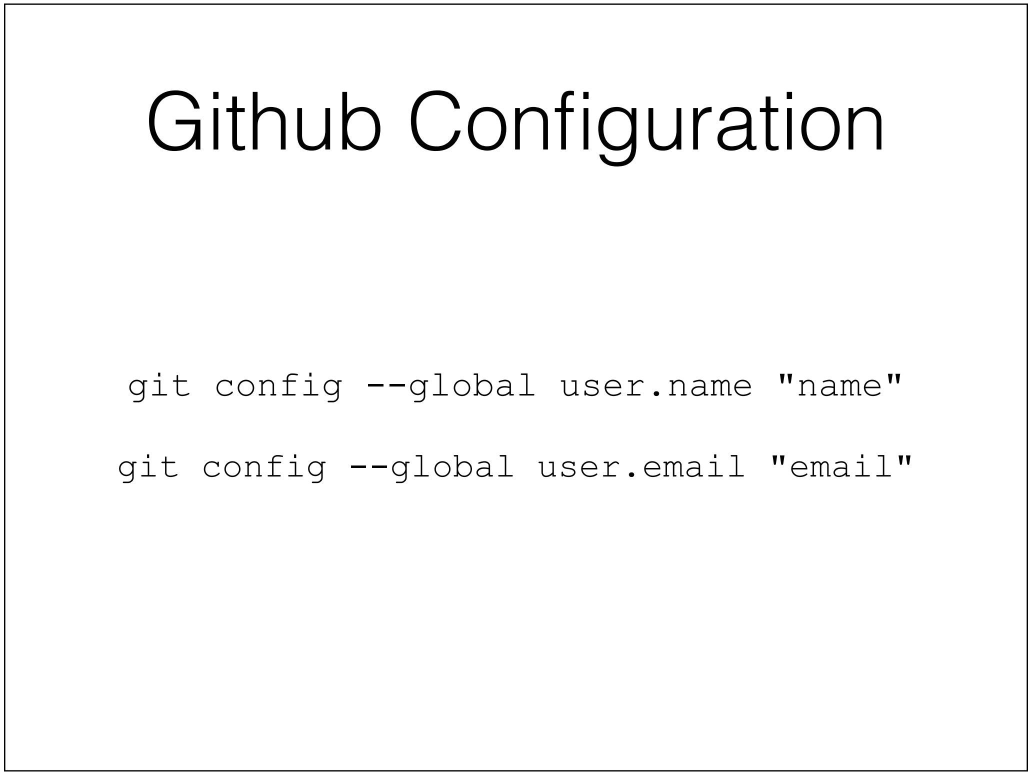
Is a web-based hosting service for projects
that use Git.

Here is a brief tour.

Github is not the only site like it of its kind, bitbucket, a few other competitors exist

used by many startups to stay organized:

- issues
- code review/commenting "from the year 3000"

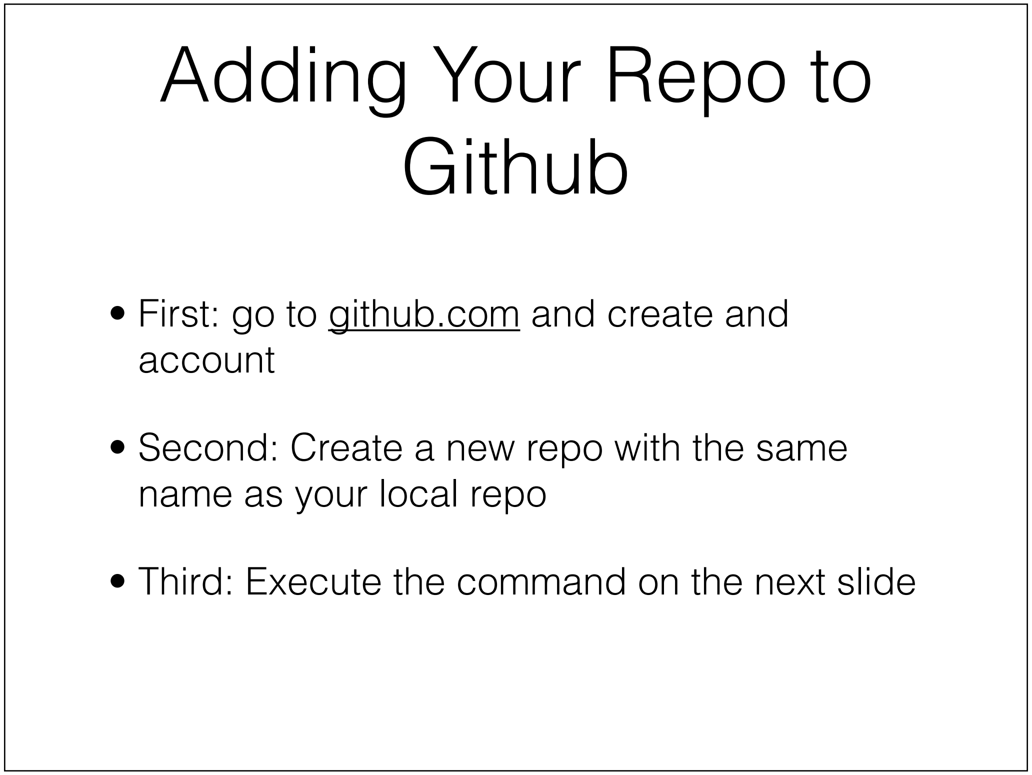


Github Configuration

```
git config --global user.name "name"

git config --global user.email "email"
```

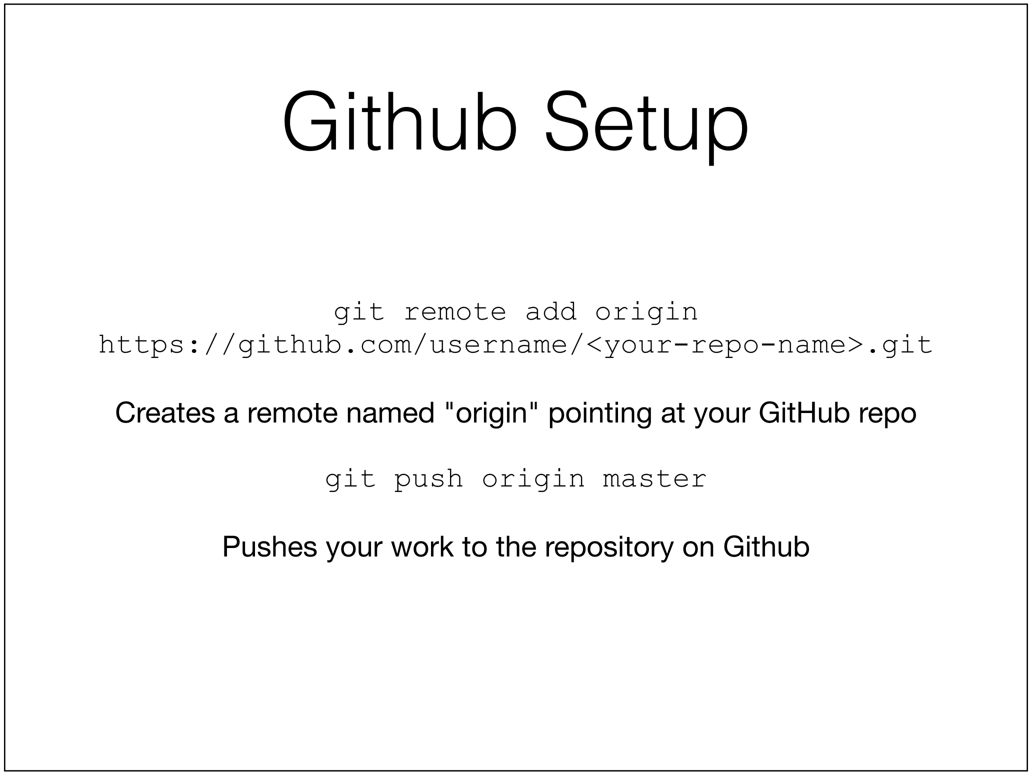
This initial configuration will make it so that your commits have your name on them.
-may only want to do this on your personal computer, to avoid confusino



Adding Your Repo to Github

- First: go to github.com and create and account
- Second: Create a new repo with the same name as your local repo
- Third: Execute the command on the next slide

-first you must create your repo on github, then you can push to it
-use the same name as the folder you created locally if your repo was created locally first



Github Setup

```
git remote add origin
https://github.com/username/<your-repo-name>.git
```

Creates a remote named "origin" pointing at your GitHub repo

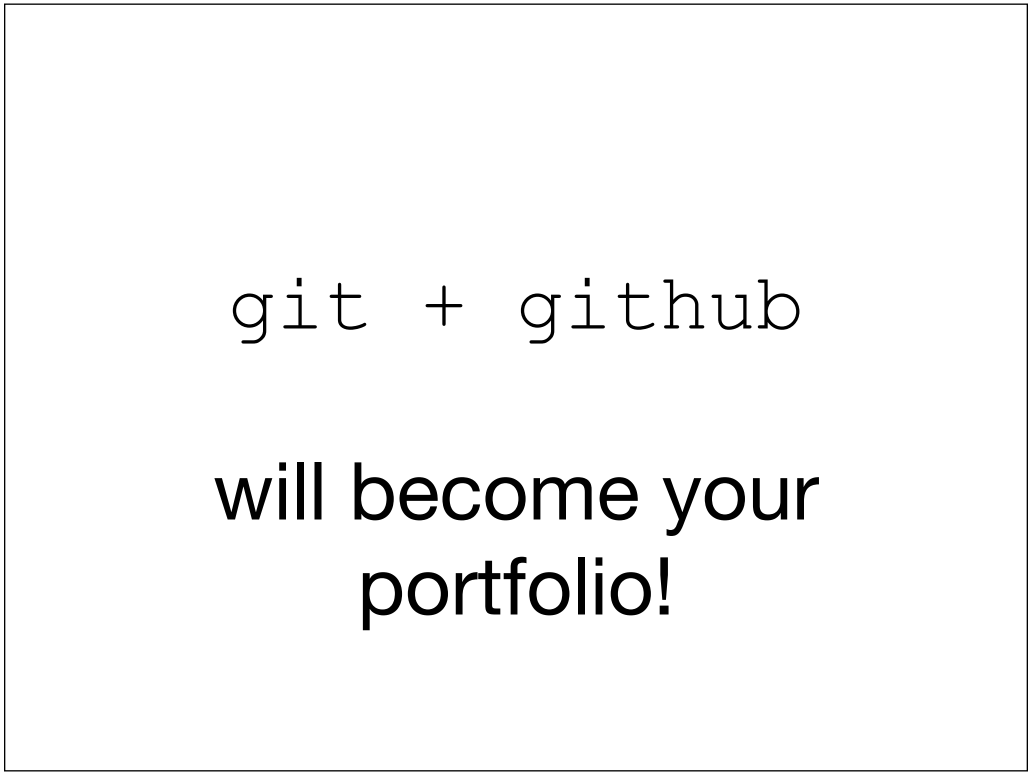
```
git push origin master
```

Pushes your work to the repository on Github

git remote add origin
git - hey git!
remote -check out this remote
add - and add it.
origin - oh yeah, the name of the remote is going to be "origin".

git push origin master
git - hey git
push - push UP to a remote
origin - that remote being remote "origin"
master - to the branch "master"

-sometimes, it's easier to use git clone after you've created a fresh repository on github.
    -go into the folder you'd like your repo to live in and type "git clone <repo url found on git>



`git + github`

will become your portfolio!

your github will be:
-a history of this class
-proof that you are a programmer
-an easy way for you to send me and dio your homework and tell us exactly what line is "bugging" you
-a portfolio for potential employers (you can remove early examples later on, and always do another commit to tweak code)
     -"send me your github"

later on, we will be using Git to deploy your Ruby on Rails applications to a service called "Heroku". Feel free to do some research on it if you'd like!