

# Introduction to the Bash Unix Shell & GitHub

Lewis Blake

August 20, 2020

AMS Bootcamp  
Department of Applied Mathematics & Statistics  
Colorado School of Mines



# In this talk...

I will try to:

- Introduce you to the VERY basics of Bash and GitHub
- Show you how I use these tools
- Prepare a roadmap to get everyone started
- Get you to [follow me on GitHub](#)

I will not:

- Explain every command
- Pretend this is the only way to accomplish many of these things
- Present them as best practices
- Follow you back (kidding)

---

# The Bash Unix Shell



## What is Unix?

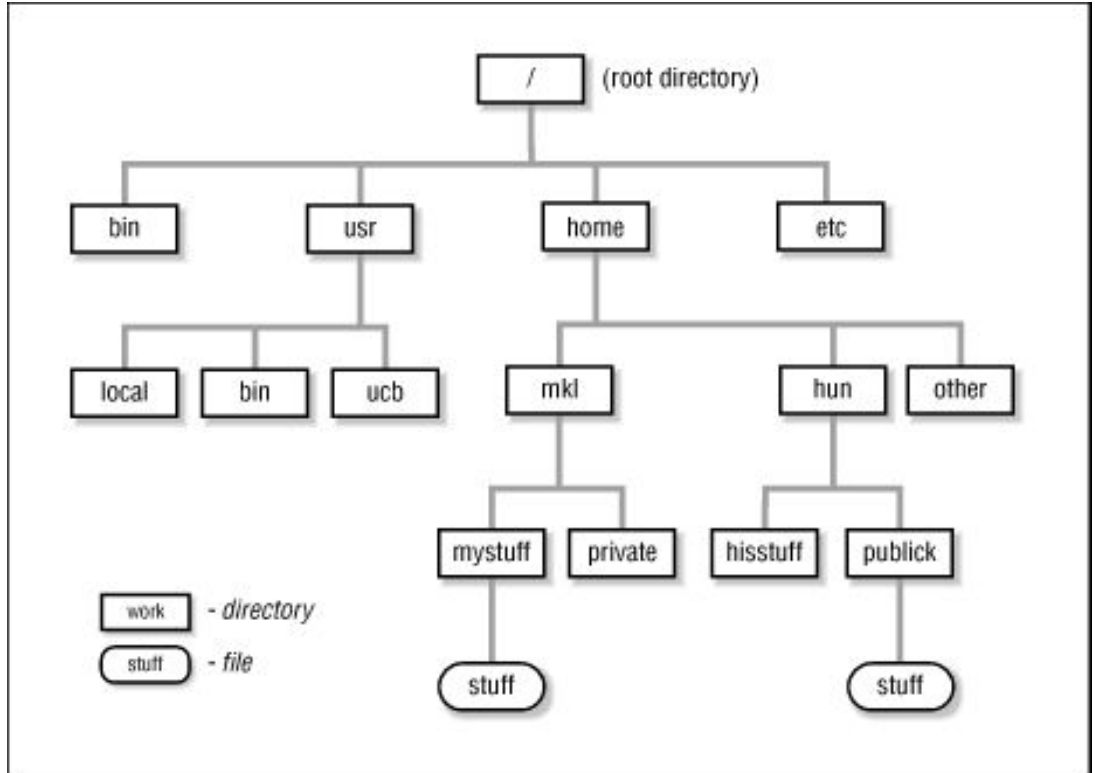
- (family of) **Operating system(s)**
- Unix is basically an open standard
- Many OSs are “Unix-like”: Linux, MacOS

## What is Bash?

- Command line interpreter for Unix-like OSs
- Interactive & Scripting language
- Default login shell for most Linux distributions & MacOS

# Unix File System

- Hierarchical “tree” structure
- A “directory” is a “folder”
- Directories can contain other directories and files



# File pathing



- Each directory & file in this tree structure has a unique file path
  - E.g., `/home/mkl/mystuff/stuff`
- Unix-like systems use forward slashes
- There are *absolute* and *relative* file paths.
  - Absolute: specifying the location of a file or directory from the root directory (/).
  - Relative: specify the location of a file or directory from current working directory

# Some Common Bash Commands

<u>Bash Command</u>	<u>Description</u>	<u>Example Uses</u>
pwd	Print working directory	pwd
ls	List contents of directory	ls , ls mydir
cd	Change directory	cd mydir , cd .. , cd ~
mkdir	Make directory	mkdir mydir
rm	Remove	rm myfile.R , rm -r mydir
cp	Copy	cp myfile.R ../
mv	Move	mv myfile.R ../
touch	Create file	touch myfile.R
cat	Display file contents	cat myfile.R



## Some Bash resources

[Bash Cheat Sheet](#)

[University of Washington Bash Reference Manual](#)

[GNU Bash Documentation](#)

MasOS: Terminal, or [iTerm2](#)

Windows: [Git for Windows](#)





# An Example

```
(base) [Lewisblake@Lewiss-MacBook-Pro ~]$ pwd
/Users/Lewisblake
(base) [Lewisblake@Lewiss-MacBook-Pro ~]$ ls
Applications          Dropbox              Library              Music                java.log.2118
Desktop              Dropbox (Old)       MEGAsync            Pictures             sill_grid_mse_matrix.Rdata
Documents            Dropbox (Your team) MEGAsync Downloads  Public              vcpkg
Downloads           Google Drive        Movies              anaconda3

(base) [Lewisblake@Lewiss-MacBook-Pro ~]$ cd Documents/School/Mines
(base) [Lewisblake@Lewiss-MacBook-Pro Mines]$ ls
F17      F18      F19      F20      Miscellaneous  NCAR18      NCAR19      S18      S19      S20

(base) [Lewisblake@Lewiss-MacBook-Pro Mines]$ mkdir tmp
(base) [Lewisblake@Lewiss-MacBook-Pro Mines]$ cd tmp
(base) [Lewisblake@Lewiss-MacBook-Pro tmp]$ pwd
/Users/Lewisblake/Documents/School/Mines/tmp
(base) [Lewisblake@Lewiss-MacBook-Pro tmp]$ ls
(base) [Lewisblake@Lewiss-MacBook-Pro tmp]$ touch tmp.py
(base) [Lewisblake@Lewiss-MacBook-Pro tmp]$ ls
tmp.py
(base) [Lewisblake@Lewiss-MacBook-Pro tmp]$ cd ~
(base) [Lewisblake@Lewiss-MacBook-Pro ~]$ pwd
/Users/Lewisblake
(base) [Lewisblake@Lewiss-MacBook-Pro ~]$
```

---

# Version Control with Git & Github



# What is Version Control?

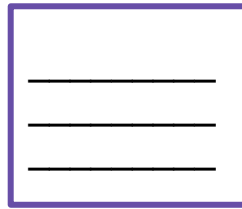
- Version control allows you to record changes to your files over time
- Allows different people to work on the same project at the same time
- Can go back to previous versions if needed



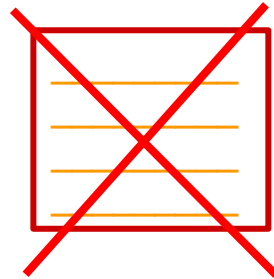
S1



S2



S3





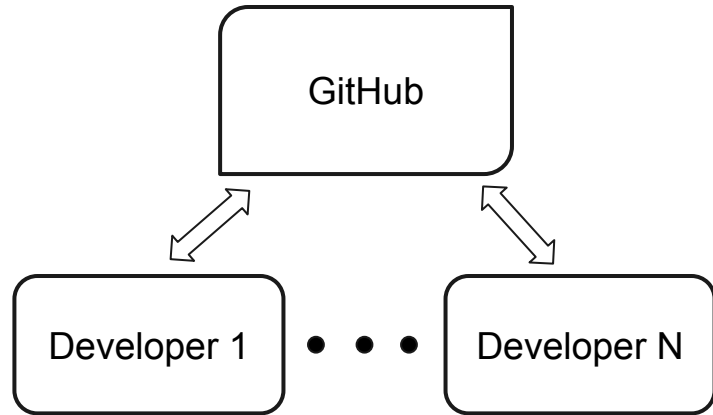
# What is Git?

**Version Control System** for tracking changes in computer files

- Distributed version control (Local & Remote Repositories)
- Personal work tracking
- Collaborations between developers
- Lightweight & fast

# What is GitHub?

Website interface to store projects





## Git

- Keeps track of code history
- Take “snapshots” of your files
- You decide when to take snapshots by making a `commit`
- You can visit snapshots at any time

## GitHub

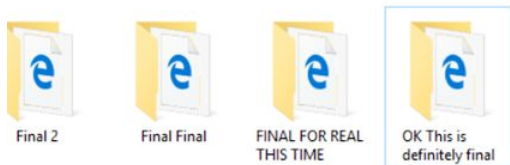
- Remote hosting website
- Get street cred for coding



I prefer the real version control



I said the *real* version control




Perfection



## Step 1: Create a GitHub account and install Git

Create a Github account

Install Git



You already  
did this!





## Step 1.1: Configure git on your machine

After installing git, in the terminal type:

- 1) `git config --global user.name "Your name here"`
- 2) `git config --global user.email "your_email@example.com"`

To check configuration: `git config --list`

[Useful link](#)



## Step 2: Create a Repository on GitHub

Top LHS home page:

**Repositories**



Find a repository...

# Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Owner



Repository name \*



Great repository names are short and memorable. Need inspiration? How about **legendary-eureka**?

Description (optional)

This is an example for the talk I'm giving in Math Bio.



**Public**



Anyone can see this repository. You choose who can commit.



**Private**



You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.



**Initialize this repository with a README**

This will let you immediately clone the repository to your computer.

Add .gitignore: **None** ▼

Add a license: **None** ▼



Create repository

Click this!



This is an example for the talk I'm giving in Math Bio.

Edit

[Manage topics](#)

1 commit

1 branch

0 packages

0 releases

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download



lewisblake Initial commit

Latest commit c8f0e92 now



README.md

Initial commit

now



README.md



# MathBioTalk

This is an example for the talk I'm giving in Math Bio.

<> Code ! Issues 0 🔗 Pull requests 0 ▶ Actions 📁 Projects 0 🛡 Security 0 📊 Insights ⚙ Settings

This is an example for the talk I'm giving in Math Bio.

Edit

[Manage topics](#)

🔗 1 commit

🔗 1 branch

📦 0 packages

🏷 0 releases

Branch: master ▾

New pull request

Create new file

Upload files

Find file

Clone or download ▾



lewisblake Initial commit

📄 [README.md](#) Initial commit

📄 README.md

# MathBioTalk

This is an example for the talk I'm giving in Math Bio.

Clone with HTTPS ?

[Use SSH](#)

Use Git or checkout with SVN using the web URL.

<https://github.com/lewisblake/MathBioTalk>



Open in Desktop

Download ZIP



## Step 3: Clone GitHub Repository Locally

Option 1: Command line git (already installed)

Option 2: [GitHub Desktop App](#) (not covering)



## Step 3: Terminal

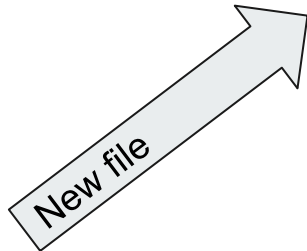

```
(base) [lewisblake@Lewiss-MacBook-Pro GittinIt]$ pwd
/Users/lewisblake/Documents/School/Mines/Miscellaneous/GittinIt
(base) [lewisblake@Lewiss-MacBook-Pro GittinIt]$ ls
progit.pdf
(base) [lewisblake@Lewiss-MacBook-Pro GittinIt]$ git clone https://github.com/lewisblake/MathBioTalk.git
```

```
(base) [lewisblake@Lewiss-MacBook-Pro GittinIt]$ ls
MathBioTalk      progit.pdf
(base) [lewisblake@Lewiss-MacBook-Pro GittinIt]$
```



## Step 4: Make changes in local repo

```
(base) [lewisblake@Lewiss-MacBook-Pro MathBioTalk]$ pwd
/Users/lewisblake/Documents/School/Mines/Miscellaneous/GittinIt/MathBioTalk
(base) [lewisblake@Lewiss-MacBook-Pro MathBioTalk]$ ls
README.md          helloworld.py
```







## Essential Git Commands

- `git add`
- `git commit`
- `git push`
- `git pull`



## git add

Adds files to be part of next commit

Uses:

- `git add .` // Adds all files in to next commit
- `git add -A` // “ ”
- `git add <filename1>, <filename2>`

**Note** `git add` has a counterpart: `git rm`



## git commit

Takes a “snapshot” of the repository

e.g.

- `git commit`
- `git commit -m "Include commit message here."`

Note: `git commit` (i.e., w/o `-m "<message>"` will open up Vim. More on that later.)



## git push

Move local commit to central remote repository (in this case GitHub).

e.g.

- `git push`



## git pull

Fetch and integrate from GitHub to local repository

e.g.

- `git pull`

```
(base) [lewisblake@Lewiss-MacBook-Pro MathBioTalk]$ pwd
/Users/lewisblake/Documents/School/Mines/Miscellaneous/GittinIt/MathBioTalk
(base) [lewisblake@Lewiss-MacBook-Pro MathBioTalk]$ ls
README.md      helloworld.py
(base) [lewisblake@Lewiss-MacBook-Pro MathBioTalk]$ git add .
(base) [lewisblake@Lewiss-MacBook-Pro MathBioTalk]$ git commit -m "Initial Commit"
[master 705f437] Initial Commit
 1 file changed, 1 insertion(+)
 create mode 100644 helloworld.py
(base) [lewisblake@Lewiss-MacBook-Pro MathBioTalk]$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 308 bytes | 308.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/lewisblake/MathBioTalk.git
   c8f0e92..705f437  master -> master
(base) [lewisblake@Lewiss-MacBook-Pro MathBioTalk]$
```



## Next time I work on the repository

Always do a `git pull` first!

```
(base) [lewisblake@Lewiss-MacBook-Pro MathBioTalk]$ pwd
/Users/lewisblake/Documents/School/Mines/Miscellaneous/GittinIt/MathBioTalk
(base) [lewisblake@Lewiss-MacBook-Pro MathBioTalk]$ ls
README.md      helloworld.py
(base) [lewisblake@Lewiss-MacBook-Pro MathBioTalk]$ git pull
Already up to date.
(base) [lewisblake@Lewiss-MacBook-Pro MathBioTalk]$
```

**IF YOU DON'T GIT PULL BEFORE  
MAKING CHANGES**



**YOU'RE GONNA HAVE A BAD TIME**

MEMES BY WAGAT



---

# More Advanced Git

## Types of Headache

Migraine



Hypertension



Stress



GIT ERRORS



# Getting into more advanced topics...

## Disclaimer:

Up until now I've shown you the basics which should help you get started and keep track of your work.

This comprises about 90% my personal Git & Github use.

My (unsolicited) advice is to get comfortable with the four essential commands and core concepts before going deeper.

In my experience, going further is how I've opened myself up to a world of pain and frustration.

There is a lot about Git I still have to learn.

Proceed with diligence and caution.



## Core Concepts of Git

Recall: Snapshots of our projects are called commits.

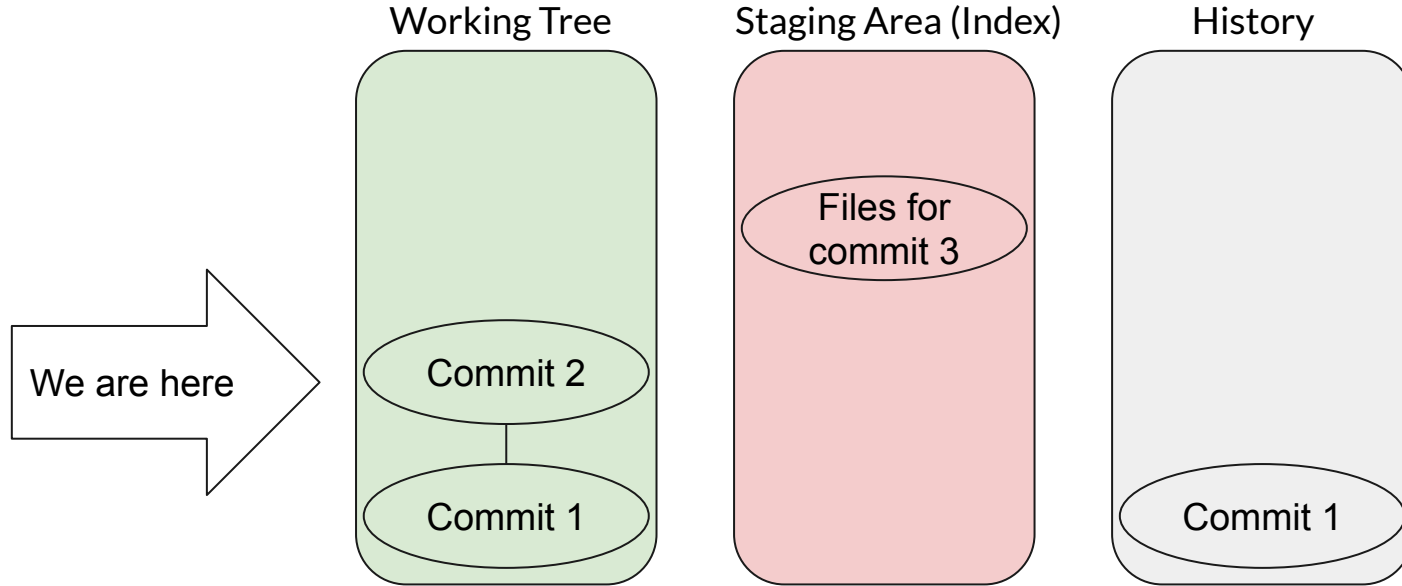
Each commit has a unique hexadecimal index called a commit hash.

Sequences of commits are called a “branch”.

The primary branch is called the “master” branch.

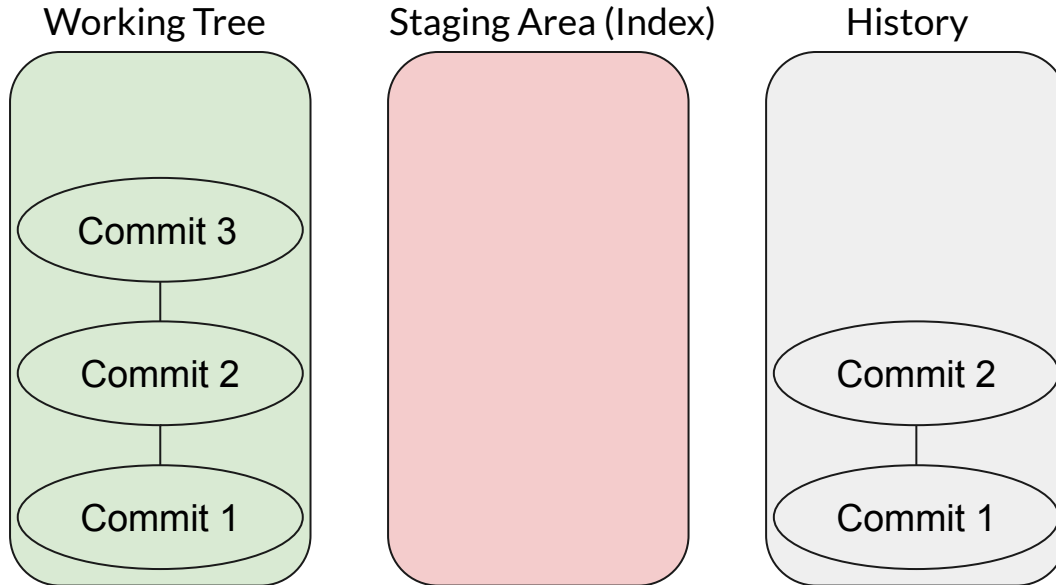


# Three conceptual areas: an example

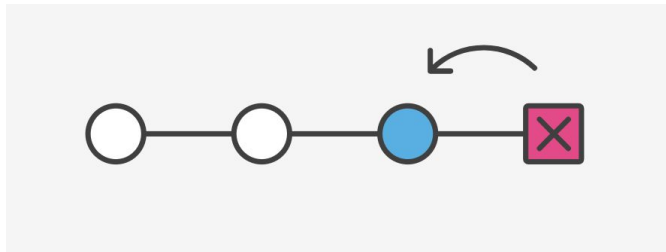




## Example cont.



# Undoing commits and changes



Your friend: `git log`, `git log --oneline`

- Test a specific commit: `git checkout <commit hash>`
- `git revert` is the best tool for undoing shared public changes
- `git reset` is best used for undoing local private changes

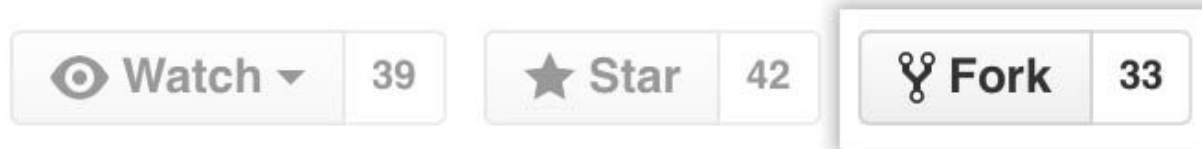
[Good tutorial I stole the above image from.](#)



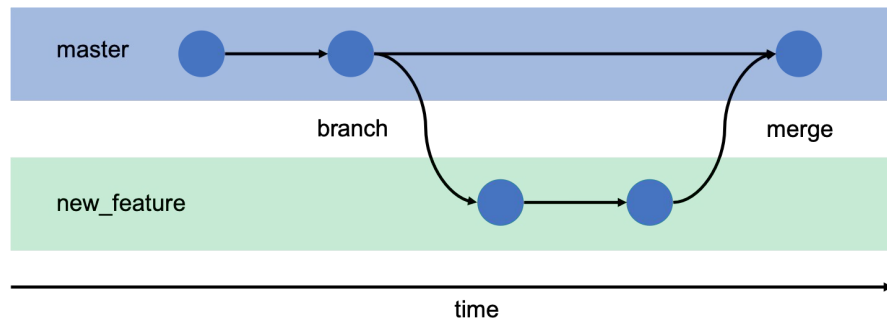
# Forking

A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.

On GitHub go to the repository. On the top-right corner of the page click “Fork”.



# Branching & Merging



The “master” branch is by default the primary branch on which we work.

We can create other branches to test new ideas for our code without fear of losing our previous work.

New branch: `git checkout -b <branch_name>`

Existing branch: `git checkout <branch_name>`





# Branching and Merging Example

```
(base) [lewisblake@Lewiss-MacBook-Pro MathBioTalk]$ git branch
* master
(base) [lewisblake@Lewiss-MacBook-Pro MathBioTalk]$ git checkout -b dev
Switched to a new branch 'dev'
(base) [lewisblake@Lewiss-MacBook-Pro MathBioTalk]$ git branch
* dev
  master
(base) [lewisblake@Lewiss-MacBook-Pro MathBioTalk]$ ls
README.md      helloworld.py
(base) [lewisblake@Lewiss-MacBook-Pro MathBioTalk]$ vim helloworld.py
(base) [lewisblake@Lewiss-MacBook-Pro MathBioTalk]$ python helloworld.py
Hello world!
I like chunky peanut butter.
```

```
(base) [lewisblake@Lewiss-MacBook-Pro MathBioTalk]$ git commit -a -m "Included important print statement"
[dev cc742c3] Included important print statement
 1 file changed, 1 insertion(+)
(base) [lewisblake@Lewiss-MacBook-Pro MathBioTalk]$ git branch
* dev
  master
(base) [lewisblake@Lewiss-MacBook-Pro MathBioTalk]$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
(base) [lewisblake@Lewiss-MacBook-Pro MathBioTalk]$ git merge dev
Updating 705f437..cc742c3
Fast-forward
 helloworld.py | 1 +
 1 file changed, 1 insertion(+)
(base) [lewisblake@Lewiss-MacBook-Pro MathBioTalk]$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 357 bytes | 357.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/lewisblake/MathBioTalk.git
 705f437..cc742c3  master -> master
(base) [lewisblake@Lewiss-MacBook-Pro MathBioTalk]$ git pull
Already up to date.
```

**Good luck!**



## Useful references

[Visual Git Reference](#)

[Official Documentation](#)

[Setting up an SSH key so you don't have to keep signing in](#)



## All the Vim you'll need

If you commit with just `git commit`, a Vim window will enter fullscreen.

- 1) Hit “i” to enter insert mode.
- 2) Type your commit message.
- 3) Hit Esc
- 4) Type “:wq”
- 5) Hit Enter