COMMAND LINE & GIT

DATE 11/3

Why bother with the command line?

- Godmode for using your computer
- It’s how we will run standalone Ruby applications.
- It’s the best way to issue commands to a remote server, view logs
- Faster than GUIs when you know what you are doing, especially if you limit mouse usage.

Why bother with the command line?

- Godmode for using your computer
- It’s how we will run standalone Ruby applications.
- It’s the best way to issue commands to a remote server, view logs
- Faster than GUIs when you know what you are doing!

```
            state        service
22/tcp      open         ssh

o exact OS matches for host

map run completed -- 1 IP address (1 host up) scanne
sshnuke 10.2.2.2 -rootpw="Z1ON0101"
nnecting to 10.2.2.2:ssh ... successful.
tempting to exploit SSHv1 CRC32 ... successful.
seting root password to "Z1ON0101".
stem open: Access Level <9>
sh 10.2.2.2 -l root
t@10.2.2.2's password:
```

Downsides

- Easily destroy your entire computer if you aren’t careful
- When you delete something, it’s GONE.
- Takes some getting used to, like typing or learning to swim.

NAVIGATION WITH THE TERMINAL

Launching Sublime Text from the command line

```
ln -s "/Applications/Sublime Text.app/Contents/SharedSupport/bin/subl" /usr/local/bin/subl
```

don’t worry I’ll slack it to you

Give it a try!

1. Create a directory on your desktop named bewd_day_one

2. Change into this directory (hint tab completion).

3. Create another directory called class1_command_line_and_git

4. In that directory, Create a file named "pitch.txt" using the touch command.

5. Open "pitch.txt" in Sublime Text from the terminal.

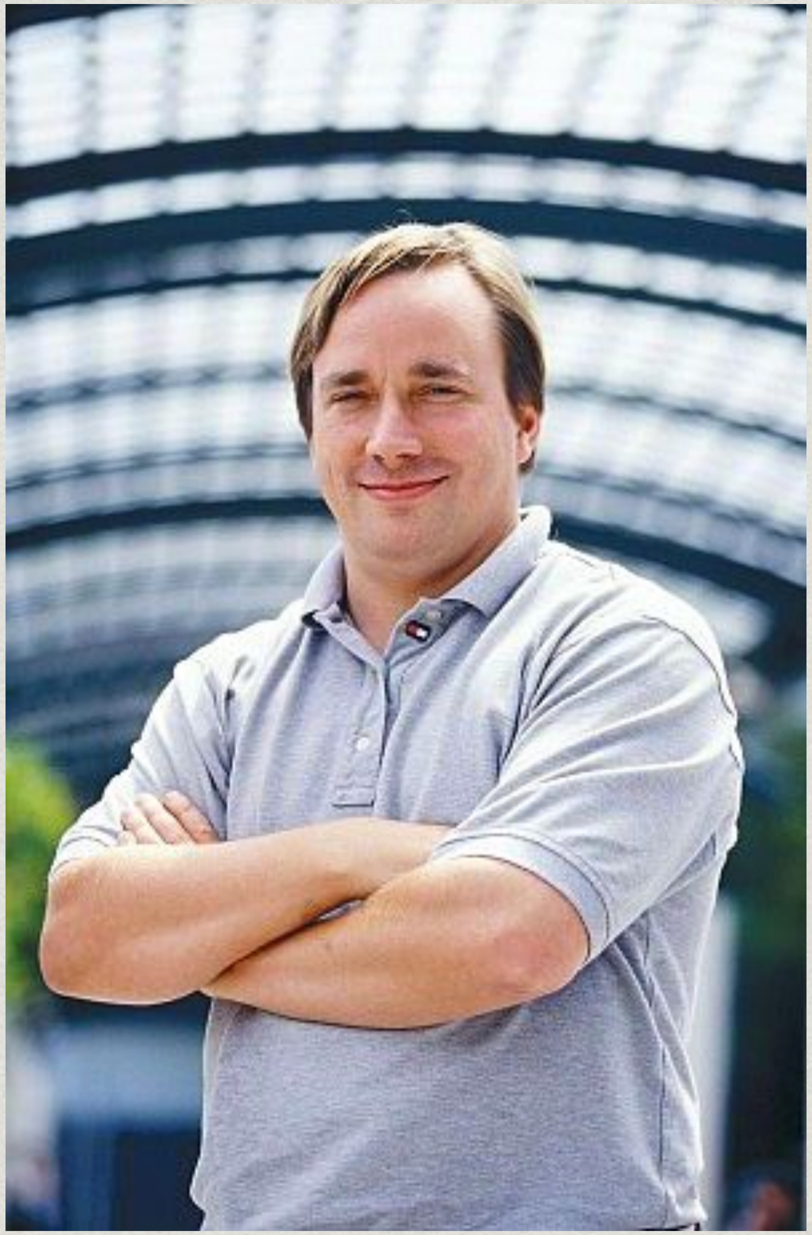
6. Type out your idea for your final project if you have one, if you don't type a general goal you have for the class

GIT AND GITHUB

THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.

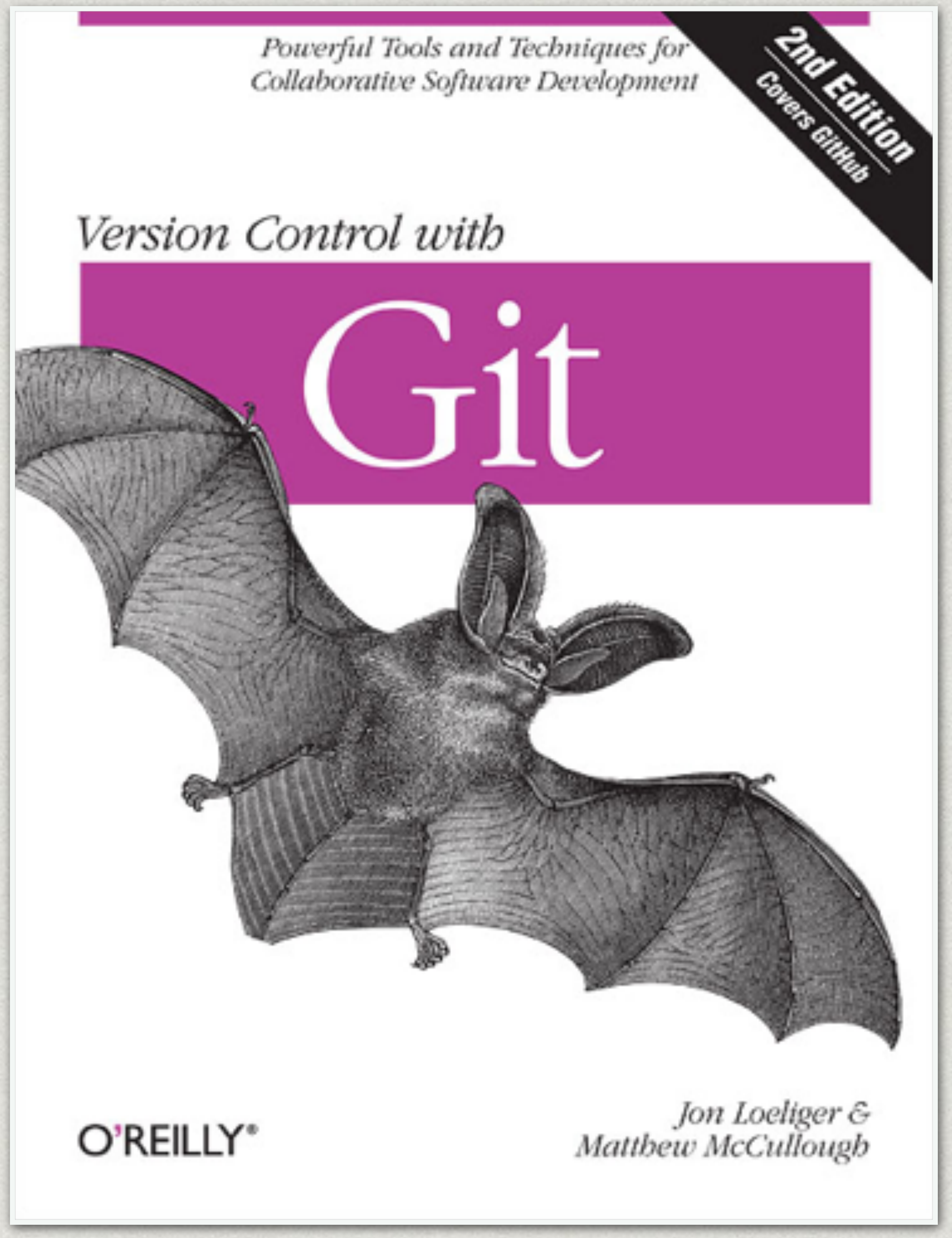
What is Git?

- Git is a source control management tool started by Linus Torvalds.
- Git allows you to store and update your code in a structured way.
- Git includes history of changes you make, so you can create "snapshots" and track your work better over time.



Git can be super tricky

- I mess something up with git roughly twice a week, probably.
- It takes a lot of getting used to. We'll cover the absolute basics today



456 pages!!!

How about Github

- GitHub is a service that lets you host Git repositories in the cloud.
- In other words, they are hosted remotely by GitHub, and can be downloaded from / uploaded to over the internet.
- GitHub allows you to easily distribute code to others by sharing your repository.
- GitHub lets you view your code online easily with a web interface.
- GitHub is free to use as long as you make your code public. (open source)

Setting up git and Github

```
git --version
```

If that doesn’t work, you will need to install Xcode command line tools:

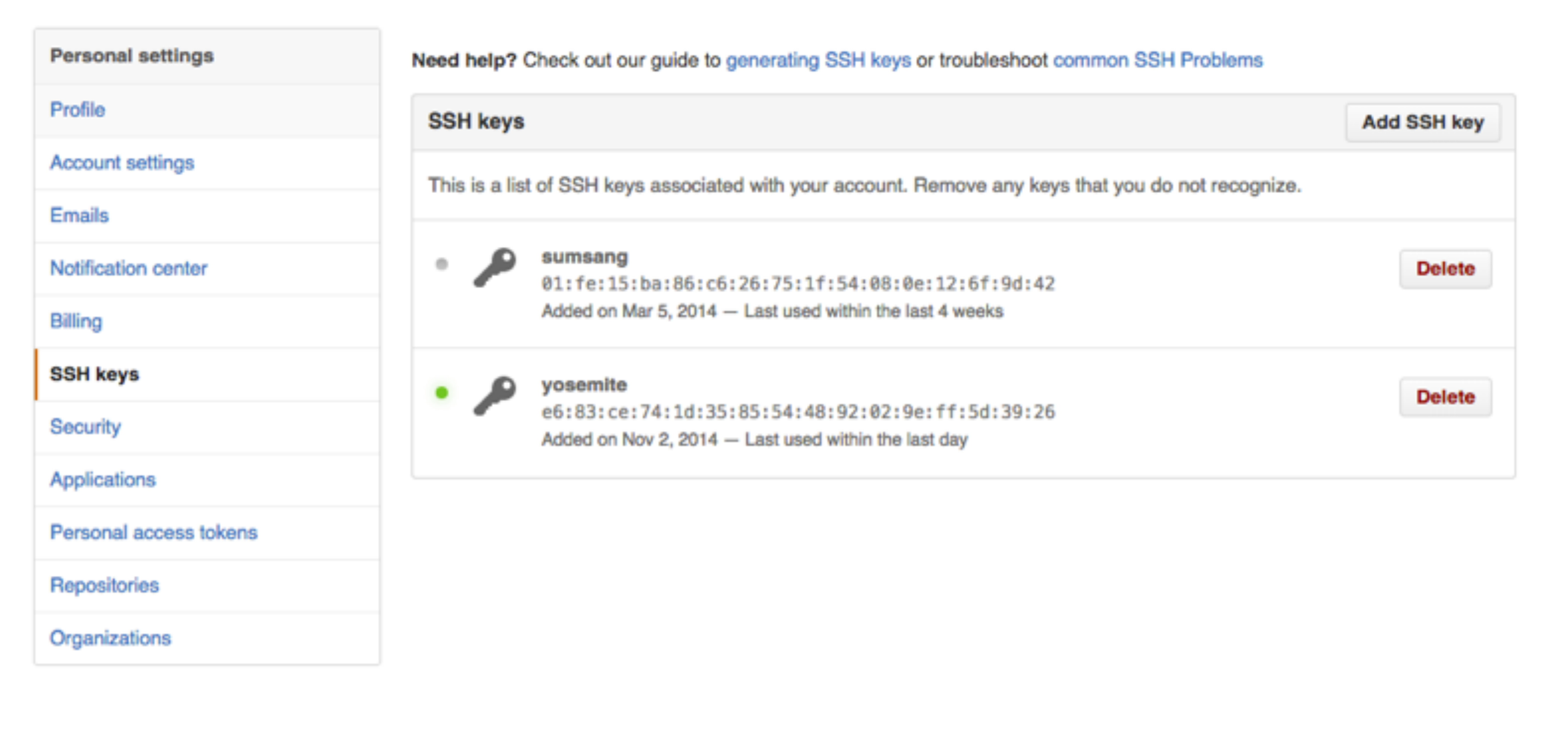
```
xcode-select --install
```

Setting up git and Github

```
git config --global color.ui true

git config --global user.name "YOUR NAME"

git config --global user.email
“theemailyouused@tosignupforgithub.com"

cd ~/.ssh

ssh-keygen -t rsa -C
“theemailyouused@tosignupforgithub"

#don't enter a passphrase
```

Now go to Github

- Hit the menu with your profile photo in the upper right.
- Click settings, then click SSH keys on the left.



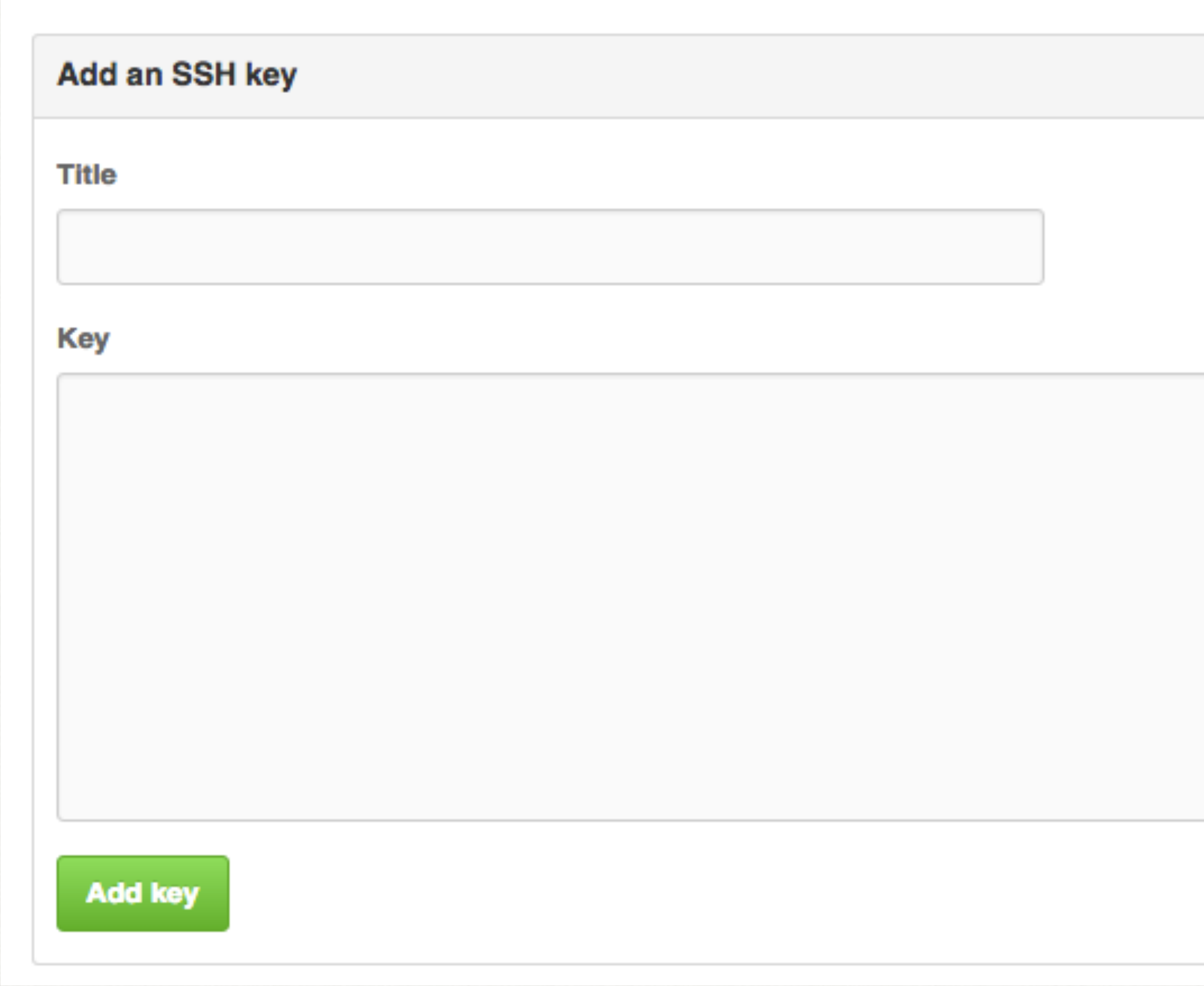
Need help? Check out our guide to generating SSH keys or troubleshoot common SSH Problems

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

sumsang
01:fe:15:ba:86:c6:26:75:1f:54:08:0e:12:6f:9d:42
Added on Mar 5, 2014 — Last used within the last 4 weeks

yosemite
e6:83:ce:74:1d:35:85:54:48:92:02:9e:ff:5d:39:26
Added on Nov 2, 2014 — Last used within the last day

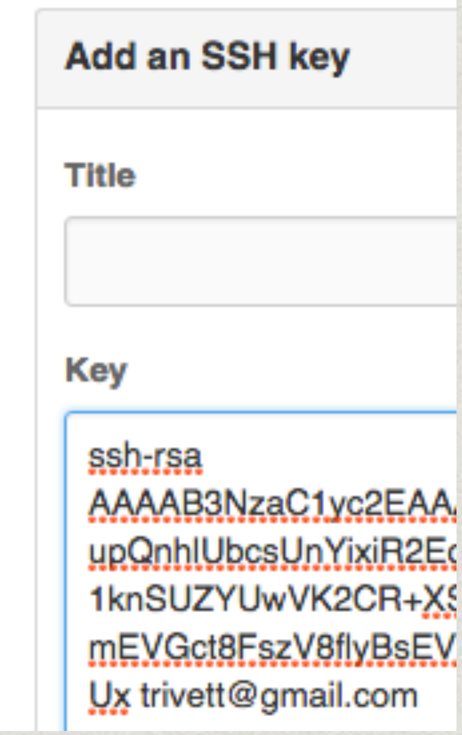
- Click Add SSH Key
- Give your computer a name



Add an SSH key

Back in the Terminal

- Run this to copy your SSH key that you just generated
- `pbcopy < ~/.ssh/id_rsa.pub`
- `(you may have to mkdir ~/.ssh`
- It will start with ssh-rsa, then a bunch of jibberish
- Ends with your email.
- Paste it into the Key field like so



Add an SSH key

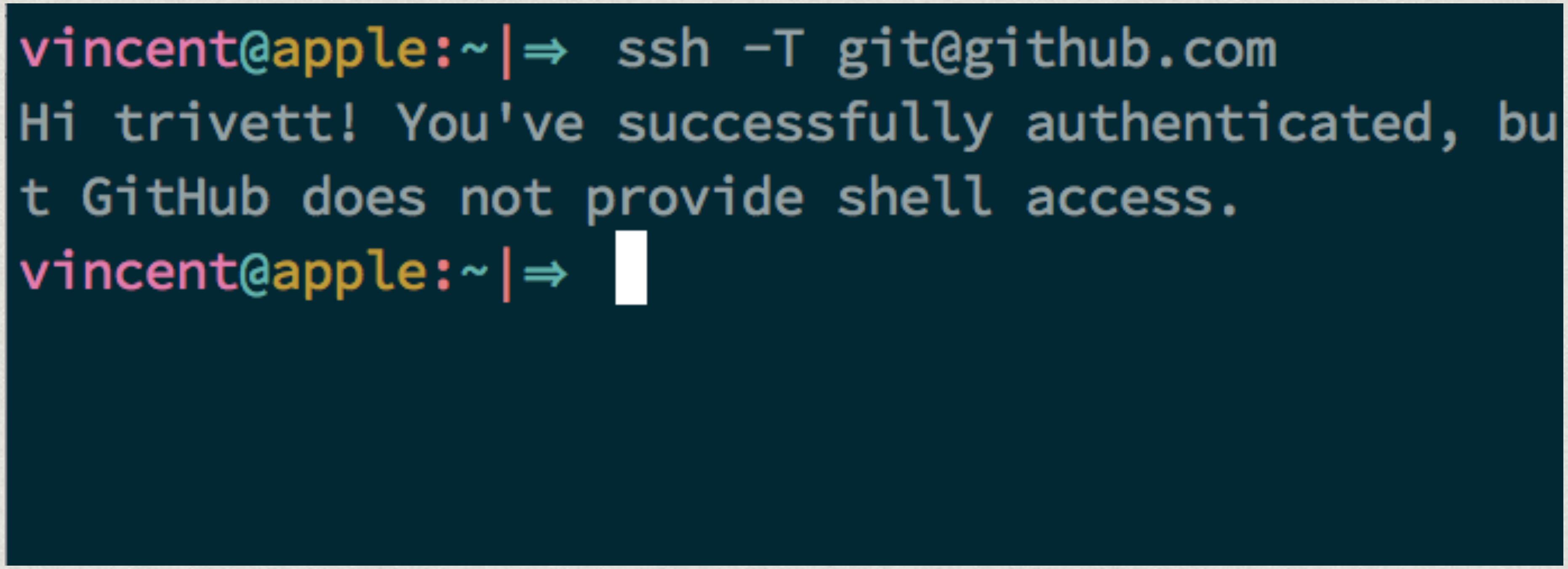
Title

Key

ssh-rsa
AAAAB3NzaC1yc2EAA
upQnhlUbcsUnYixiR2Ed
1knSUZYUwVK2CR+XS
mEVGct8FszV8flyBsEV
Ux trivett@gmail.com

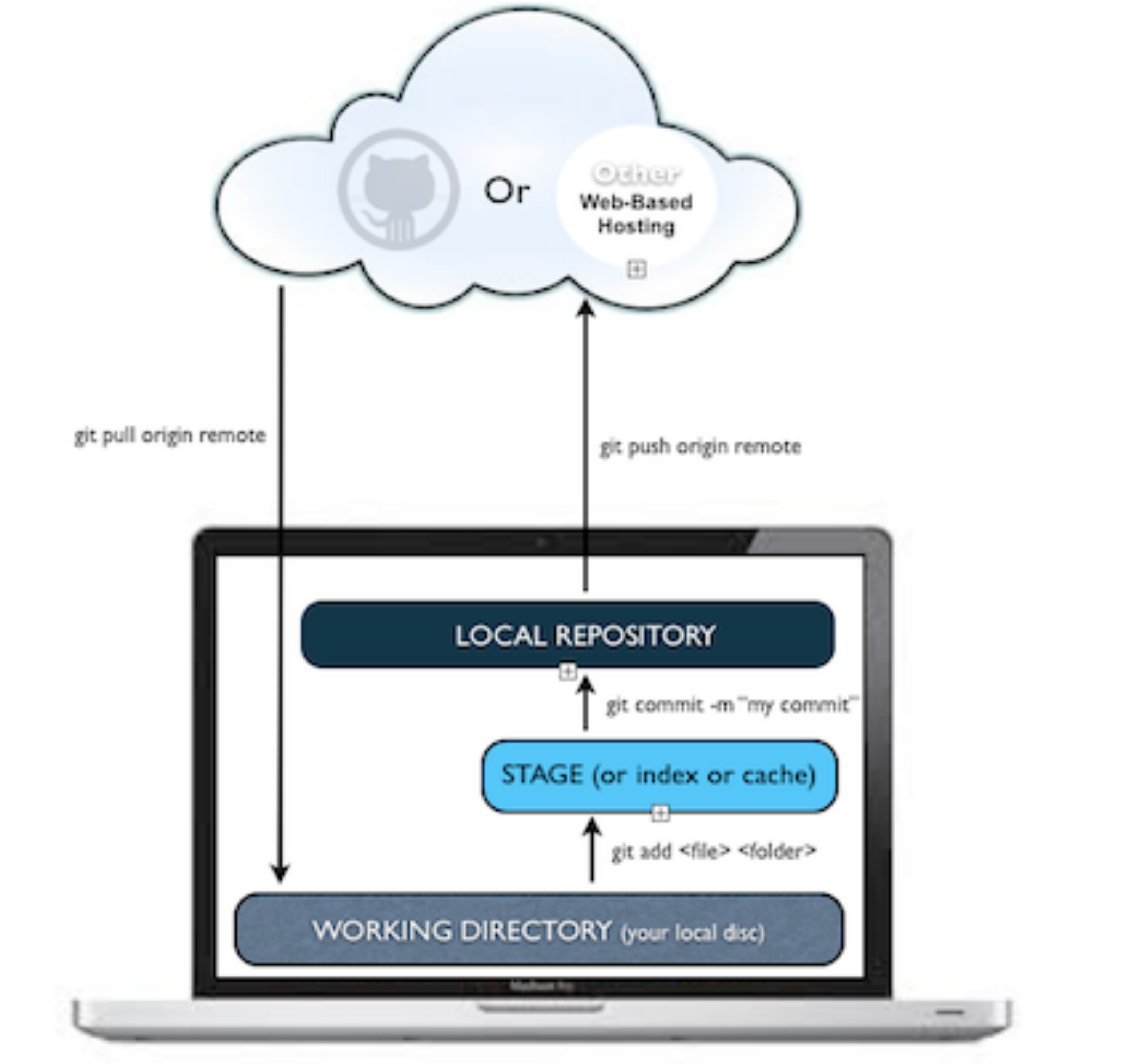
Check if it works

- ssh -T git@github.com



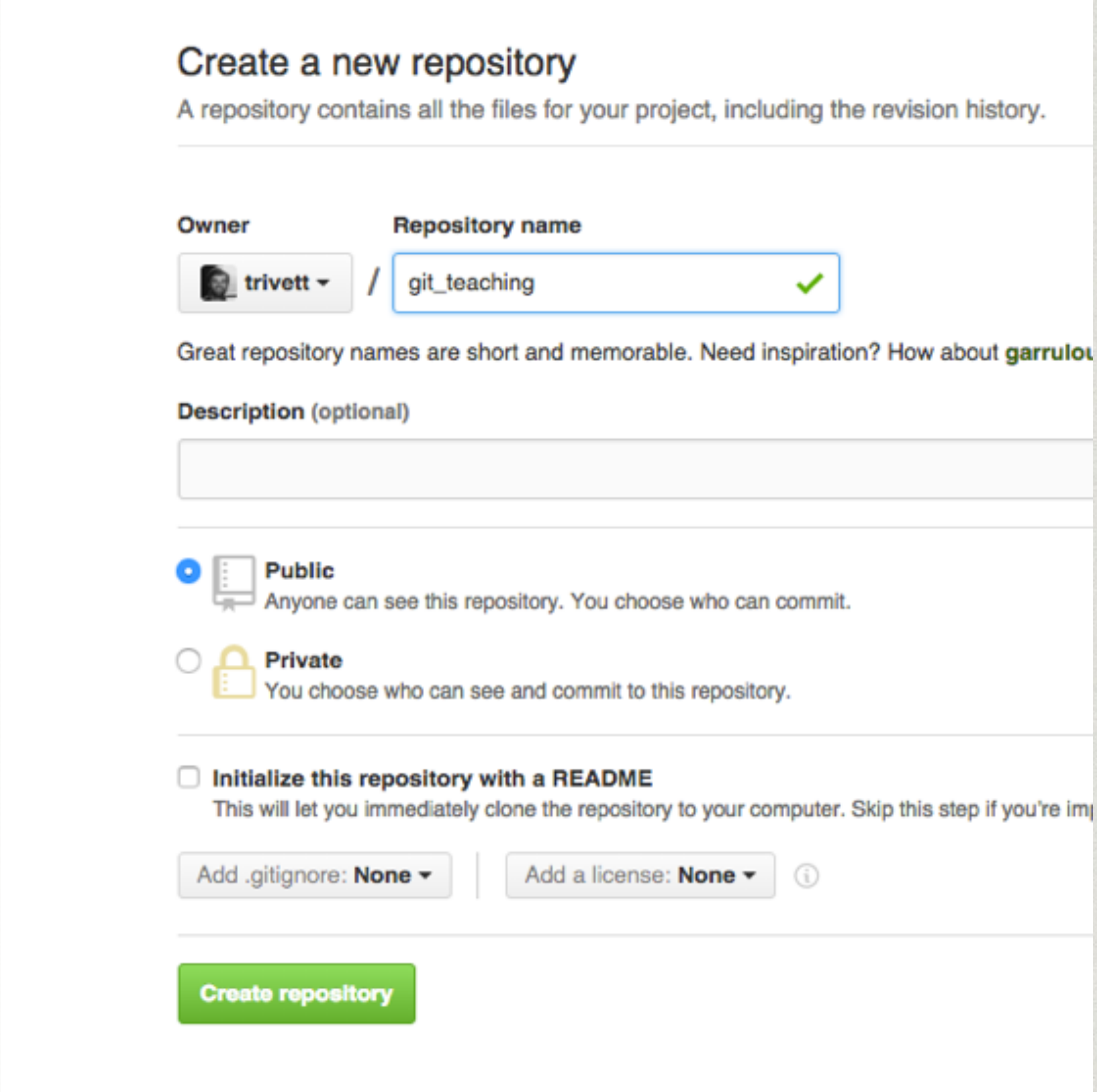
```
vincent@apple:~|⇒  ssh -T git@github.com
Hi trivett! You've successfully authenticated, bu
t GitHub does not provide shell access.
vincent@apple:~|⇒  
```

Git workflow



LET’S DO IT TOGETHER

Make a new repo



Create a new repository

A repository contains all the files for your project, including the revision history.

Owner trivett ▾ / **Repository name** git_teaching ✓

Great repository names are short and memorable. Need inspiration? How about **garrulou**

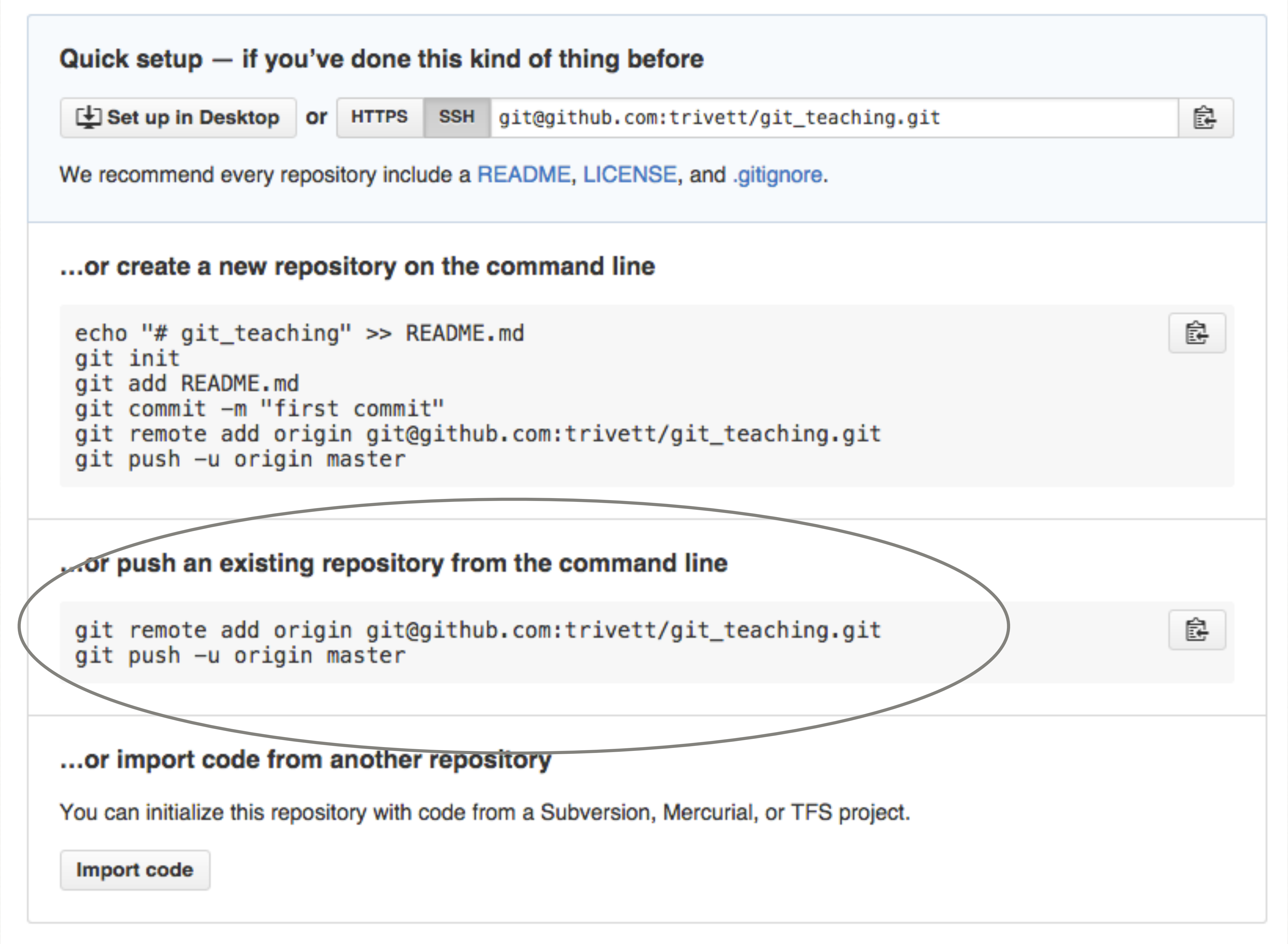
Description (optional)

● **Public**
Anyone can see this repository. You choose who can commit.

○ **Private**
You choose who can see and commit to this repository.

□ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if you're im

Add .gitignore: **None** ▾ | Add a license: **None** ▾



Quick setup — if you've done this kind of thing before

git@github.com:trivett/git_teaching.git

We recommend every repository include a README, LICENSE, and .gitignore.

…or create a new repository on the command line

```
echo "# git_teaching" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin git@github.com:trivett/git_teaching.git
git push -u origin master
```

…or push an existing repository from the command line

```
git remote add origin git@github.com:trivett/git_teaching.git
git push -u origin master
```

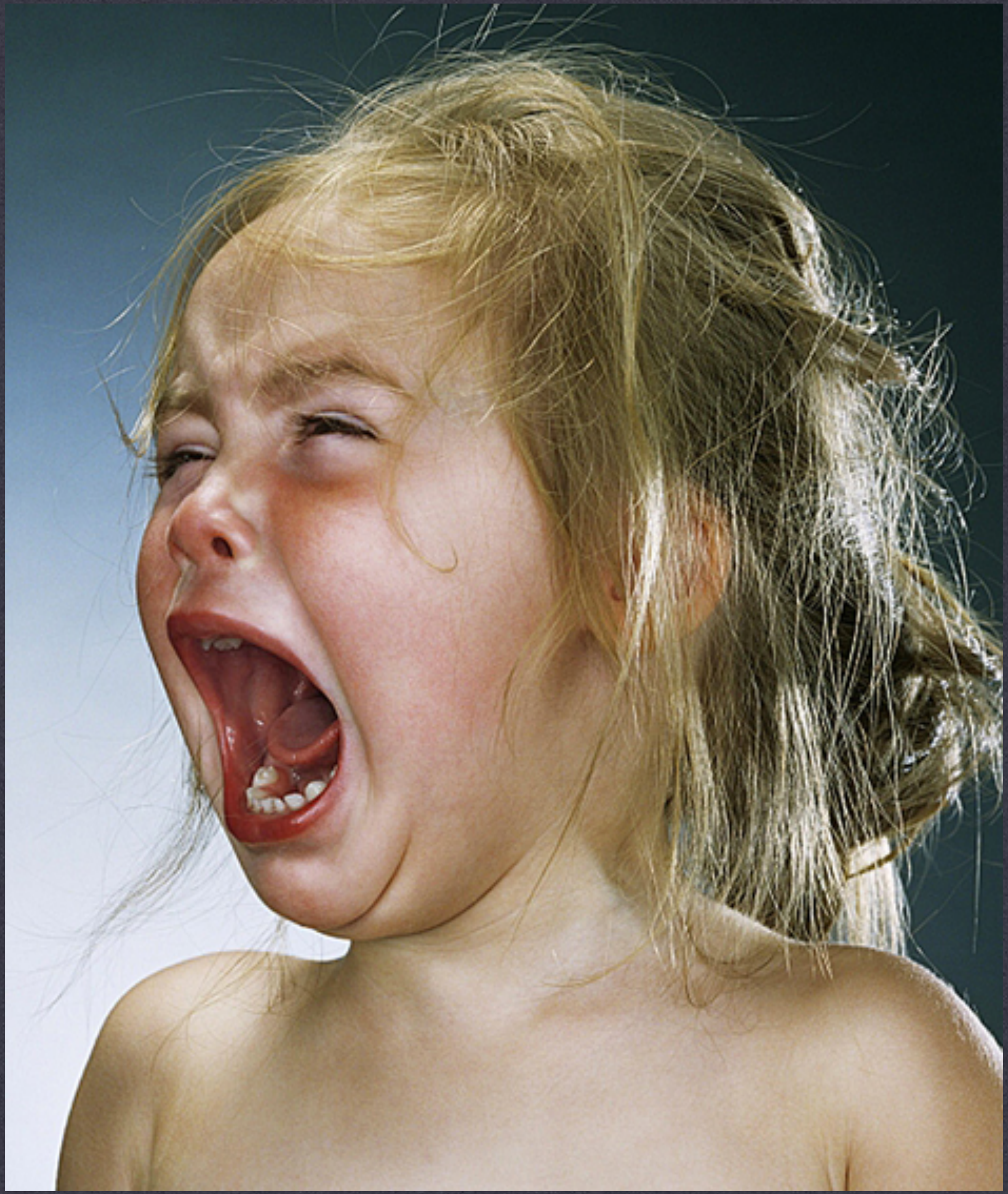
…or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

YOUR TURN!

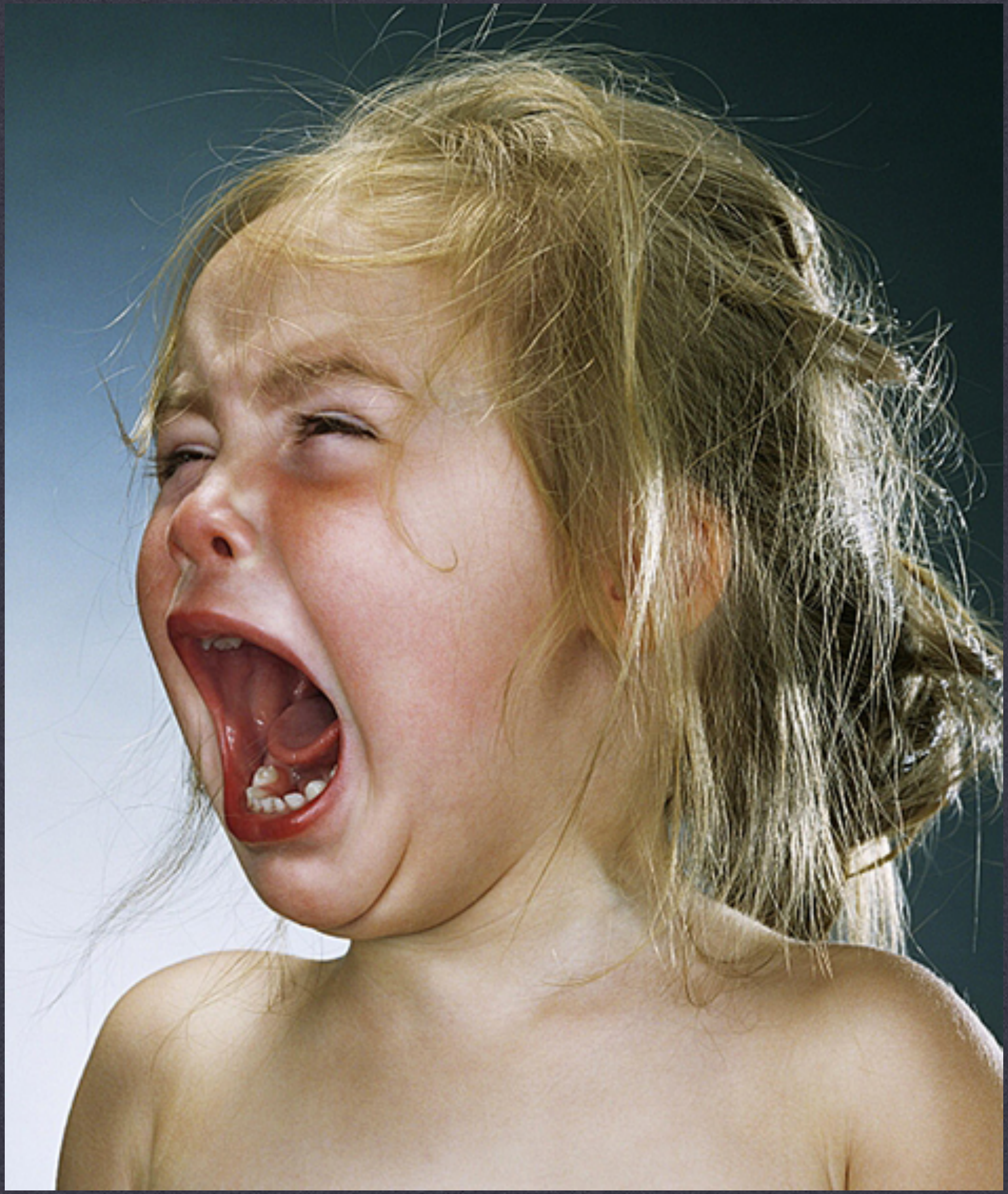
- Your turn.
- Create another file, edit it, add it, commit it, push it up to your repo
- remove sup_git.txt
- add and commit
- push up to github
- Look at the commits tab on the repo on github.
- We will help.

HOMEWORK!



HOMEWORK!

We will make a personalized glossary of terms and commands



https://github.com/trivett/BEWD-NYC/blob/master/assignments/class1/homework1