SSH + GitHub

Learning Objective

- Understand the basic mechanics of SSH technology
- Make the process of pushing to GitHub easier

Why learn SSH?

- You'll be able to communicate securely with servers around the world
- Passwords won't be necessary for secure communication between your computer and any other computer if you use it
- It will save you a lot of time over your development career!

What is SSH?

- SSH is a psedo-acronym that stands for "Secure Shell"
- The technology provides a protocol for secure communication between a client (your computer) and a remote server
- Authentication works through the exchanging of keys, which are simply files
  - A public key given out to trusted servers
  - A private key that lives on your machine to verify the public key against

Using SSH

- In order to use SSH, you'll need to generate a public and private key pair, simply two files
- This pair is almost always stored in the ~/.ssh directory, which will only exist if you've created SSH keys before
- Before going through the key creation process (next slide), you should check if you already have a public and private key by cding to the ~/.ssh directory (if you do have keys, skip to the final step)

```
$ cd ~/.ssh
$ ls
# If you get anything back that ends in `.pub`, you might have ssh keys!
```

Creating your keys

Assuming you don't already have a set of keys, use these commands to generate them

```
$ ssh-keygen -t rsa -C "youremail@example.com"
# Enter file in which to save the key (/Users/username/.ssh/id_rsa): [Press enter]
# Enter passphrase (empty for no passphrase): [Type a passphrase]
# Enter same passphrase again: [Type passphrase again]
```

Assuming all has gone well, you'll receive a message saying that your keys have been saved

Adding your new key to `ssh-agent`

`ssh-agent` is the go-between program that sends keys - you'll want to make it aware of your new set of keys

```
# start the ssh-agent in the background
$ eval "$(ssh-agent -s)"
# Agent pid 34234
# Add your key
$ ssh-add ~/.ssh/id_rsa
```

Final steps: copy your key

Now that you've generated and stored your SSH keys, you'll need to copy your public key - the below instructions assume that you've named it `id_rsa.pub` (the default name)

```
$ pbcopy < ~/.ssh/id_rsa.pub
```

(Windows/Git Bash users can use `cat ~/.ssh/id_rsa.pub` and copy the results)

Final steps: adding keys to your accounts

- If you're adding the copied key to GitHub, just go to Settings -> SSH Keys -> Add SSH key on GitHub.com
- Copy what's on your clipboard into the text area, hit the "Add Key" button, and you should be good to go
- Now you can use the SSH URL of a repository instead of the HTTPS url, **which avoids the necessity to enter your password each time you clone a repository**! Nice.

Exercise

- Create a new `git` repository locally and then create a new repository on GitHub.com
- Copy the SSH URL from your new repository on GitHub.com
- Add this URL as a remote to your repository locally

```
git remote add origin git@github.com:yourname/yourrepo.git
```

- Make sure you can successfully push to that repository

```
git push origin master
```

- Finished? Try it again to make sure you have it down!

Quiz

1. How does authentication work in SSH?

2. How do you add a key to your GitHub account?