Assignment #5 - Get Yo' Tools Installed (Mac)

Overview

In this assignment, you will be installing all of the required tools and software necessary for the class. We've got a lot to install so buckle in and get ready!

Before You Begin

Make sure you sign up for these services; you'll need all of them throughout the course.

- LinkedIn: https://www.linkedin.com
- GitHub: https://github.com
- StackOverflow: http://stackoverflow.com

Don't just create logins. Job recruiters often scour these sites in search of job candidates; make sure you provide your headshot and your contact info on all three services.

P.S. Don’t forget to outline your skills and your work experience on LinkedIn.

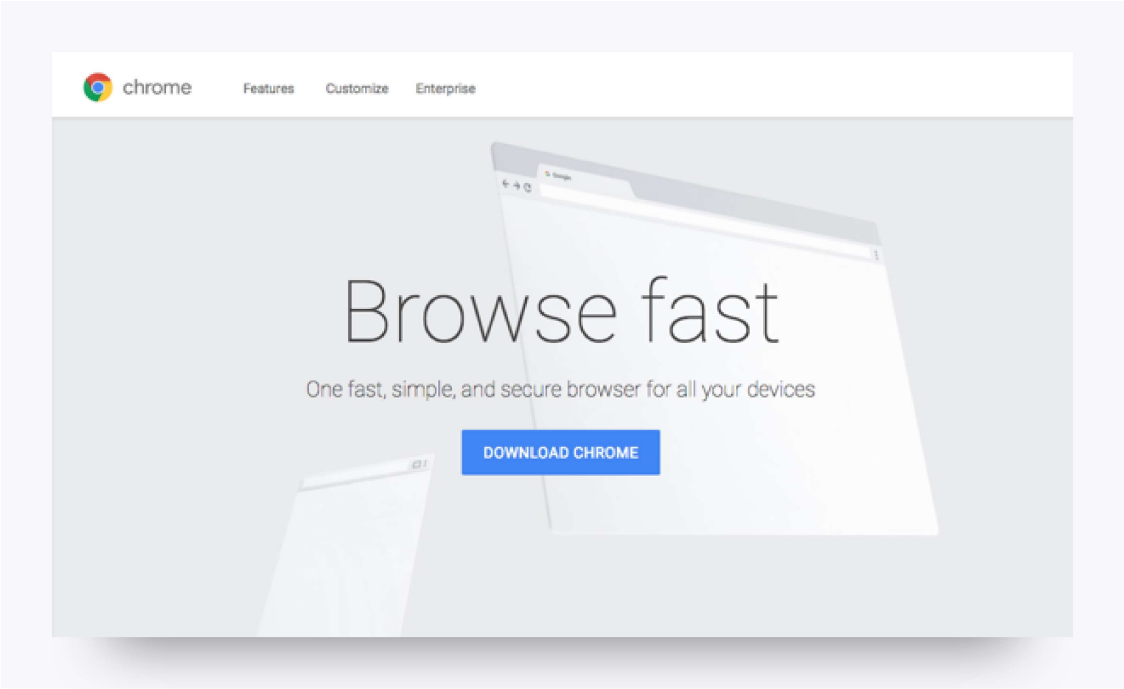
Also, make sure to accept the invite for your section on Slack as well. You will receive the link to your class-specific channel during orientation.

Da Big Installation Enchilada

The rest of this assignment will walk you through the specific steps associated with installing each of the tools you'll need. Follow the instructions closely!

Google Chrome

During this course, consider Chrome *the* web browser; it comes loaded with tools for quickly editing the web pages you’ll create.



1. If you don't already have Chrome installed, visit their download page.
2. Download, open, and run through Chrome's installation file.

Screencastify

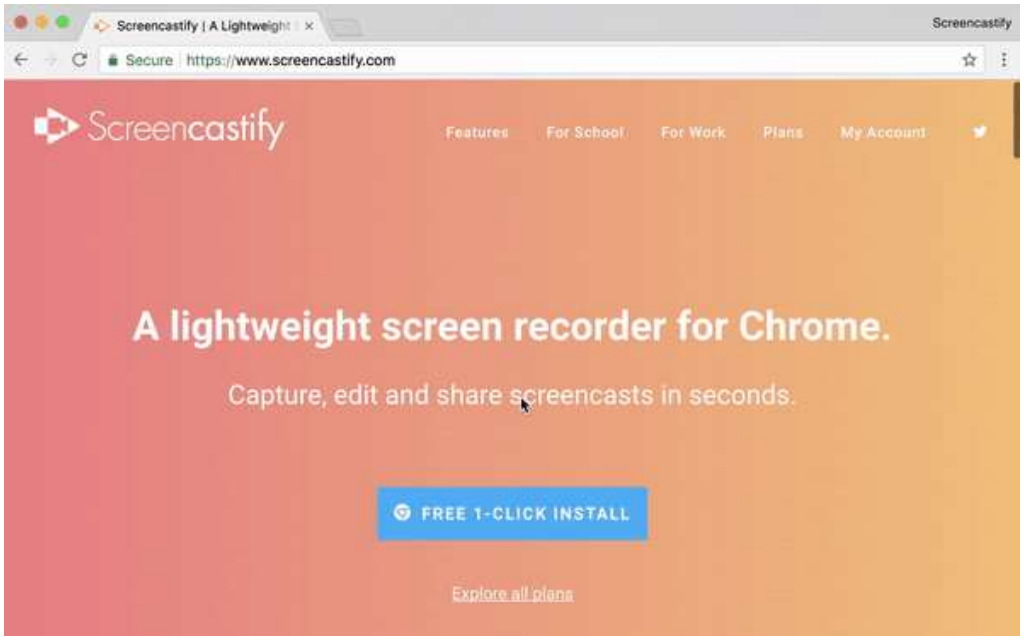
There are two places where you can install the Screencastify extension:

1. The Screencastify Website.
2. The Chrome Web Store.

Note: You must be using Google Chrome to install and use Screencastify

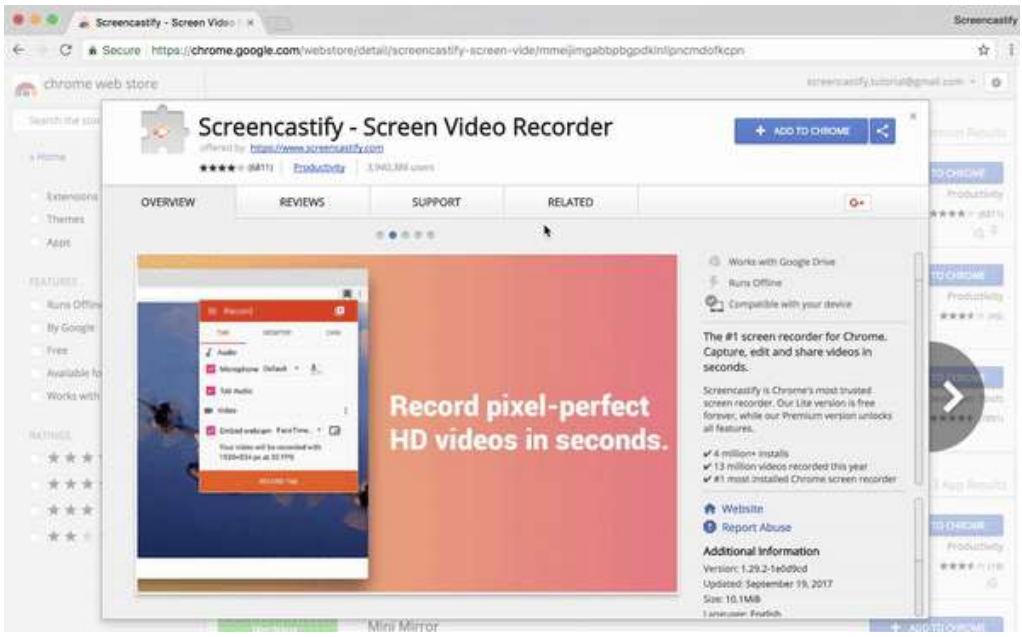
Install Screencastify from their Website:

- Go to the homepage and click the blue "**Free 1-Click Install**" button. You'll see a confirmation box confirming that you want to install the extension. Click "**Add extension**". Screencastify will then begin downloading.

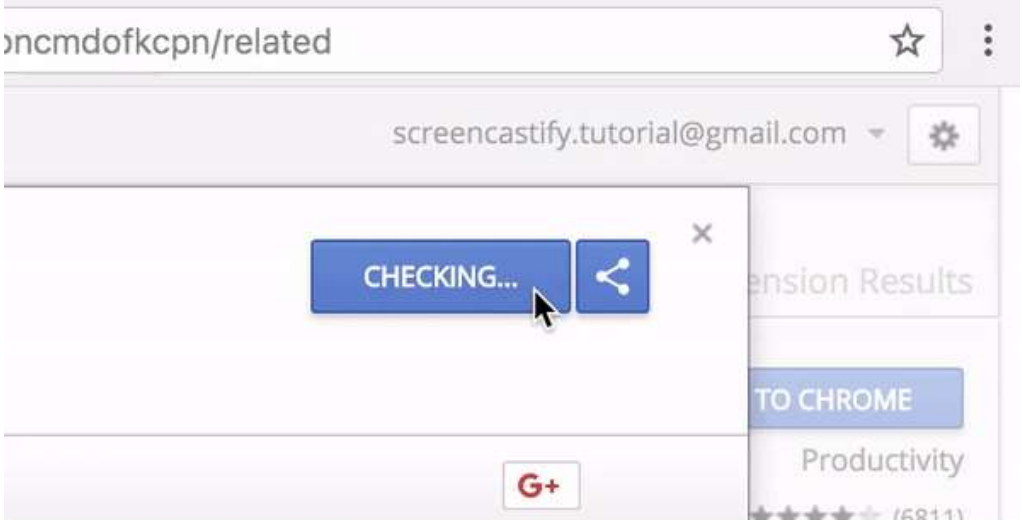


Install Screencastify from the Chrome Web Store:

- Click here to view their listing in the Chrome Web Store, then click "**Add to Chrome**". You'll see a confirmation box confirming that you want to install the extension. Click "**Add extension**". Screencastify will then begin downloading.



- Once Screencastify is finished recording, their grey film strip icon will appear in the top right-hand corner of your browser. Click the icon to get started.

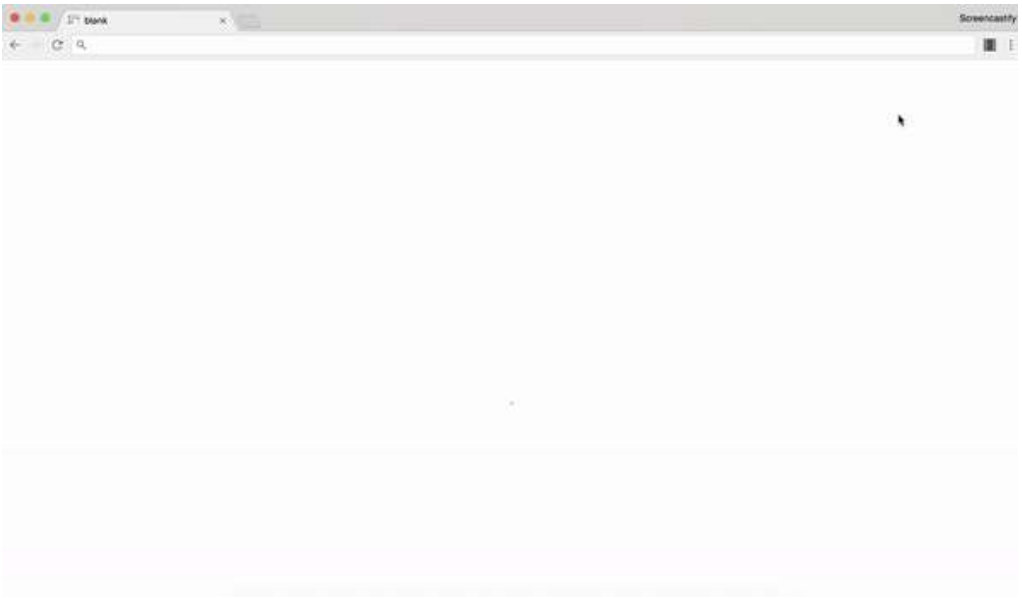


Set up your webcam and microphone:

The first time you click on the Screencastify extension icon, you'll be asked to give Screencastify permission to access your webcam and microphone. They ask for these permissions so that you're able to narrate over your recordings and embed / record your webcam.

Note: While you're required to allow access to your camera and microphone in order for Screencastify to function properly, you don't need to include audio or webcam video in your recordings. Screencastify never records audio or video unless you expressly ask it to.

- When you click on the grey film strip icon, you'll be taken to a page where you can allow camera and microphone access.
- Click the "**Setup Camera Access**" button. Chrome will confirm once more that you'd like to allow microphone and camera access. Click "**allow**".



- And that's it! From now on, you'll be able to narrate over your recordings and record your webcam whenever you'd like.

Troubleshooting: If Screencastify is unable to detect your microphone, click here. If it's unable to detect your webcam, click here.

Connect your Google Drive to Screencastify:

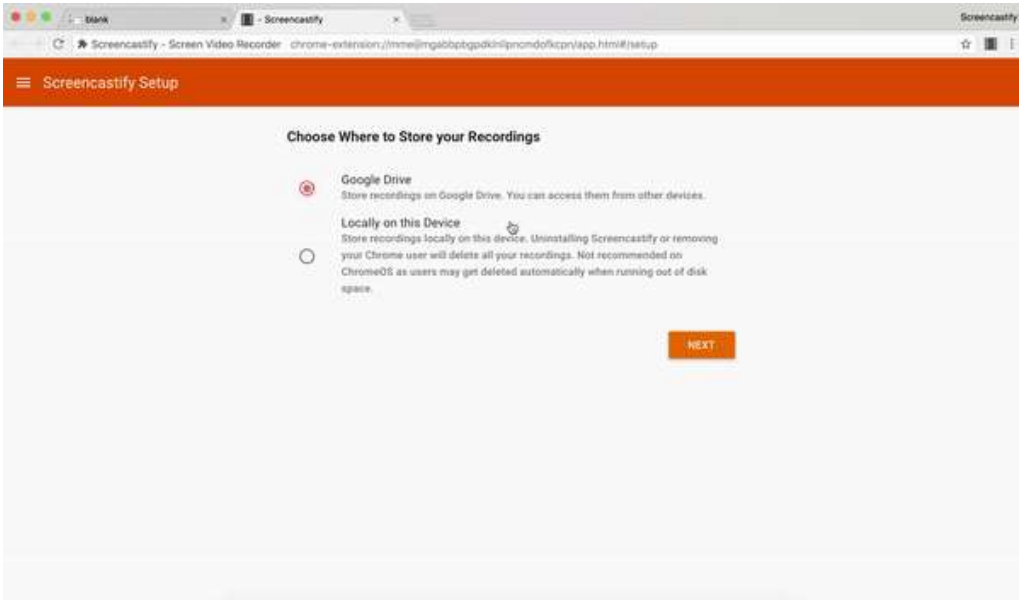
When you connect your Google Drive to Screencastify, all of your recordings will automatically save to your Google Drive in a folder titled " Screencastify". Google Drive, along with many other useful apps come free with a Google Account. If you don't already have one, go create one!

Depending on how long a particular recording is, this process could take a few minutes to complete after you end a recording.

Note: Screencastify highly recommends connecting your Google Drive to Screencastify and saving all of your recordings there vs. locally on your computer. It's much safer.

The first time you open the Screencastify extension, you'll be asked where you want to store your recordings (right after you allow camera and microphone access).

- Make sure that "**Google Drive**" is selected, then click "**Next**". You'll then need to sign in with your Google account and allow Screencastify permission to access your Google Drive. You can sign in with **any** Google-based account, not just a @gmail.com email.



Now, all of your videos will automatically save to your Drive after you finish recording.

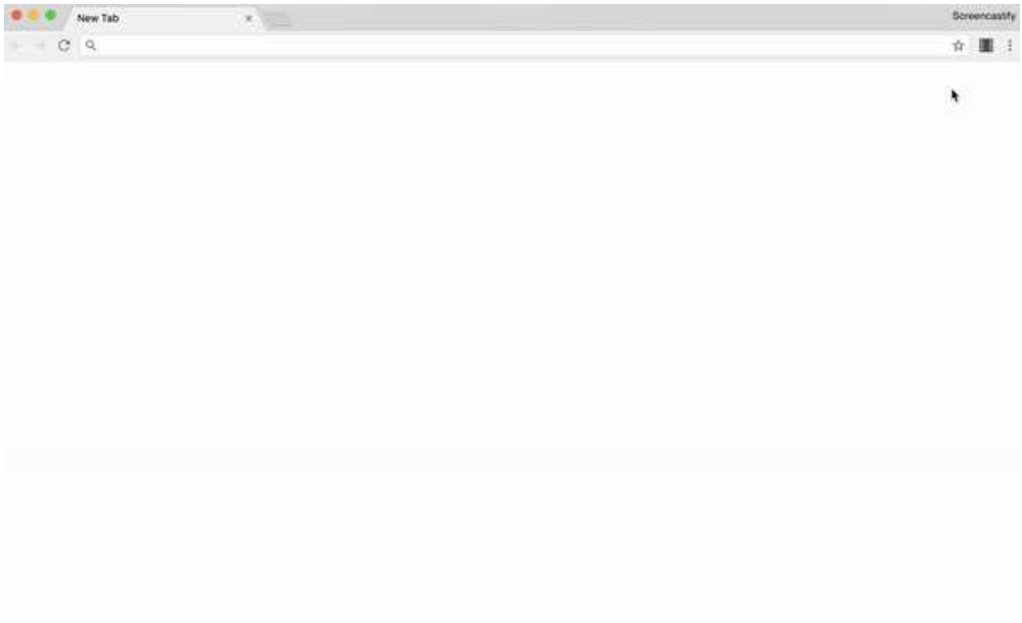
As soon as you finish your first recording, a new " Screencastify" folder will be created in your Drive. All of your recordings will be saved in this folder.

The videos you will see in Your Recordings library in Screencastify are synced with this "Screencastify" folder in your Drive. This means:

- If you delete a video in Screencastify, it will also be deleted from your Drive
- If you move a video out of the "Screencastify" folder in your Drive, it will no longer be accessible from Your Recordings in Screencastify
- If you rename a video in Screencastify, it will also be renamed in your Drive

Note: Screencastify never views, modifies, stores, or in any way interacts with the other files in your Google Drive.

You can always change the default storage location for your recordings, even after initial setup. Simply open the extension menu and click "**Options**". The first thing you'll see is an option to select where you want to store your recordings.

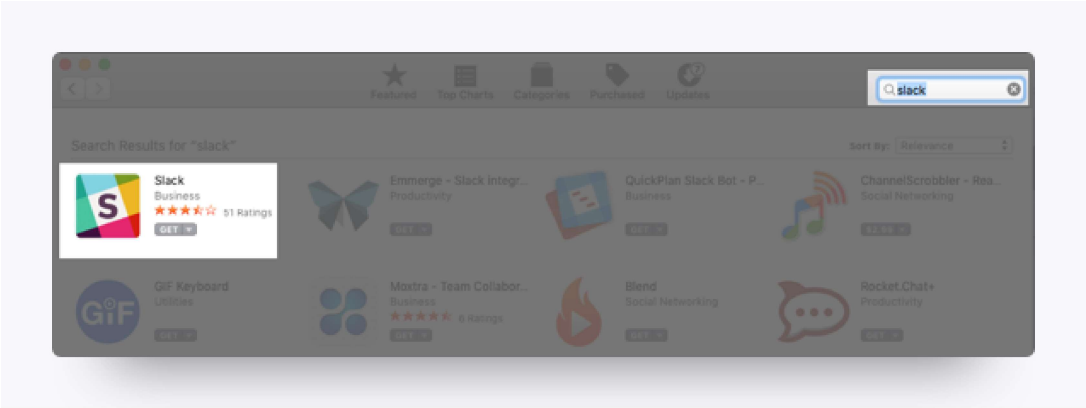


Important Resource: Want to know how to record, save to your Google Drive and share your recordings? Check it out here!

Slack

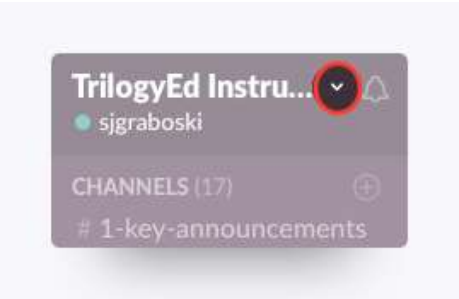
You'll be messaging your instructor, your TAs, and your fellow classmates with this business-centric chatting app. The teaching and career staff will post some of their most important announcements here, so set this program up as soon as you can. You will receive the link to your class-specific channel during orientation.

1. **If you don't have the Slack app yet,** search for Slack in your Mac's App store, and then click the Get button under the app's listing. Click the button again when it displays "install."

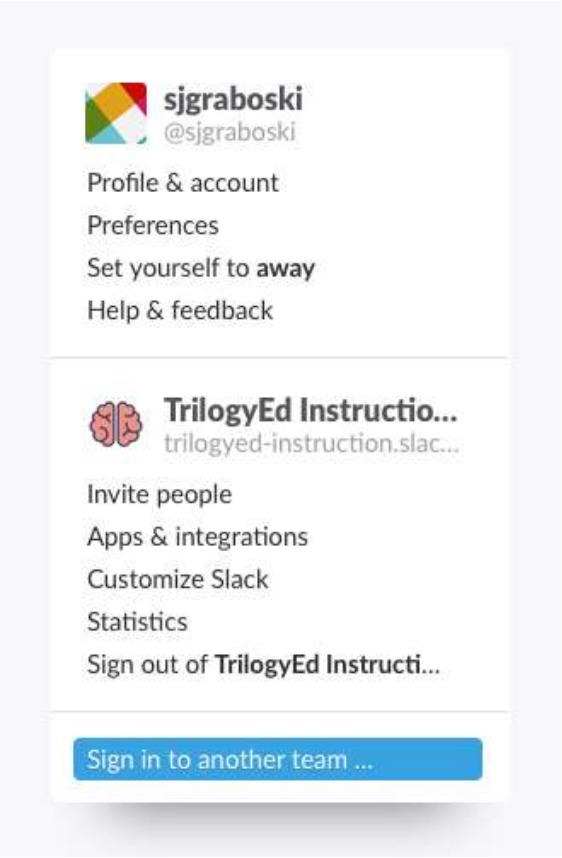


- When the app finishes installing, open it and move on to step 2.

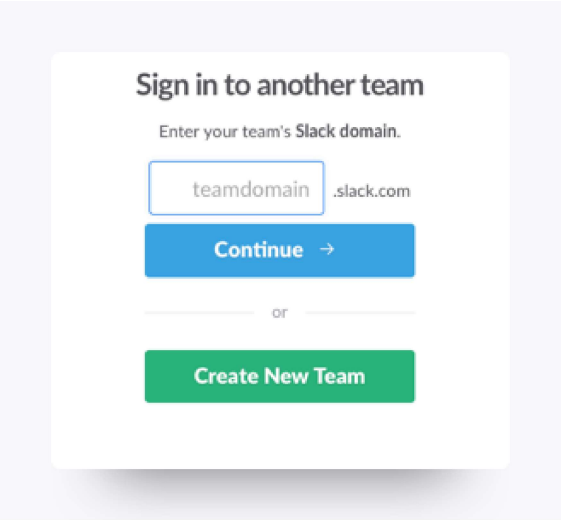
2. **If you already use the Slack app,** you just need to add our channel to your application.
   - Click the header of your current Slack Channel.



   - Then select "Sign in to another team …"



3. As you run through the guide, make sure you do the following:
   - Enter in the domain we gave you for Slack.

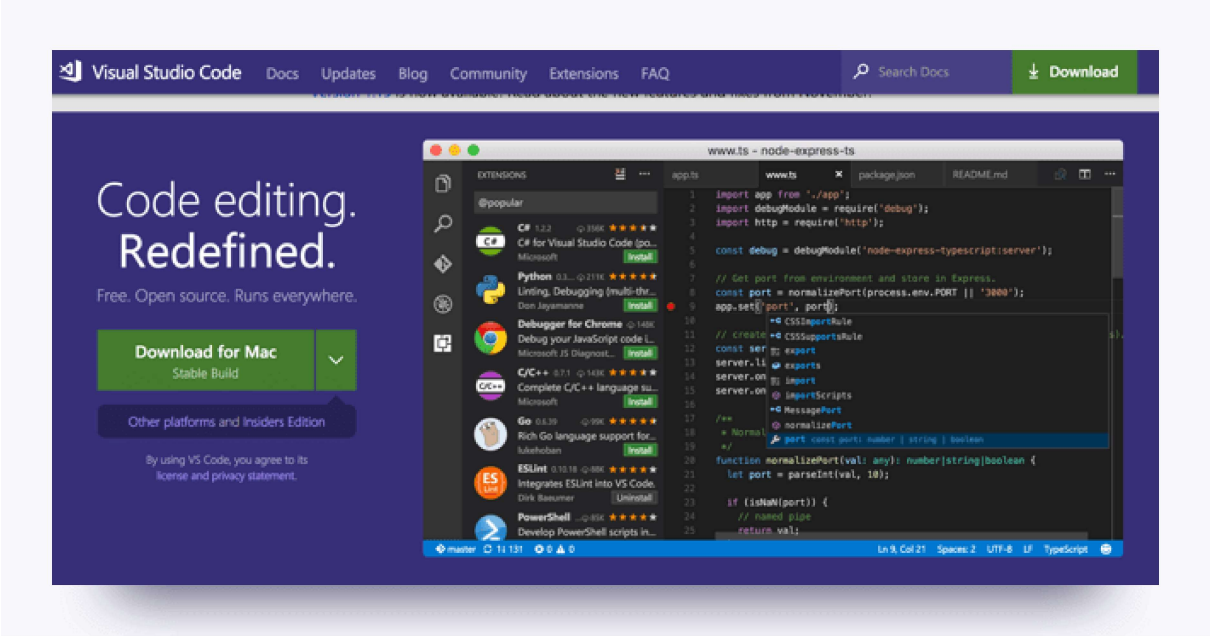


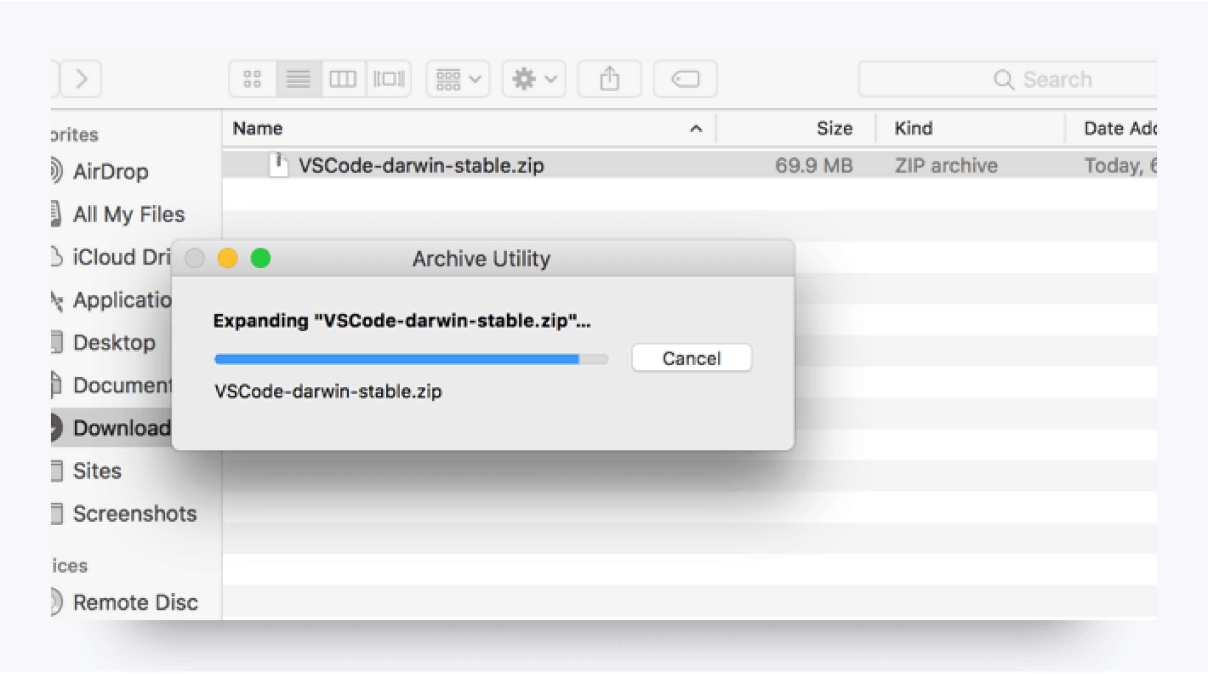
- Enter in the email with which we invited you, as well as your password, when prompted.

Visual Studio Code

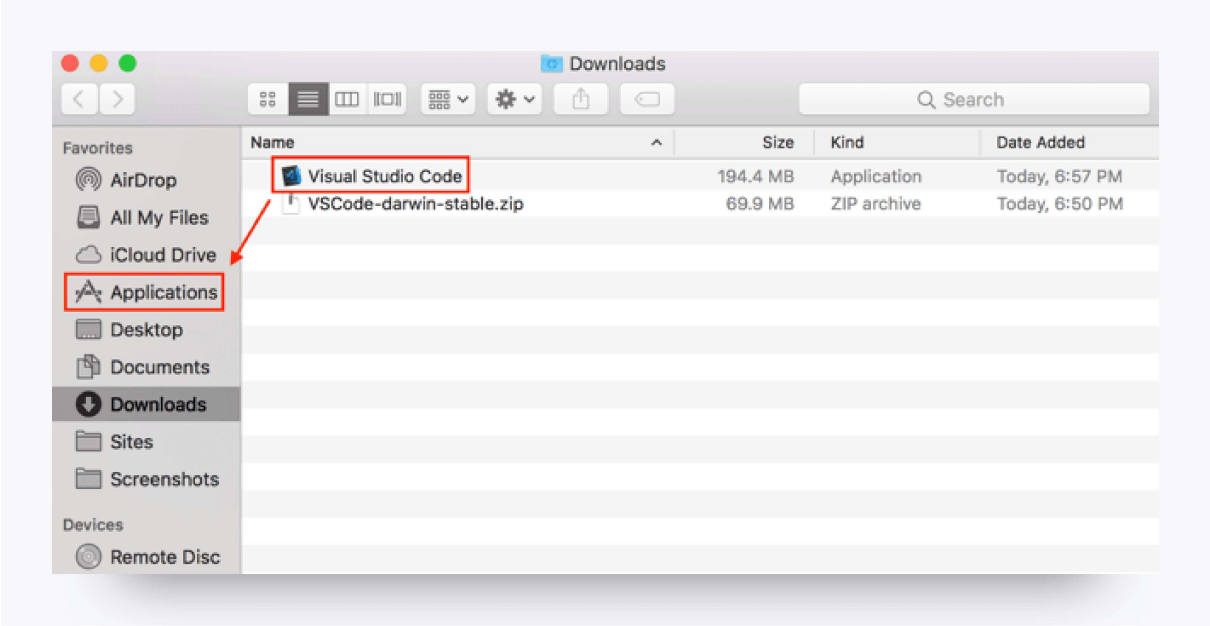
This text editor features a variety of indispensable tools and customizable components that will help you code your websites and apps.

1. Download the appropriate version for your operating system.

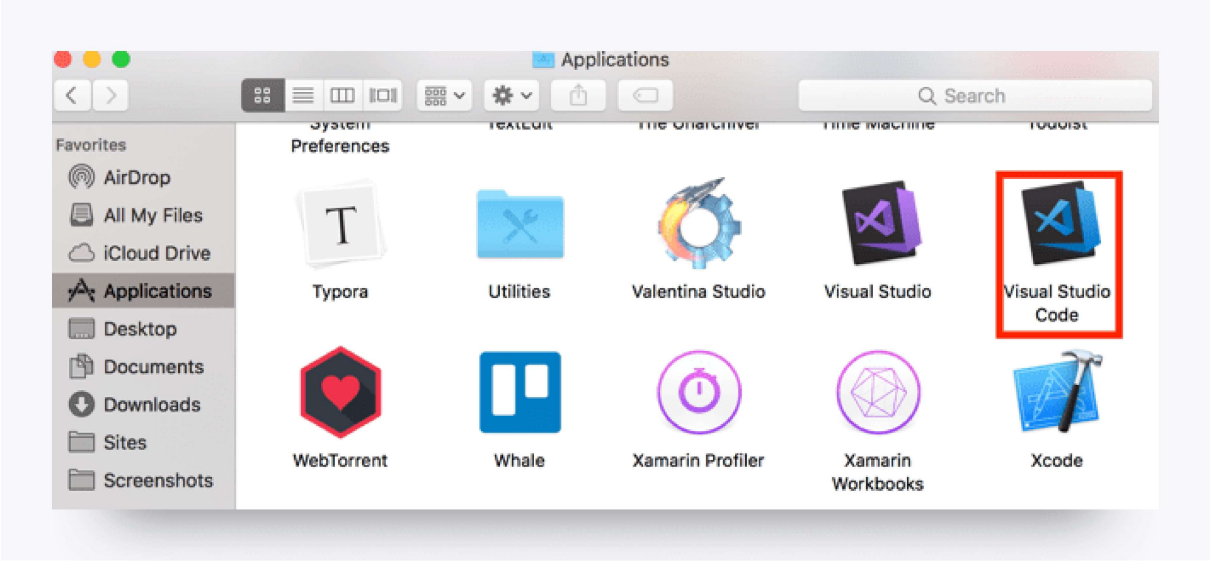




2. Unzip the file.
3. Drag the resulting file to your Applications folder.



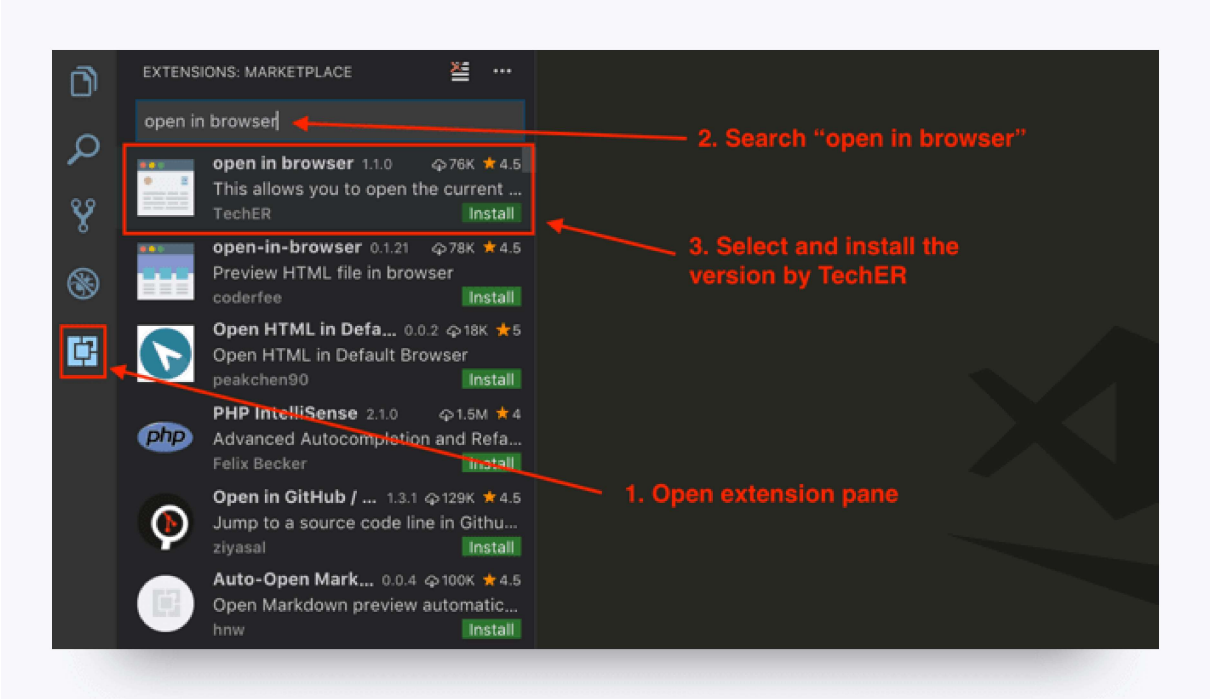
4. You should now be able to access Visual Studio Code from your Applications folder.



Install "Open in Browser" Extension

With this extension, you can open HTML files in your web browser from the Visual Studio Code editor.

1. Open Visual Studio Code.
2. Open the extensions pane and search for "Open in Browser".
3. Select the version written by "TechER" and install.

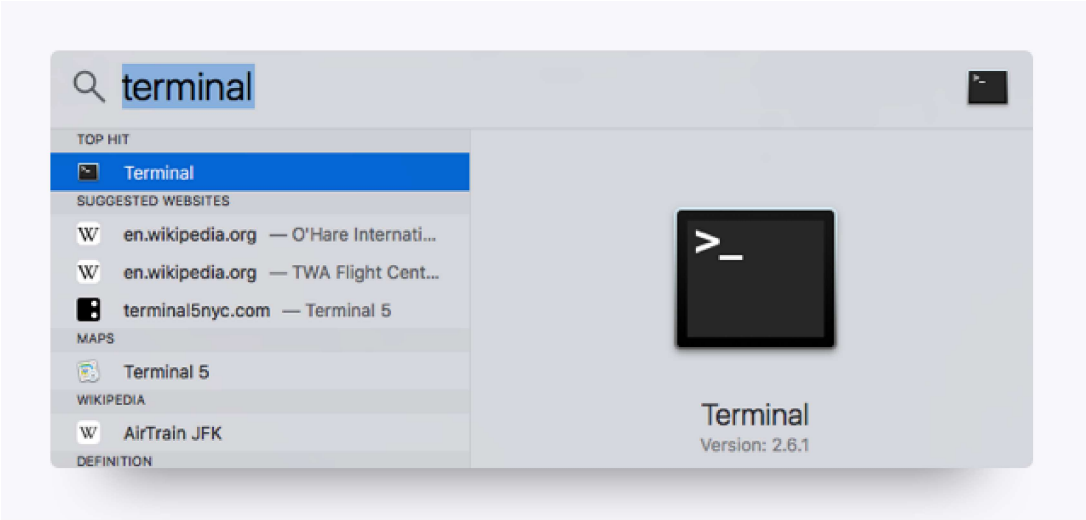


Terminal

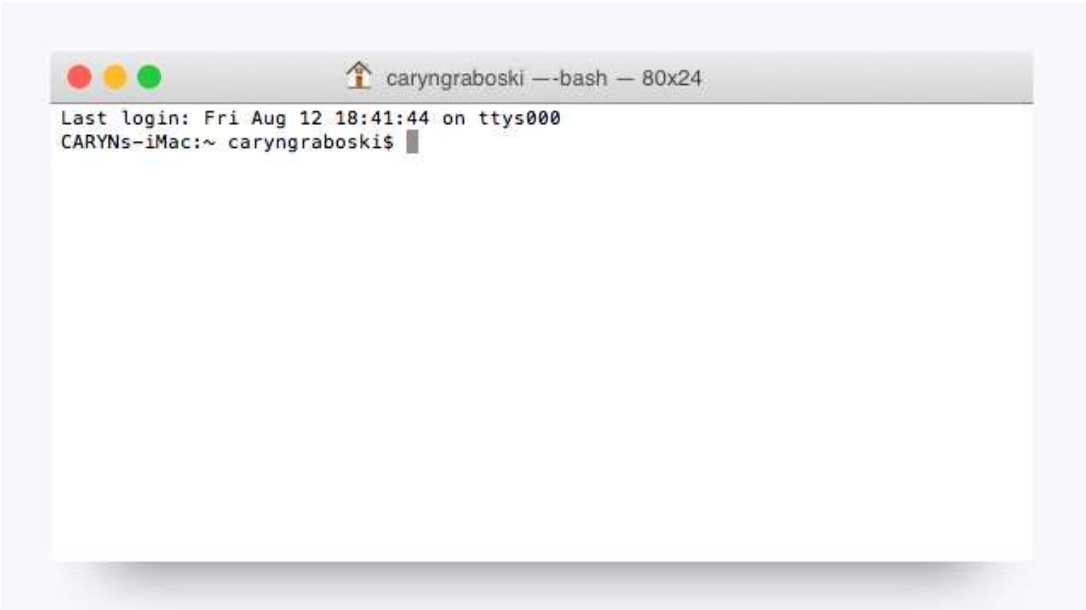
You'll be entering your command line code through this interface. Since you're on a Mac right now, you already have it! Just follow these steps to open the program.

1. Press `command` + `spacebar` to open up Spotlight Search.

2. Type "Terminal" into the search, and then hit enter.



3. Keep this window open; you'll need it for the next steps.



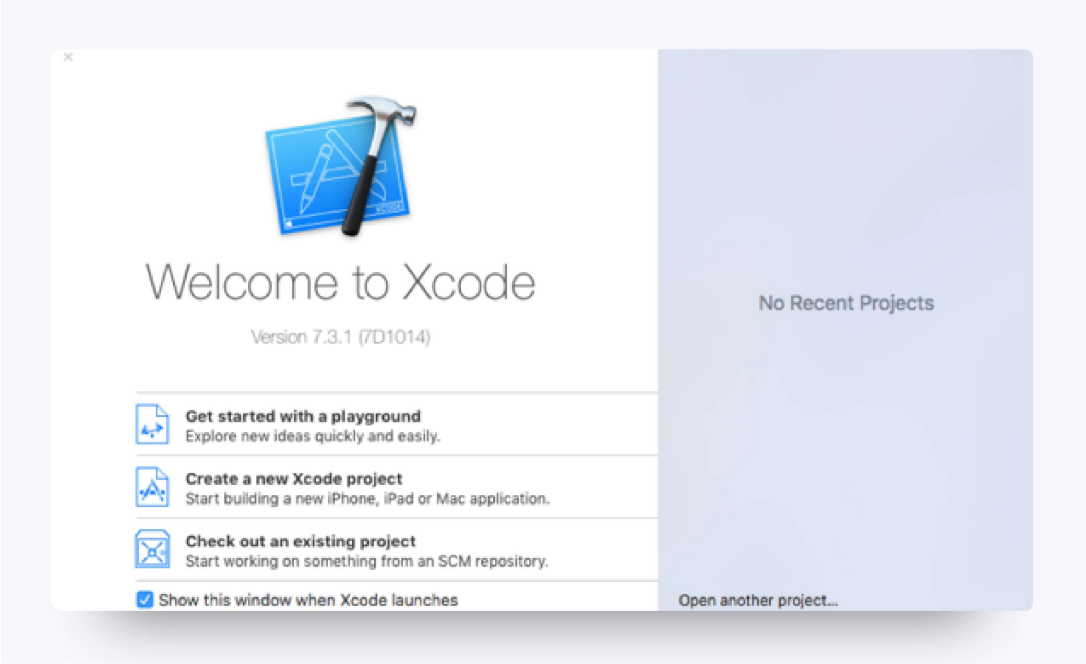
Xcode

Xcode is a development suite exclusive to the Mac. While we're going to use Visual Studio Code in this class, installing Xcode will set up some of the other required bootcamp programs. This includes the always-handy Git—coders depend on this tool for logging the development of programs and applications.

1. Open up the App Store on your Mac machine, and then search for Xcode. Download and install it.

2. Once Xcode finishes downloading, open the program. Agree to the terms of service and wait for the remaining components to install.

3. When you see the window below, you're done! Move onto the Heroku Toolbelt.



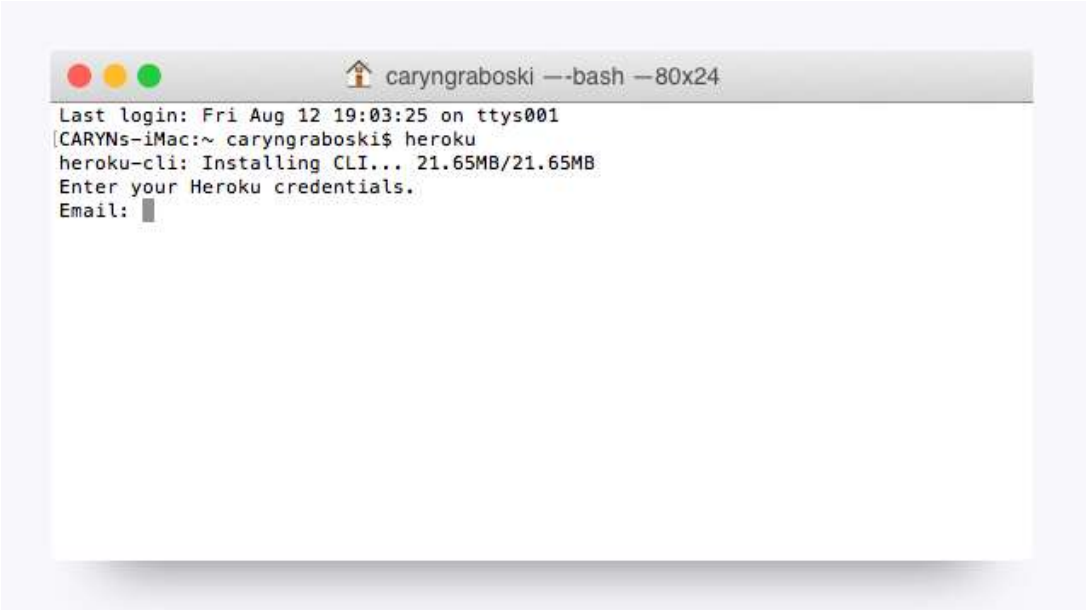
Heroku Toolbelt

This tool lets developers deploy their web apps to the cloud, allowing anyone with the right addresses to access their creations.

1. First, you need to sign up for a free Heroku account: https://signup.heroku.com/.

2. Then go to https://toolbelt.heroku.com. Download the installer.

3. Go through the install guide. Then open Terminal. Type `heroku` into the command line, and then press `return`. If `heroku` doesn't work for you, type `heroku login` instead.



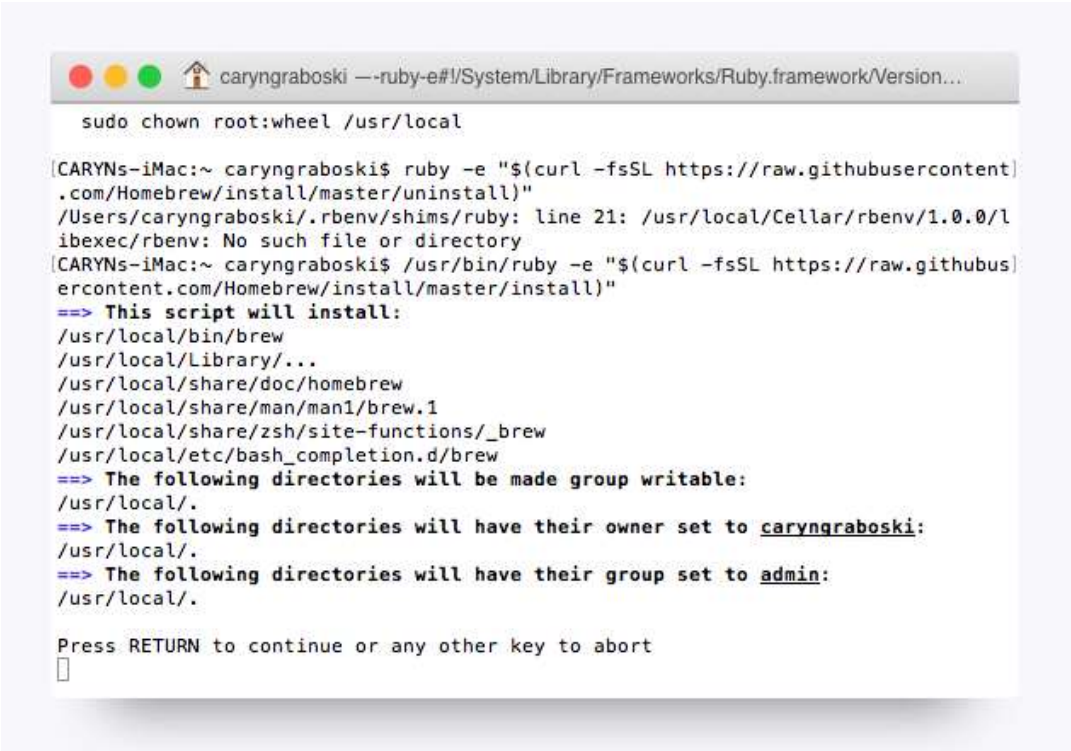
4. When prompted, enter the credentials you used when you signed up for your Heroku account.

Homebrew

This tool makes it a cinch to install new programs and libraries in your Terminal window.

1. Go to http://brew.sh. Copy the script listed under “Install Homebrew.”

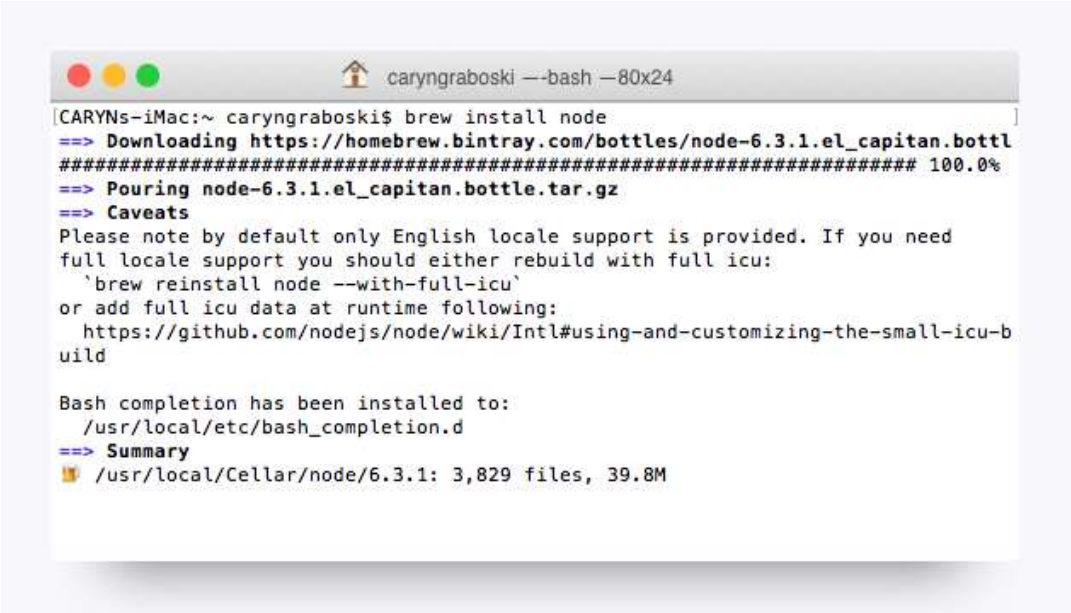
2. Paste the script into your Terminal window. Press the `return` key when prompted.



Node.js:

This runtime environment has quickly become a standard for coding back-end programs. Your code will run on Node throughout most of the course.

1. Just run this command in your Terminal window: `brew install node`



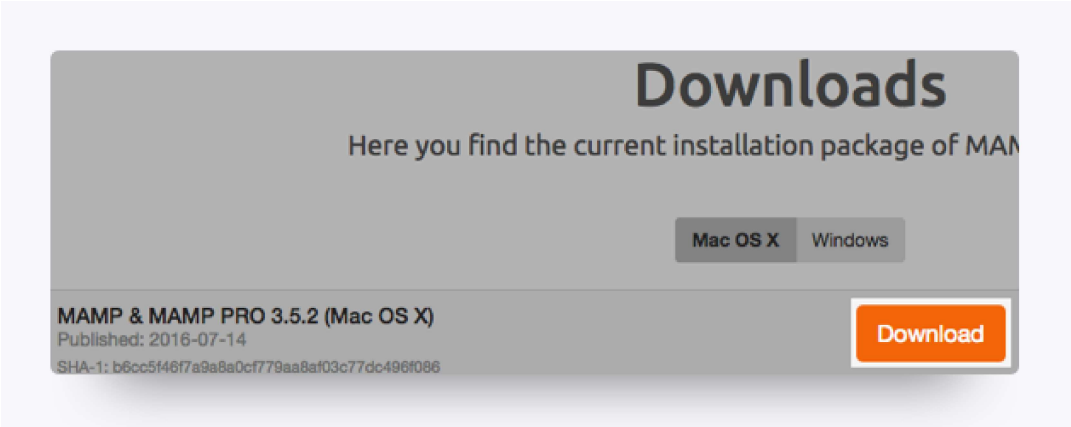
- Note: make sure you have brew installed before you run this.

2. Type `node -v` in Terminal, and hit `return`. If Terminal returns a version number, then you've successfully installed Node.

MAMP:

This tool lets developers run local servers for their back-end-reliant web apps. In other words, you can power some of your more complex sites without an internet connection.

1. Go to https://www.mamp.info/en/downloads/. Select Mac and click download.



2. Run through the installation.

SSH Key

Generating SSH keys allows developers to interface with certain remote services without having to constantly type out login information. You're going to set up an SSH key for GitHub.

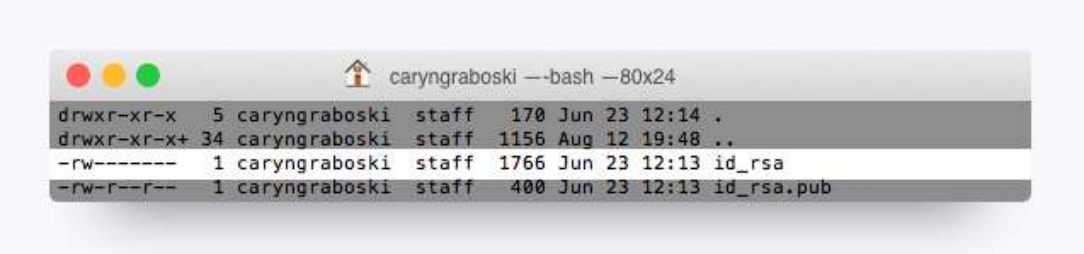
Without a key, you won't be able to push your code to GitHub without entering a password each time; trust us, that would be as irritating as needing a key to open every door in your home.

1. Sign up for an account on https://github.com.

2. Open up Terminal.

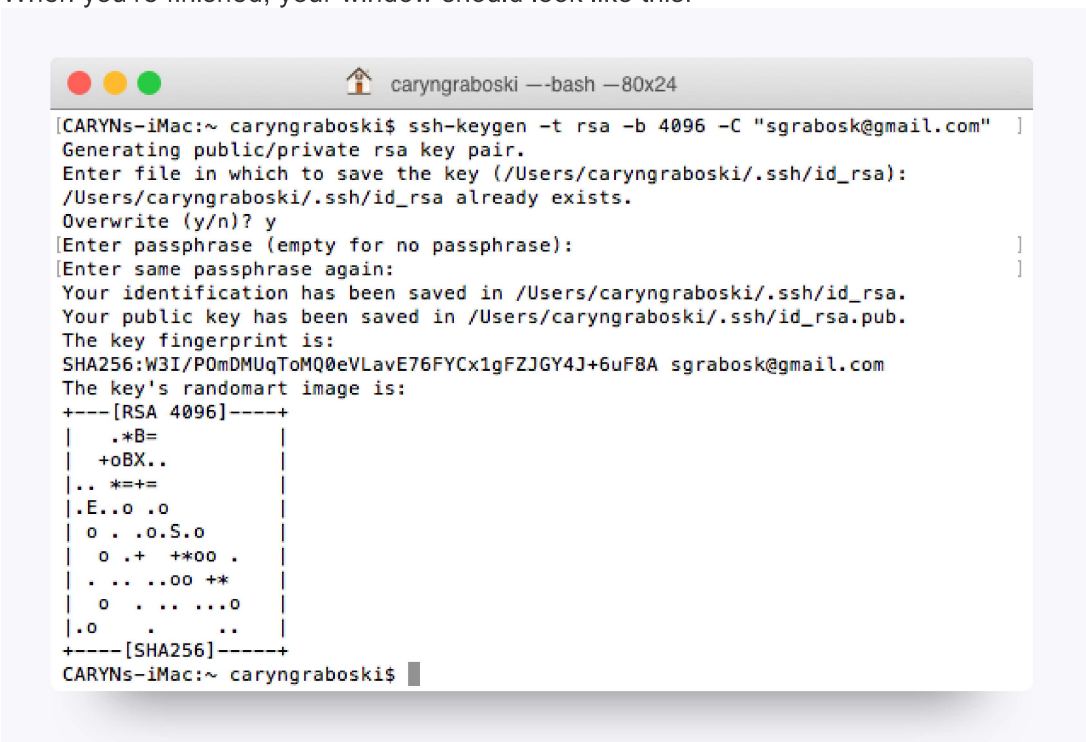
3. We need to set up SSH keys. First, let's make sure you don't already have a set of keys on your computer. Type this into your Terminal window **(copying and pasting will not work)**:

- `ls -al ~/.ssh`
- If no keys pop up, move on to step 4.
- If keys do pop up, check that none of them are listed under `id_rsa`, like in this image:

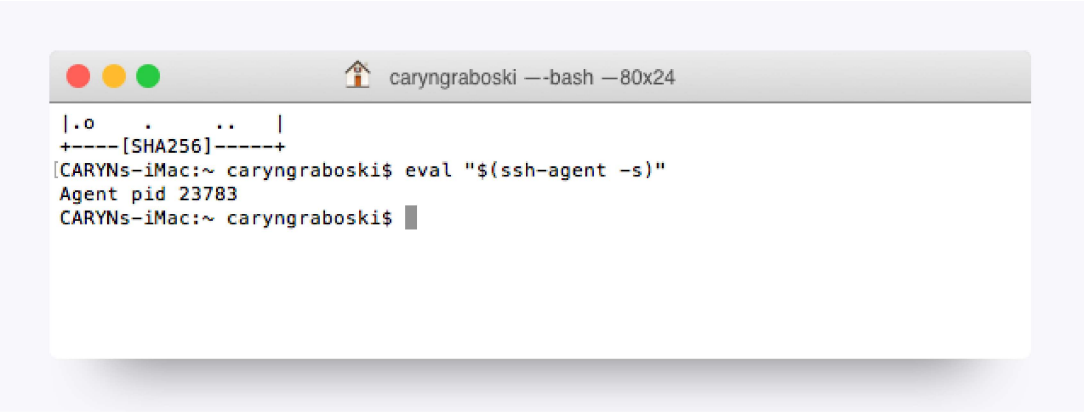


  - If you do find a key with a matching name, then you can either overwrite it by following steps 4 to 6, or you can use the same key in steps 10 and beyond. Be advised that you'll have to remember the password tied to your key if you decide not to overwrite it.

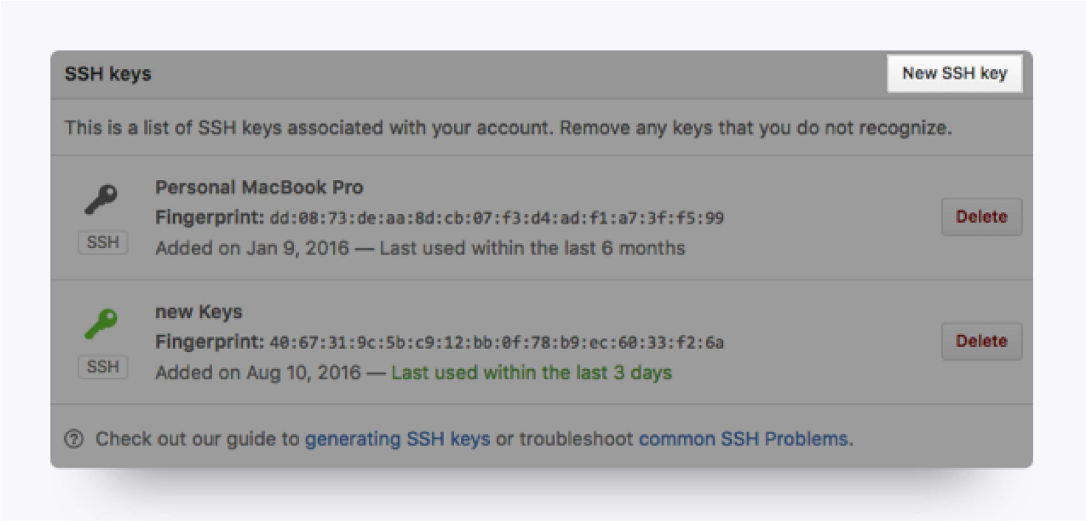
4. Type in this command along with your email to generate your keys:
   - `ssh-keygen -t rsa -b 4096 -C "YOURGITHUBEMAIL@PLACEHOLDER.NET"`
5. When asked to enter a file to save the key, just hit `return`.
   - Also enter a passphrase for your key.
   - Note: You shouldn't see any characters appear in the window while typing the password.
6. When you're finished, your window should look like this:



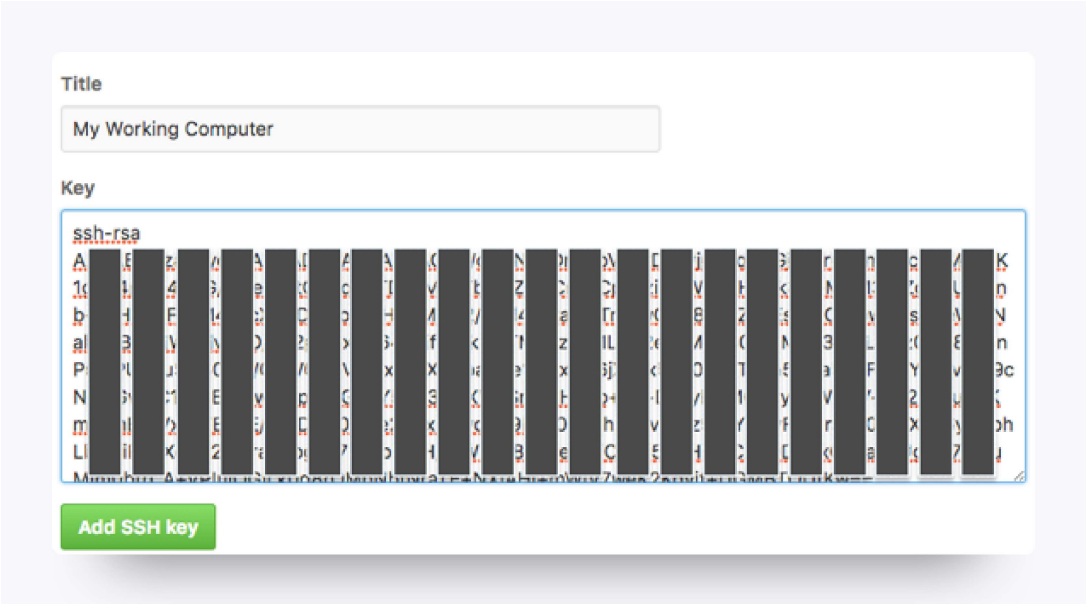
7. For the next step, we need to use a tool called an ssh agent. Let's test whether that's working on your machine. Run this command in Terminal:
   - `eval "$(ssh-agent -s)"`
   - If your Terminal window looks like the image below, move onto the next step.



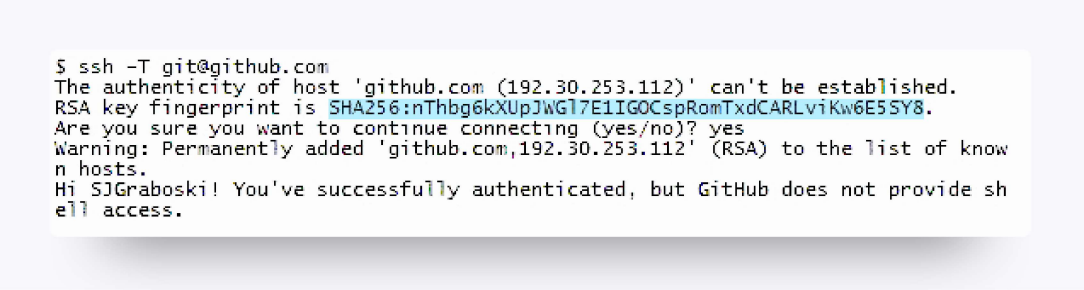
8. Now run this command:
   - `ssh-add ~/.ssh/id_rsa`
9. When prompted for a passphrase, enter the one associated with the key.
   - If you've forgotten this key, just go through step 4 to create a new one.
10. We need to add the key to GitHub. Copy the key to your clipboard by entering this command:
    - `pbcopy < ~/.ssh/id_rsa.pub`
    - You shouldn't see any kind of message when you run this command. If you do, make sure you entered it correctly.
    - Do not copy anything else until you finish the next steps. Otherwise, you'll have to enter the copy command again.
11. Go to https://GitHub.com/settings/ssh. Click the "New SSH key button."



12. When the form pops up, enter a name for your computer in the Title input. In the Key input, paste the SSH key you copied in Step 10.



13. Now we just need to add GitHub to your computer's list of acceptable SSH hosts. Go back to your Terminal window. Type in this command: `ssh -T git@github.com`
    - You should see an RSA fingerprint in your window. Only enter "yes" If it matches the one highlighted in the image below:



Setting your Git username for *every*repository on your computer

Git uses a username to associate commits with an identity. The Git username is not the same as your GitHub username.

You can change the name that is associated with your Git commits using the `git config` command. The new name you set will be visible in any future commits you push to GitHub from the command line. If you'd like to keep your real name private, you can use any text as your Git username. Changing the name associated with your Git commits using `git config` will only affect future commits and will not change the name used for past commits.

Setting your Git username for *every* repository on your computer

1. Open Terminal.
2. Set a Git username:

```
$ git config --global user.name "Mona Lisa"
```

3. Confirm that you have set the Git username correctly:

```
$ git config --global user.name
> Mona Lisa
```

Setting your email address for *every*repository on your computer

GitHub uses the email address set in your local Git configuration to associate commits pushed from the command line with your GitHub account.

You can use the `git config`command to change the email address you associate with your Git commits. The new email address you set will be visible in any future commits you push to GitHub from the command line. Any commits you made prior to changing your commit email address are still associated with your previous email address.

For more information on commit email addresses, including your GitHub-provided `noreply` email address, see "About commit email addresses."

Setting your email address for *every* repository on your computer

1. Open Terminal.
2. Set an email address in Git. You can use your GitHub-provided `no-reply` email address or any email address.

```
$ git config --global user.email "email@example.com"
```

3. Confirm that you have set the email address correctly in Git:

```
$ git config --global user.email
email@example.com
```

4. Add the email address to your GitHub account by setting your commit email address on GitHub, so that your commits are attributed to you and appear in your contributions graph.

Amaze-Balls!

If you got through all the installations, give yourself a pat on the back! Installations are never fun, but just like taxes, ya gotta do them.

Be sure to take a break before continuing with the rest of the pre-work.

Copyright

Coding Boot Camp © 2018. All Rights Reserved.