

---

# Table of Contents

Introduction	1.1
Module #1 Prepare for Pre-Work!	1.2
Module #2 - Machine Ready	1.3
Assignment #1 - Get Yo' Tools Installed on Mac	1.3.1
Assignment #1 - Get Yo' Tools Installed on Windows	1.3.2
Assignment #1 - Curling Up With Python	1.3.3
Module #3 - Time to get Employable	1.4
Assignment #2 - Connect Professional Dots!	1.4.1
Module #4 - Get Your Mind Right	1.5
Module #5- Let's Get To Work	1.6
Want More? - Supplemental Resources for the Curious of Mind	1.7

# Data Boot Camp! - Pre-Work Curriculum

*"Give me six hours to chop down a tree, and I will spend the first four sharpening the axe." - Abraham Lincoln*

---

## Welcome!



Congratulations on your acceptance to the Data Program!

You are about to embark on an intense and transformational journey that will dramatically improve your skill set. At times, it will feel challenging, frustrating, and utterly bewildering, but trust us; it *will* be worth it in the end.

Be warned! This program's curriculum is not easy. You will be exposed to new ideas at breakneck speeds and will be challenged to complete difficult exercises. Be prepared to push yourself and to be pushed.

That said, remember every one of us, behind the scenes, is rooting for you. You were accepted into this program because we believe in your potential to succeed. You will have excellent instructors, dedicated TAs, an extremely rich curriculum, and supportive peers to nudge you forward. If you put in the long hours and hard effort, we're confident you will find the success you're looking for.

## The Importance of Pre-work

Because of the challenging nature of this program, we want you to come ready to **sprint** on Day 1. To help prepare you, we've created a module to be completed before your first day.

Good luck! Have fun! Get pumped!

You are in for an amazing ride.



---

## Pre-Work Module

### Module #1 - Prepare for Pre-Work

Let's get started!

---

### Module #2 - Machine Ready (Required)

Let's get all those pesky installs out of the way.

---

### Module #3 - Time to get Employable

---

### Module #4 - Get Your Mind Right

This bootcamp won't be easy. Set the right attitude and expectations through the help of this module.

---

### Module #5 -Let's Get To Work

Submit an assignment

---

## Want More? - Supplemental Resources for the Curious of Mind (Optional)

---

### Copyright

Data Boot Camp © 2017. All Rights Reserved.

# Module #1 - Prepare for Pre-Work!

## Use The Right Tools for the Job

To ensure you can successfully use all of the tools and technologies used in this course, you need to have the right equipment to work on. Below are the equipment requirements for this course **-if you don't meet these requirements, please contact your Student Success Manager.**

Taking an in-person class?

- Mac or Windows with 8GB ram and dual processor (64 bit, no 32 bit)
- No Linux allowed

Taking an online class?

- Mac or Windows with 8GB ram and dual processor (64 bit, no 32 bit)
- No Linux allowed
- Second monitor compatible with laptop
- Webcam, microphone, headphones
- High speed internet connection in a quiet space (Minimum: Download 25 Mbps, Upload 5 Mbps)

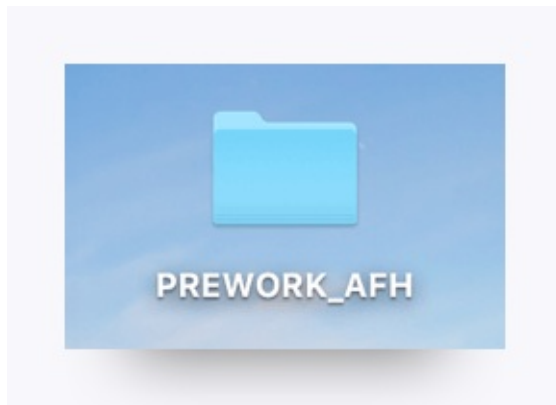
---

## How to Submit Pre-Work

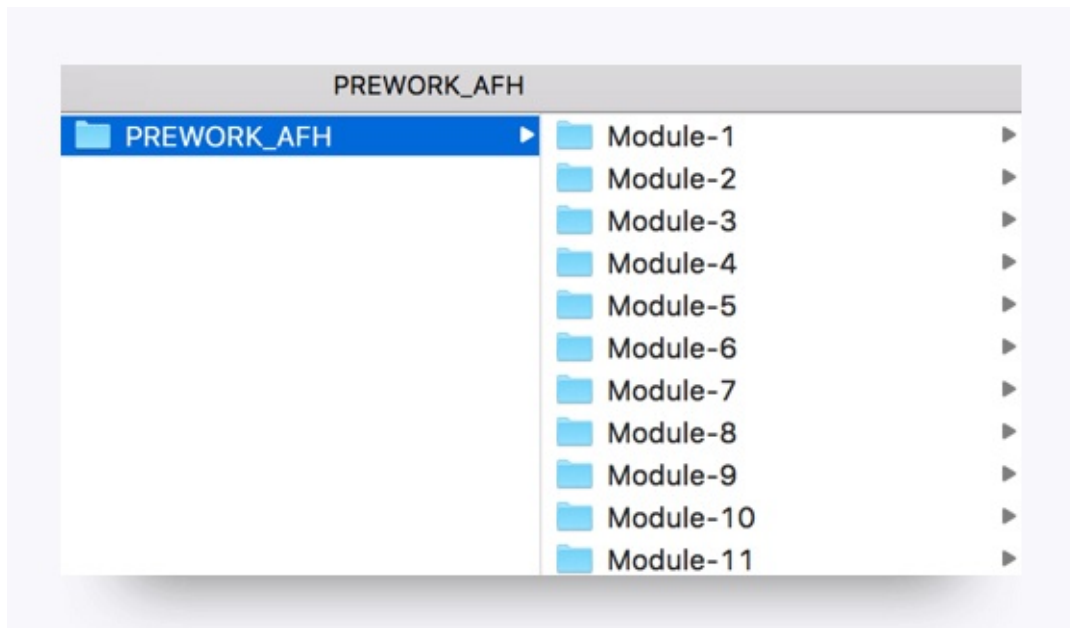
Ultimately, you will be submitting each of the completed assignments in a single folder onto [Bootcampspot-v2.com](https://bootcampspot-v2.com). You will be given access to the site upon enrollment.

Once you gain access, you can use the below instructions as a final submission guide.

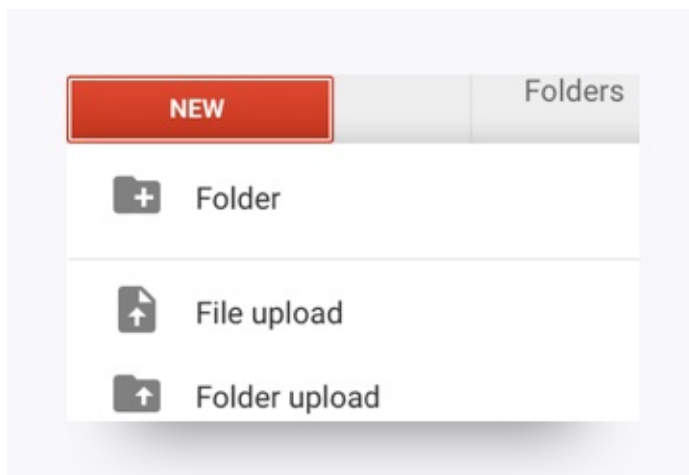
1. Create a local folder on your computer titled: PREWORK\_{INITIALS} (with your initials replacing the stuff in brackets, e.g. "PREWORK\_AFH").



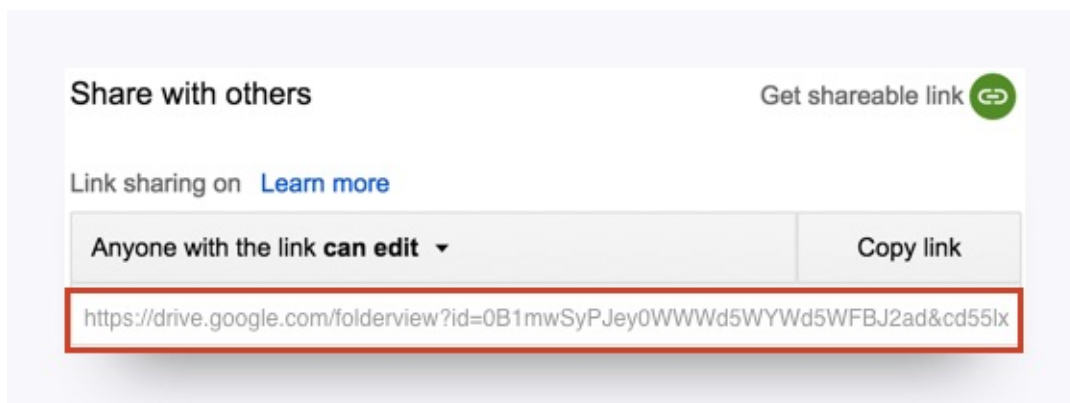
2. Inside of the folder create sub-folders for each of the assignments. Your folder should look something like the below.



3. Upon completing each assignment, save your solution into the associated sub-folder of your pre-work folder. In some cases, this will mean copying multiple files into the sub-folder.
4. Then create an account on [Google Drive](#) if you do not already have one.
5. Once you are done with all of the pre-work, utilize the "Folder Upload" button on Google Drive to upload the folder onto Google Drive.



6. Then right click the folder on Google Drive and change the sharing settings such that anyone with the link can edit.



7. Copy the link associated with your folder in Google Drive.
8. Then log in to Bootcampspot and find the Pre-Work Assignment associated with your class.
9. Click on the assignment and paste the Google Drive link so your Instructor and TA can review your assignment.

And that's it. Now get cracking on those modules!

---

## Assignment:

- None!

## Supplemental Resources:

- [Google Drive](#)
-

## Copyright

Coding Boot Camp © 2018. All Rights Reserved



## Module #2 - Get Yo'self Machine Ready

---

### More than Data...

In this class, you can expect to be exposed to a wide range of tools with names you may have never heard—tools like Excel, Git and VS Code. At first, the sheer number of tools you'll be expected to use may seem overwhelming, but trust us! With a little time and with a little effort, they will be as familiar to you as a scalpel is to a surgeon or a sewing needle is to a seamstress.

They're all just tricks of the trade.



### Ready for Action!

Coming into your first day of class, you will be expected to have several tools already installed. This will ensure you're ready to start learning immediately. The purpose of this module is to walk you through the process of installing all your tools and give you a brief primer on the roles they play.

After completing this module, you will have each of the following installed:

- Google Chrome
- Slack
- Microsoft Excel
- Git / GitHub
- Git Bash (Windows Only)
- Terminal (Mac Only, Pre-Installed)
- Heroku Toolbelt
- Homebrew (Mac Only)
- Python & Anaconda
- Mongo DB

- VS Code

In addition, you will also have accounts on each of the following websites:

- LinkedIn
- GitHub
- Stack Overflow
- Slack

**Having trouble with set up? Not to worry. Your instructional staff will help you troubleshoot any errors and answer any questions on the first day of class. Just sit tight until then!**

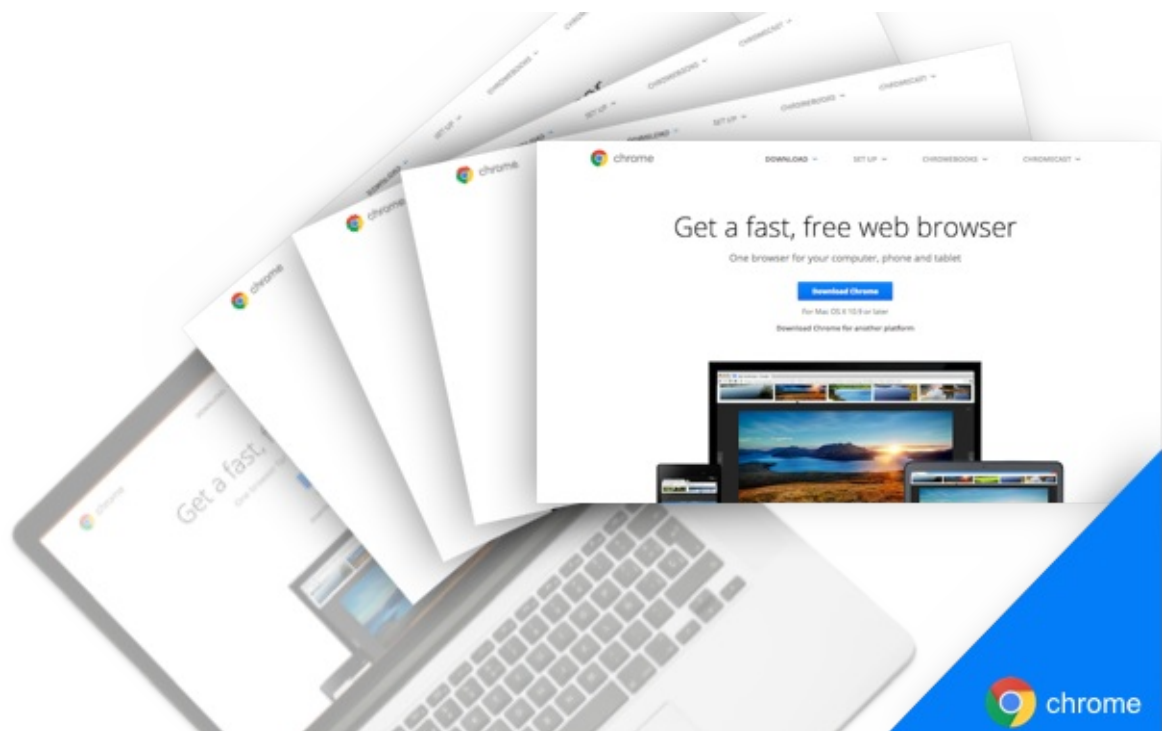
---

## Tools for Fools

Before we start installing everything willy-nilly, let's take a moment to examine each of these tools to better understand the role they play.

### Google Chrome

This one is straightforward. It's a web browser. In our case, we'll be using it to quickly see if our code is working. While in truth, you can use any web browser, Google Chrome has a number of tools that make it an ideal platform for developing so we strongly encourage you to make the switch.



## Slack

This is one you will be using literally every single day for the next six months. Slack is an online communication tool that is a mix of forum, of instant messenger and of email - all rolled in one. It's a tool that is used by countless organizations worldwide.

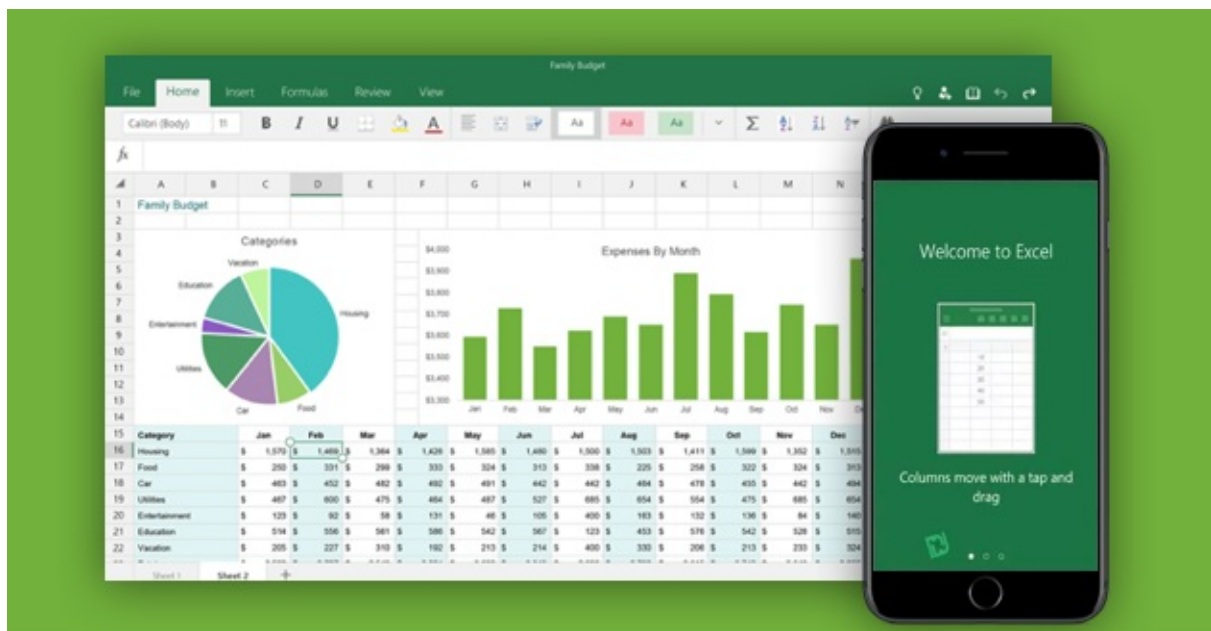
In our Bootcamp, we'll be using Slack extensively to send code snippets during class, to relay important announcements and to facilitate group exercises. You will receive the link to your class-specific channel during orientation. You will definitely want to have this installed on Day 1. (Note how we said installed and not simply logged into. While the web client is good, for our class, you will want to install the actual program on your machine.)



## Microsoft Excel

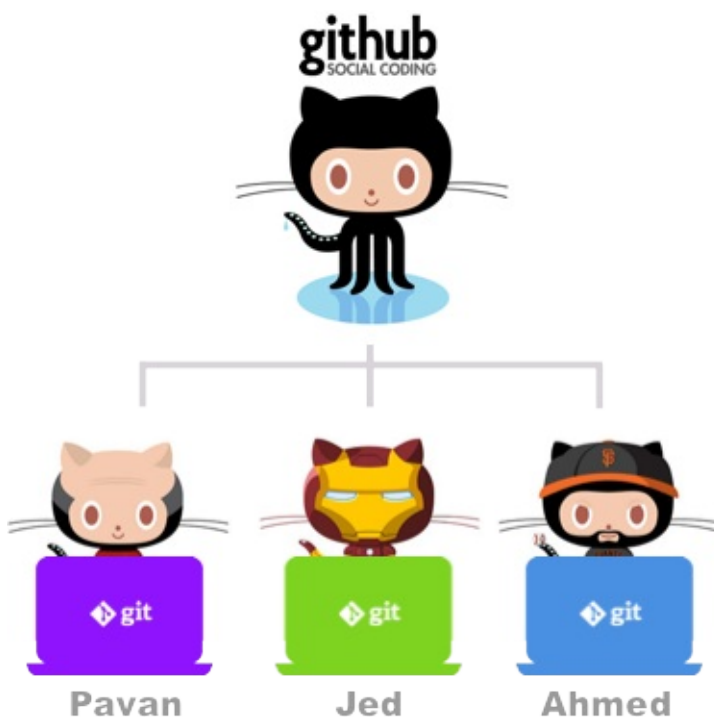
You'll be learning the ins and outs of Microsoft Excel in the first few weeks of the course. You can get Excel on a subscription basis via Office 365 or as a one-time purchase. Details on both options are available at the Microsoft Store. Be sure to select the correct operating system and that it is Excel 2016 or later before purchase.

Once you've paid for and downloaded the software, you'll be prompted with the latest installation instructions for your platform. Simply follow them and confirm your installation after by opening Excel!



## Git / Github

Because code files are ones in which multiple developers need to carefully build upon each other's work, Git offers a specialized set of strategies for orchestrating the collaboration. GitHub then takes all these collaborative actions and stores them online. In a way, GitHub serves as a sort of Dropbox for coders. It offers a central place for individuals or teams to upload their code, to view revision history and to make changes to a master set of files. You'll come to learn a lot about Git and GitHub in your first week of class. You will receive the link to your class-specific repository during orientation.

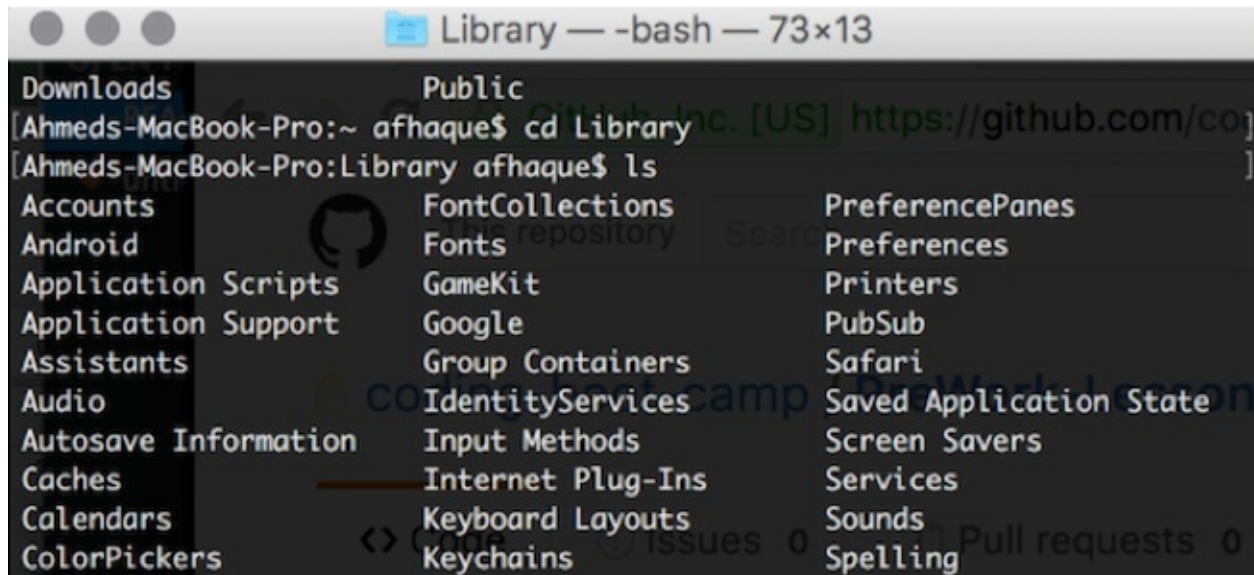


## Git Bash (Windows Only) / Terminal (Mac, Pre-installed)

Git Bash (Windows) and Terminal (Mac) offer a command line interface for working with the files and folders on your computer.

*So is it like Finder or Windows Explorer?*

Kind of... except there are no pictures or visuals. It's just a box with text

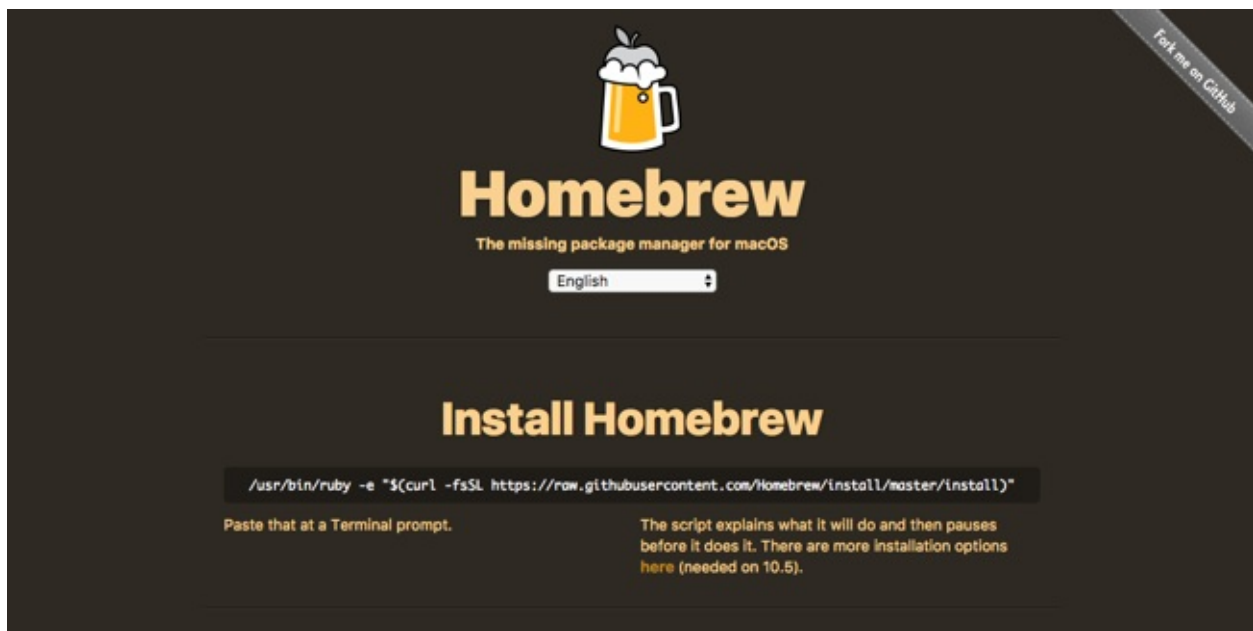


*Uh, why would I want that?*

You'll come to understand over time, but in many situations, utilizing a command line interface can be faster and can be more powerful than relying on the operating system's GUI. You'll get plenty of exposure to the command line on day one.

## Homebrew (Mac Only)

Homebrew is a Mac-specific toolkit that makes it easy to install, using the command line, a variety of applications. It can greatly simplify the installation process for various tools you'll be using in your development career.

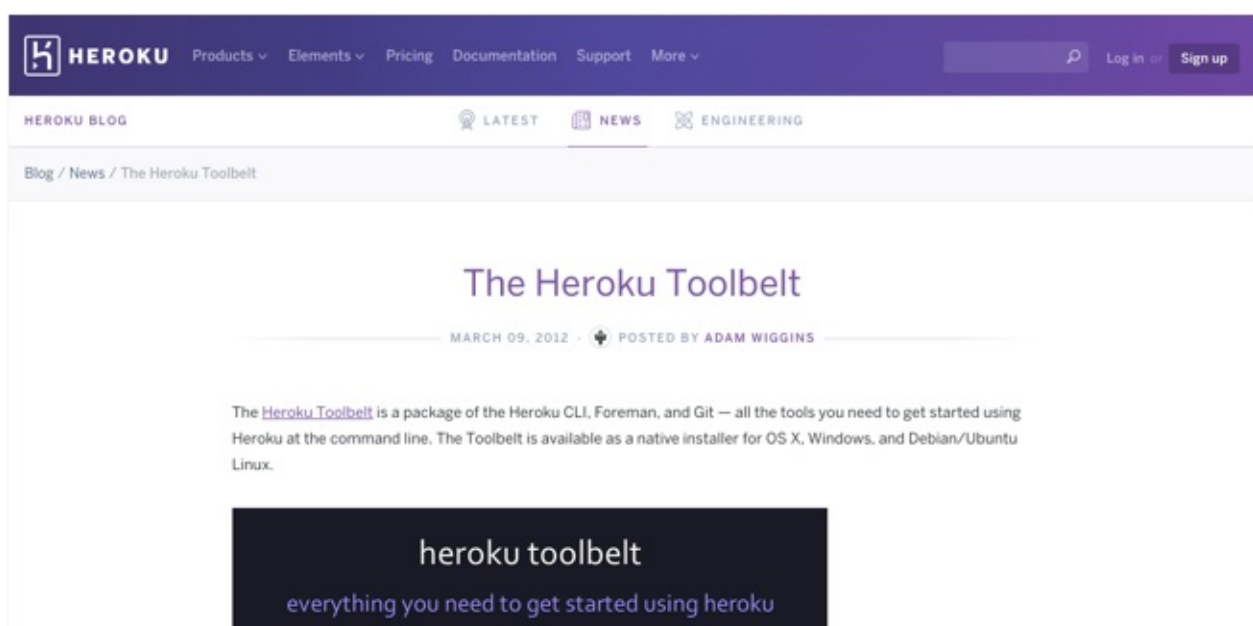


## Heroku Toolbelt

While you may not feel ready yet, very quickly, you'll be creating complete websites on your computers. But once you have these sites made, how do you get them online for the world to see?

Hosting platforms like Heroku effectively serve as a dumping ground for web applications. These platforms are set up to take your web applications' code, to activate them and to then assign them to a URL for other visitors to see. In a sense, they host your applications so that every internet user has access to them. You'll learn a lot about how this works towards the tail end of the course.

As it relates to the pre-work, you'll be installing the Heroku Toolbelt which offers a set of easy-to-use tools for interacting with the Heroku platform online.





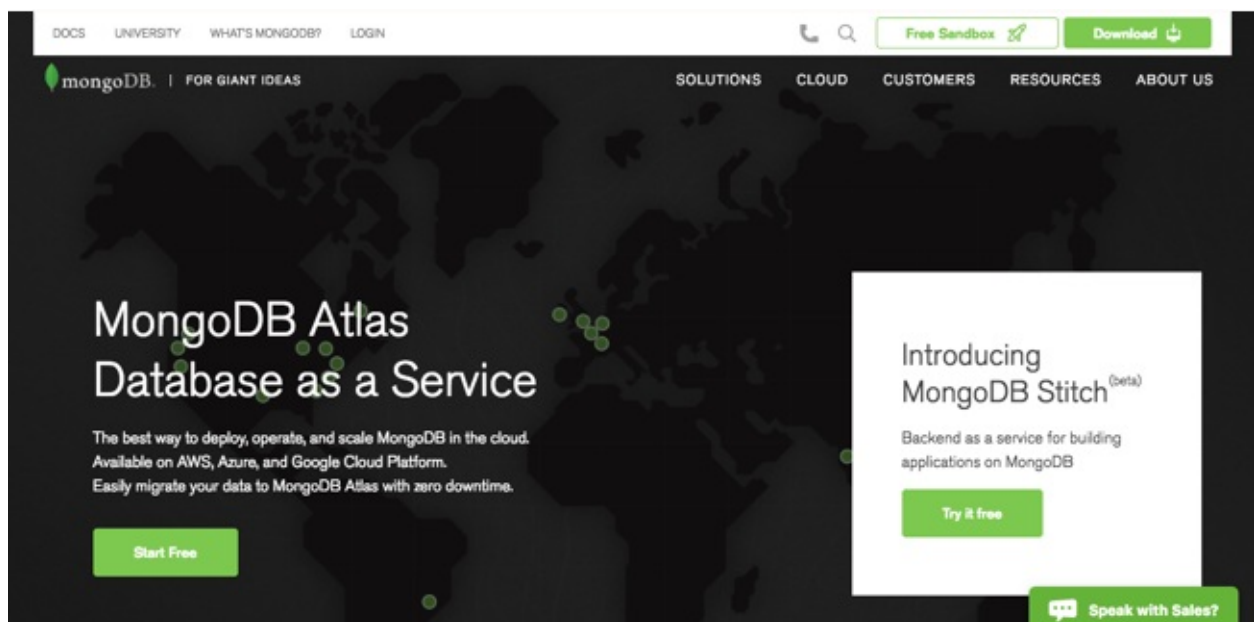
## Python & Anaconda

Python and its ecosystem are extremely popular tools in the world of data analysis. This course will use a popular all-in-one solution called Anaconda which contains everything we'll need for the Python units of the course.



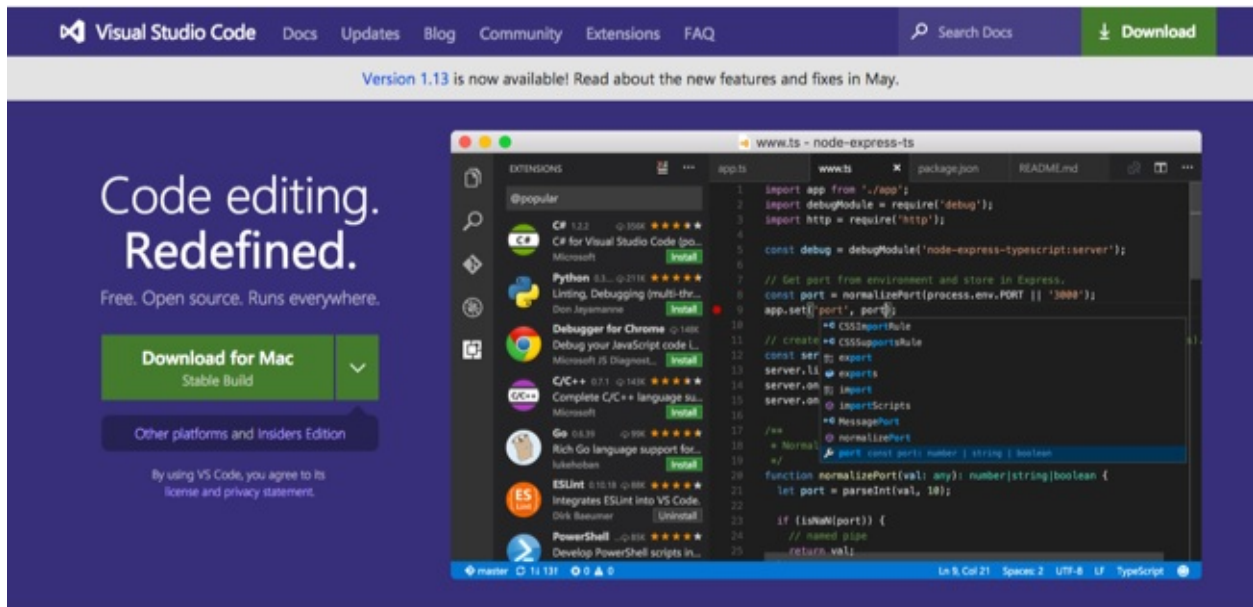
## Mongo DB

Like MySQL, MongoDB is a database but it works in a fundamentally different way and has certain advantages and disadvantages as compared to MySQL. We'll use it during the web portion of the course.



## VS Code

Oh, the power of VS Code! A little program that does so much!



VS Code (Visual Studio Code) is a free text editor that runs on the Mac, Linux and Windows operating systems. Now, for the uninitiated, the first thought that comes to mind when we say text editor may be something basic like Notepad or TextEdit. But for developers, text editors are like the cozy pillow on which they rest their heads. This is because fundamentally, programming is all about creating text in files with various extensions. When we create a block of HTML like the one below...

```

1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="utf-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1">
7      <title>Data Viz</title>
8    </head>
9    <body>
10
11      <h1>Yo Yo Yo</h1>
12      <h2>The Data Viz peeps are in the building!</h2>
13
14    </body>
15  </html>

```

...what we've really done is just created a block of text. There are funny symbols in there like those angle brackets and what not but at it's most basic level, it's just text.

Now to a simple text editor, this is where the comprehension stops. Our block of HTML remains a block of text. But to more powerful text editors like VS Code on the other hand, these blocks of text are immediately recognized as code (so long as you include the right file extension). This means that VS Code can give us a more visually intuitive understanding of



the code through indicative coloring, smart tabs and through its autocomplete functionality. The result is that creating the above block of HTML becomes a more natural process and also one that can be debugged more quickly.

## Collect Your Tools

And that's it!

It's time to collect your tools and begin. As you'll see in the links below, we have guides for both Mac and Windows users on the process for getting setup. Follow the instructions closely and do your best with the information you have. (Yes, we know there is a lot to install.)

Just one bit of advice! Throughout the course, you will frequently be looking at documentation to install and to utilize unfamiliar tools. Resist the urge to stop and ask, "Am I doing this right?" Instead, just take your best stab at it. This is an important attitude to start developing *now*. A lot of what you'll be exposed to over the next six months will be unfamiliar. Don't be phased by it and don't get bogged down by it. Trust your instincts! We'll make sure that anyone who is lost gets the help they need on day one.

Good luck! Make sure your tools are extra pointy.

---

## Assignments (Required)

- [Get Yo' Tools Installed on Mac](#)
  - [Get Yo' Tools Installed on Windows](#)
  - [Curling Up With Python \(Mac & Windows\)](#)
- 

## Copyright

Data Boot Camp © 2017. All Rights Reserved.

# Get Yo' Tools Installed on Mac

---

In this class, you can expect to be exposed to a wide range of tools with names you may have never heard - tools like Excel, Git and VS Code. At first, the sheer number of tools you'll be expected to use may seem overwhelming but trust us! With a little time and with a little effort, they will be as familiar to you as a scalpel is to a surgeon or a sewing needle is to a seamstress.

## Overview

In this assignment you will be installing all 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>
- Stack Overflow: <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.

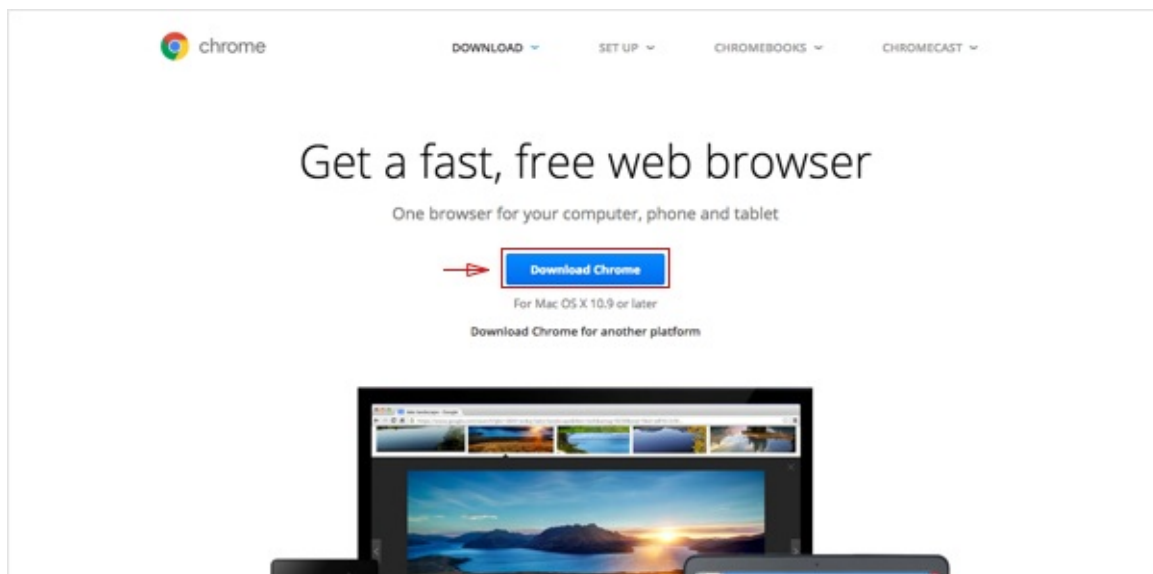
## Da Big Installation Enchilada

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 the download page [here](#).

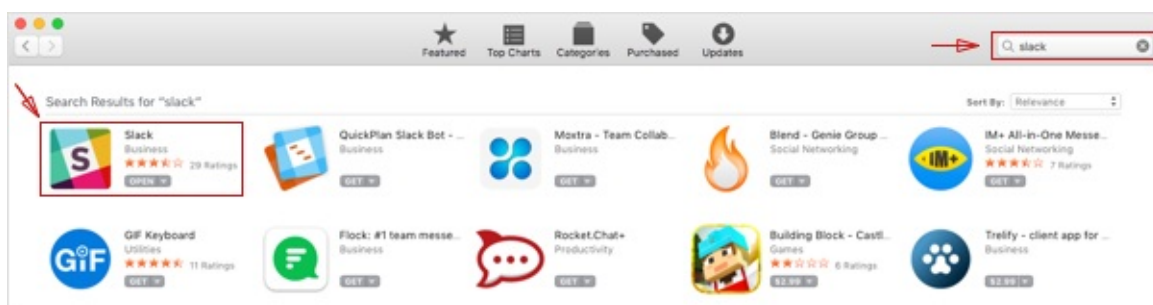


2. Download, open and run through Chrome's installation file.

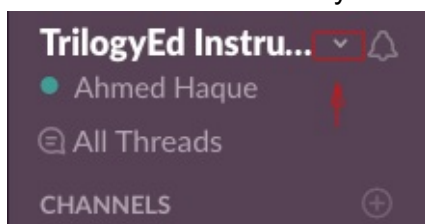
## Slack

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.

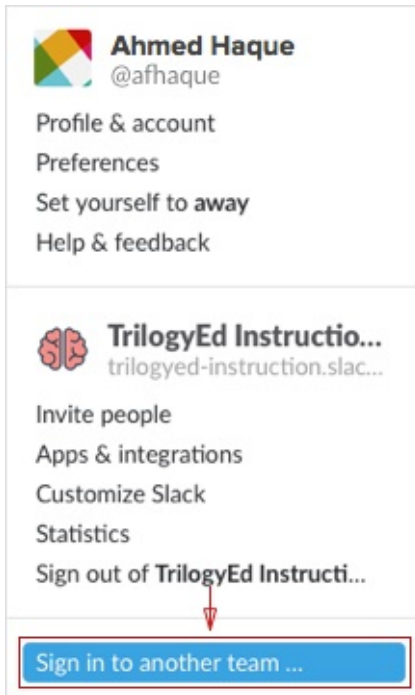
1. If you don't have the Slack app yet, search for Slack in your Mac's App store and then click the Get button under the app's listing. Click the button again when it displays "install."



- 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.

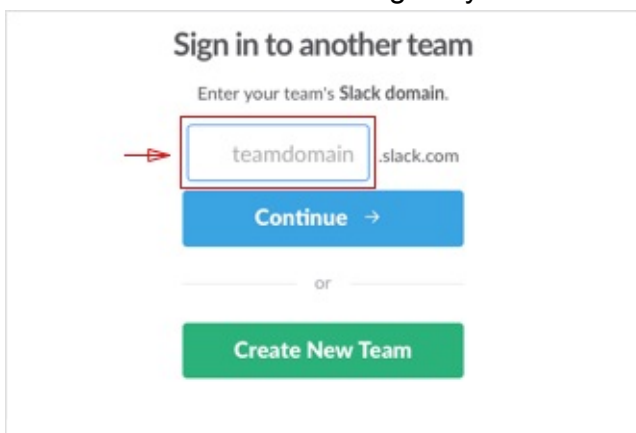


- Then select "Sign in to another team ..."



3. As you run through the guide, make sure you do the following:

- Enter in the domain we gave you for Slack.

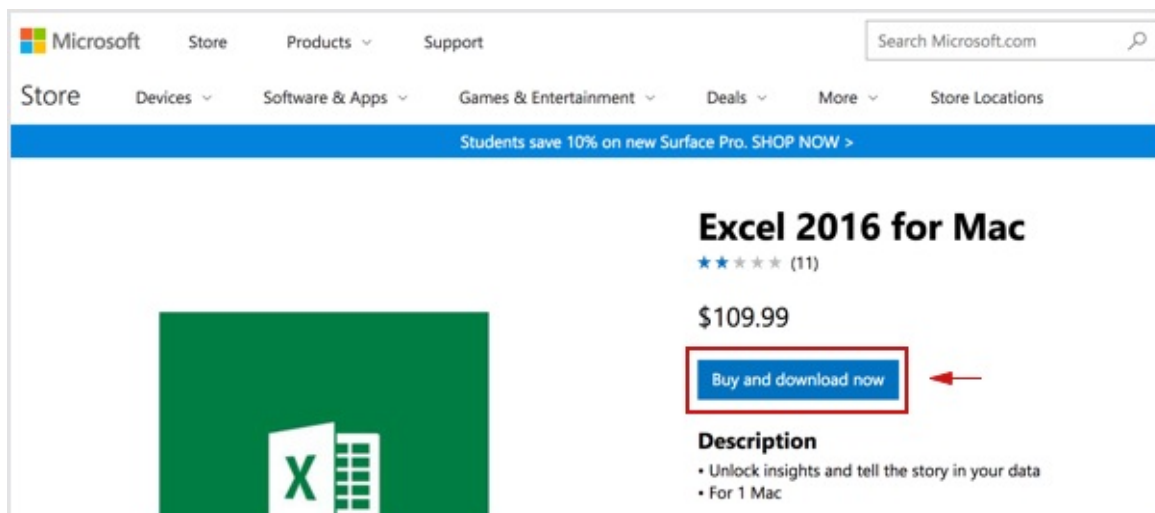


- Enter in the email with which we invited you, as well as your password, when prompted.



## Microsoft Excel

1. Go to the [Microsoft Store](#) and click the "Buy and Download" button. Make sure it's Excel 2016 or later that you are downloading.

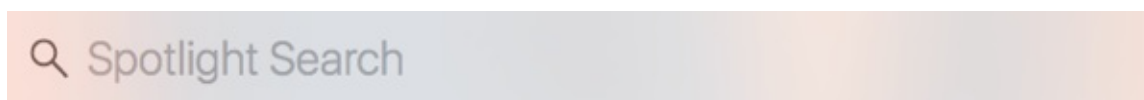


2. Follow the set up instructions after the download is completed.

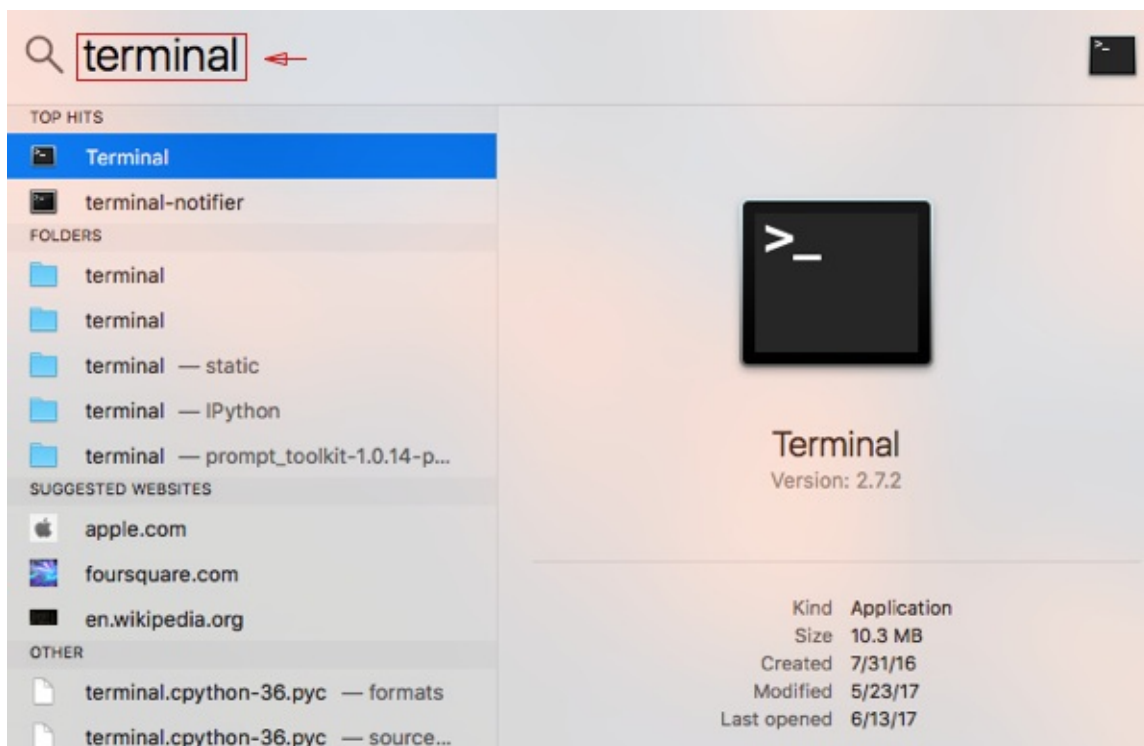
## Terminal

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+space bar to open Spotlight Search.



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



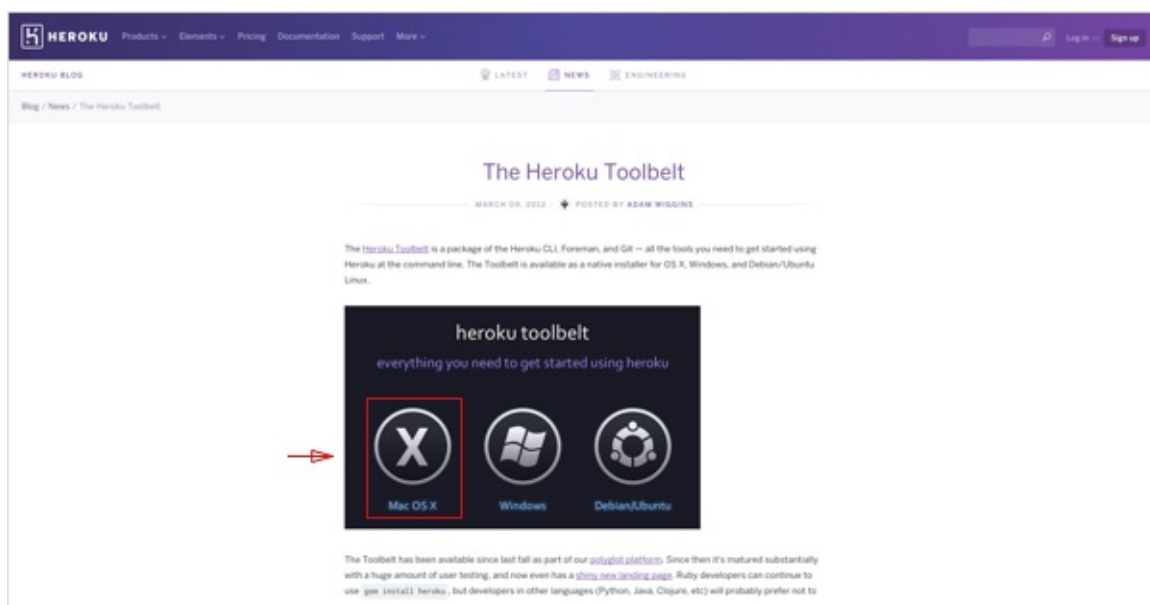
3. Keep this window open; you'll need it for the next steps.

```
Nicky — -bash — 85x26
Last login: Mon Jun 19 23:04:21 on ttys001
Nicholas-MBP:~ Nicky$
```

## 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 login` into the command line and then press return.

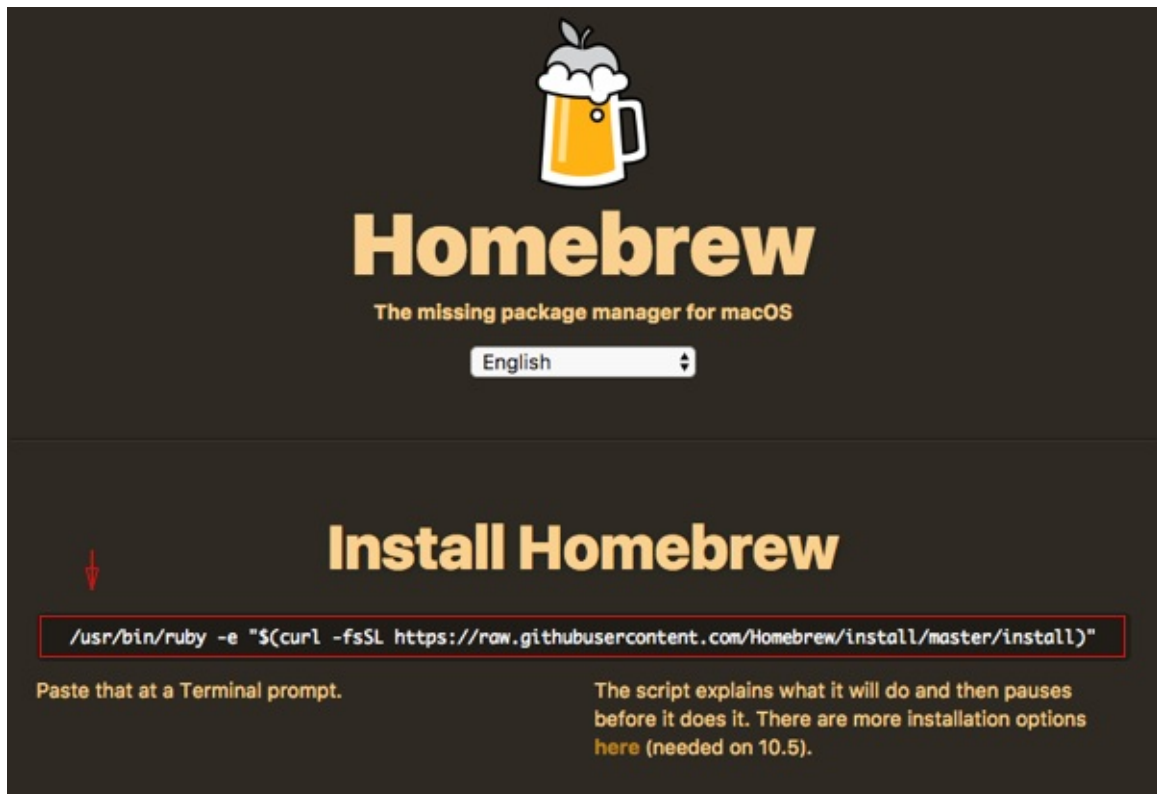
```
Nicky — -bash — 85x26
Last login: Mon Jun 19 23:04:21 on ttys001
Nicholas-MBP:~ Nicky$ heroku login
heroku-cli: Installing CLI... 21.65MB/21.65MB
Enter your Heroku credentials.
Email: 
```

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.

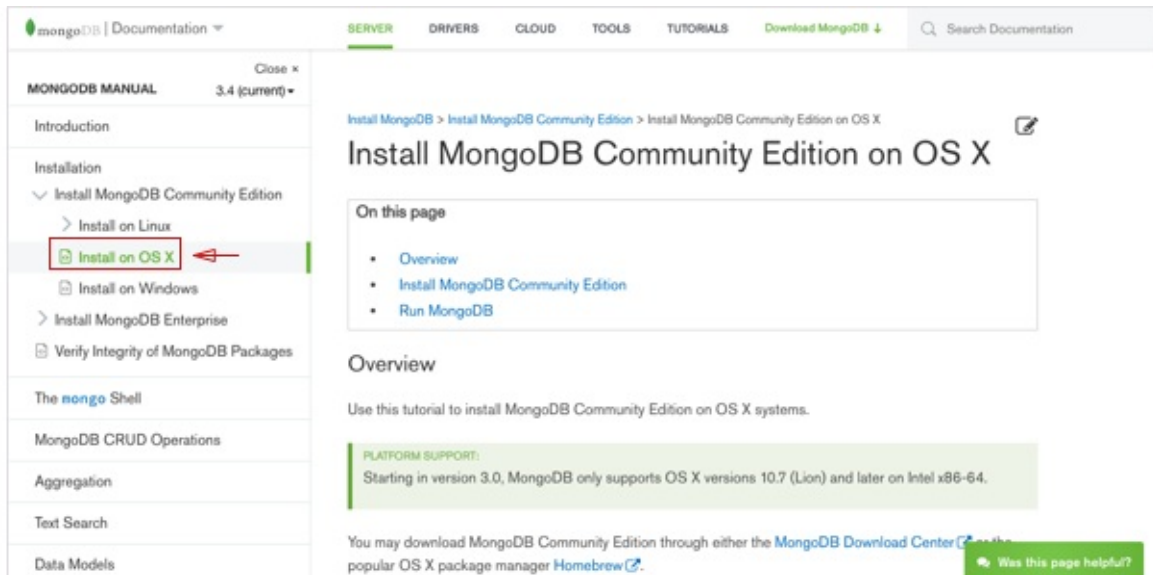
```
caryngraboski ~ ruby -e #!/System/Library/Frameworks/Ruby.framework/Versi...
[CARYNs-iMac:~ caryngraboski$ /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
==> This script will install:
/usr/local/bin/brew
/usr/local/Library/...
/usr/local/share/doc/homebrew
/usr/local/share/man/man1/brew.1
/usr/local/share/zsh/site-functions/_brew
/usr/local/etc/bash_completion.d/brew
==> The following directories will be made group writable:
/usr/local/.
==> The following directories will have their owner set to caryngraboski:
/usr/local/.
==> The following directories will have their group set to admin:
/usr/local/.

Press RETURN to continue or any other key to abort
█
```

## Mongo DB

Since installing MongoDB is somewhat involved and we won't be using MongoDB until rather late in the course, it's not necessary to have it installed on Day 1 - there will be time later for us to set it up together.





However, if you'd like to get a head start, [installation instructions for Windows user are here](#) and [installation instructions for OS X are here](#).

## 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.

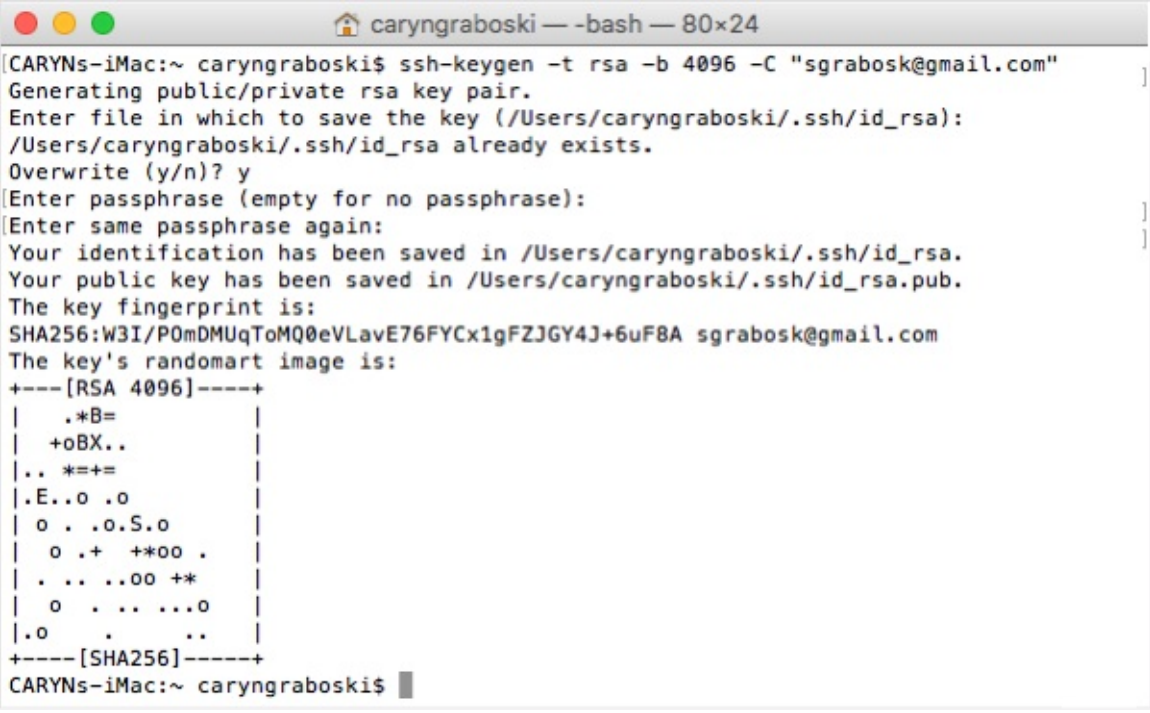
1. Sign up for an account on <https://github.com>.
2. Open up Terminal.
3. We need to set up SSH keys. First, let's make sure you don't already have a set of keys on your computer. Type this into your Terminal window (copying and pasting will not work):

- `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:

```
drwxr-xr-x  5 caryngraboski  staff   170 Jun 23 12:14 .
drwxr-xr-x+ 34 caryngraboski  staff  1156 Aug 12 19:48 ..
-rw-----  1 caryngraboski  staff   1766 Jun 23 12:13 id_rsa
-rw-r--r--  1 caryngraboski  staff    400 Jun 23 12:13 id_rsa.pub
```

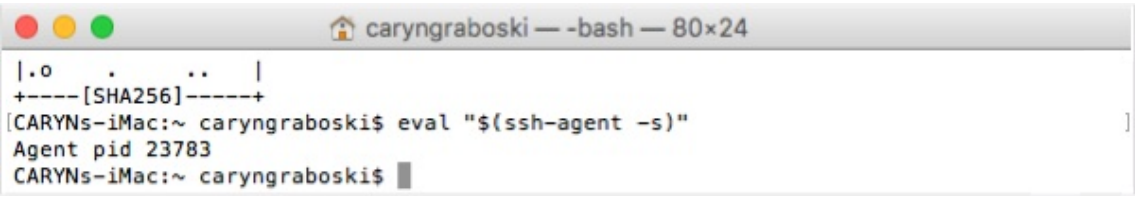


- If you do find a key with a matching name, then you can either overwrite it by following steps 4 to 6 or you can use the same key in steps 10 and beyond. Be advised that you'll have to remember the password tied to your key if you decide not to overwrite it.
4. Type in this command along with your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



```
caryngraboski — -bash — 80x24
[CARYNs-iMac:~ caryngraboski$ ssh-keygen -t rsa -b 4096 -C "sgrabosk@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/caryngraboski/.ssh/id_rsa):
/Users/caryngraboski/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /Users/caryngraboski/.ssh/id_rsa.
Your public key has been saved in /Users/caryngraboski/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:W3I/P0mDMUqToMQ0eVLavE76FYCx1gFZJGY4J+6uF8A sgrabosk@gmail.com
The key's randomart image is:
+----[RSA 4096]-----+
|  . * B =          |
| + o B X . .       |
| .. * = + =        |
| . E . . o . o     |
| o . . o . S . o   |
| o . +   + * o o .  |
| . . . . o o + *    |
| o . . . . . o     |
| . o . . . .       |
+----[SHA256]-----+
CARYNs-iMac:~ caryngraboski$
```

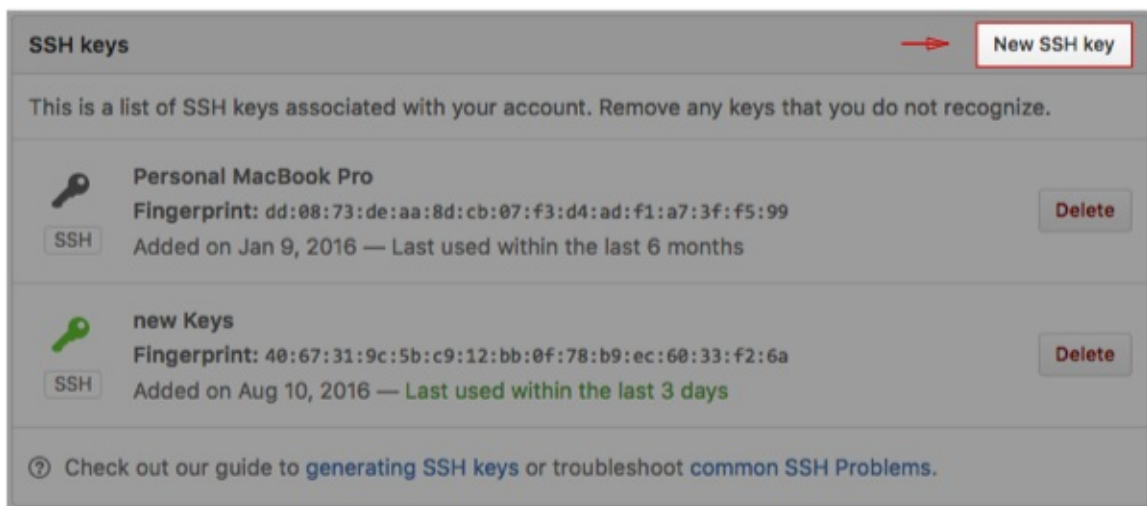
7. For the next step, we need to use a tool called an SSH Agent. Let's test whether that's working on your machine. Run this command in Terminal:
  - `eval "$(ssh-agent -s)"`
  - If your Terminal window looks like the image below, move onto the next step.



```
caryngraboski — -bash — 80x24
| . o . . . . |
+----[SHA256]-----+
[CARYNs-iMac:~ caryngraboski$ eval "$(ssh-agent -s)"
Agent pid 23783
CARYNs-iMac:~ caryngraboski$
```

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:

```
$ ssh -T git@github.com
The authenticity of host 'github.com (192.30.253.112)' can't be established.
RSA key fingerprint is SHA256:nThbg6kXUpJWGl7E1IGOCspRomTxdCARLviKw6E5SY8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'github.com,192.30.253.112' (RSA) to the list of known hosts.
Hi SJGraboski! You've successfully authenticated, but GitHub does not provide shell access.
```

## 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

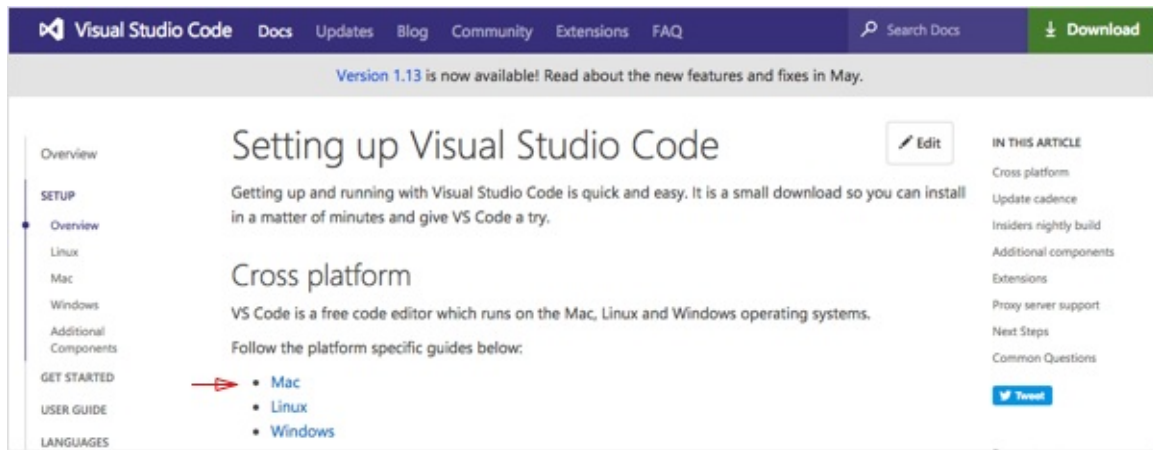
- 1 Open Terminal.
- 2 Set an email address in Git. You can use your GitHub-provided `no-reply` email address or any email address.  

```
$ 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.

## VS Code

1. Head to the [Visual Studio Code](#) docs and select the appropriate installation for your machine.
2. Click on Download Visual Studio Code for Mac.



3. Double-click the downloaded archive to expand the contents.
4. Drag the Visual Studio Code app to the Applications folder, making it available in the Launchpad.
5. Add VS Code to your Dock by right-clicking on the icon and choosing Options, Keep in Dock.

## 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

Data Boot Camp © 2017. All Rights Reserved.

# Get Yo' Tools Installed on Windows

---

In this class, you can expect to be exposed to a wide range of tools with names you may have never heard - tools like Excel, Git and VS Code. At first, the sheer number of tools you'll be expected to use may seem overwhelming, but trust us! With a little time and with a little effort, they will be as familiar to you as a scalpel is to a surgeon or a sewing needle is to a seamstress.

## Overview

In this assignment, you will be installing all 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>
- Stack Overflow: <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

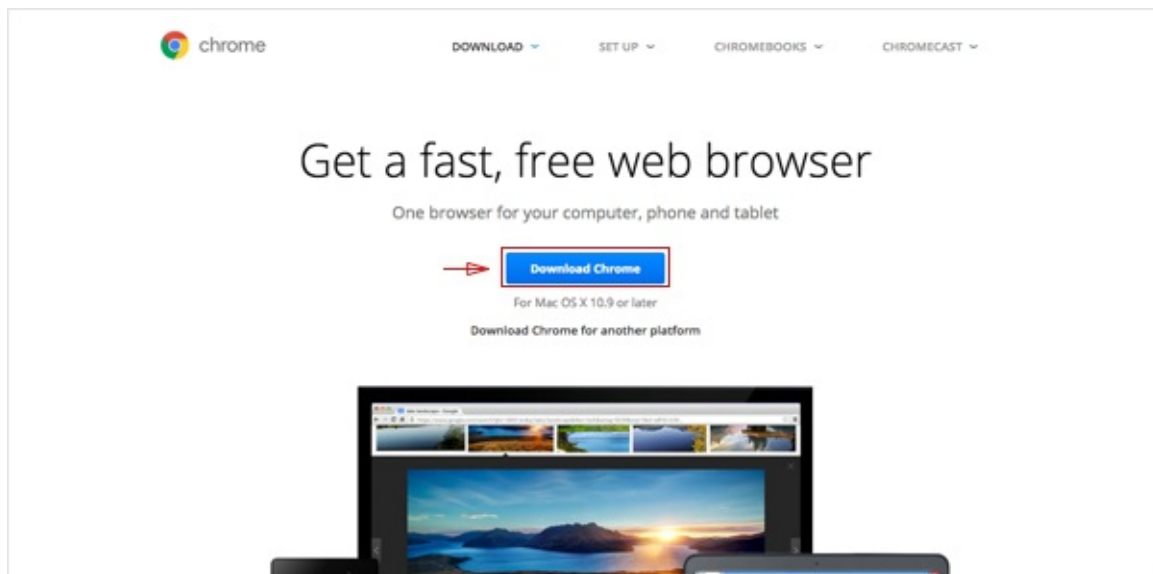
The rest of this assignment will walk you through the specific steps associated with installing each of the tools you'll need. Follow the instructions closely!

## 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 the download page [here](#).

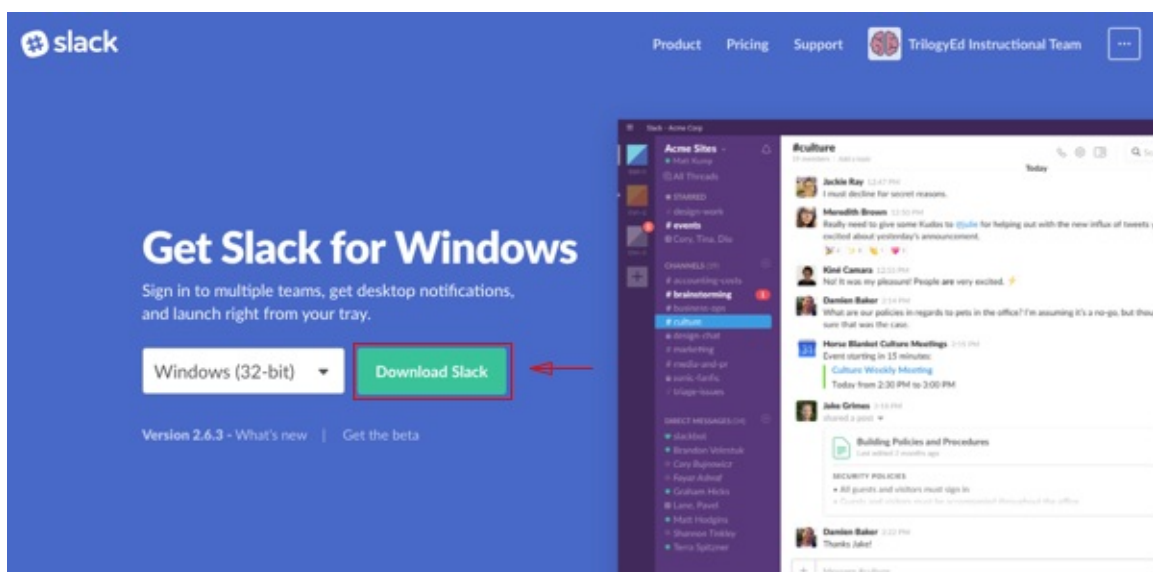


2. Download, open and run through Chrome's installation file.

## Slack

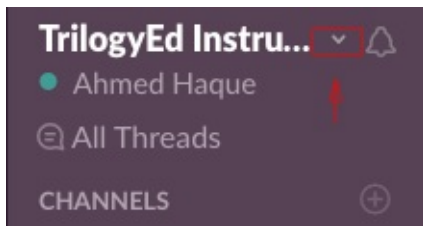
You'll be messaging your instructor, your TAs and your fellow classmates with this business-centric chatting app. The teaching and career staff will post some of their most important announcements here so set this program up as soon as you can. You will receive the link to your class-specific channel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

1. If you don't have the Slack app yet, go to <https://slack.com/downloads>. Select "Windows" to download the installation file, and then open the program.

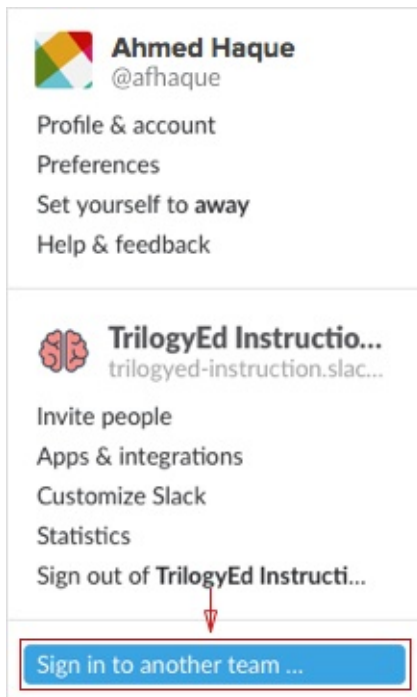


2. If you already use the Slack app, you just need to add our channel to your application.

- Click the header of your current Slack Channel.

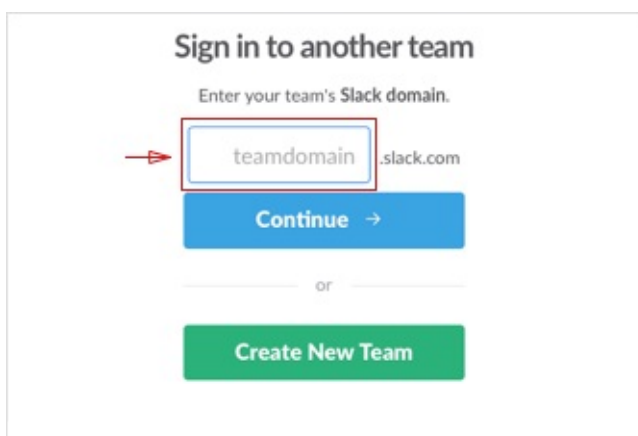


- Then select “Sign in to another team ...”



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.

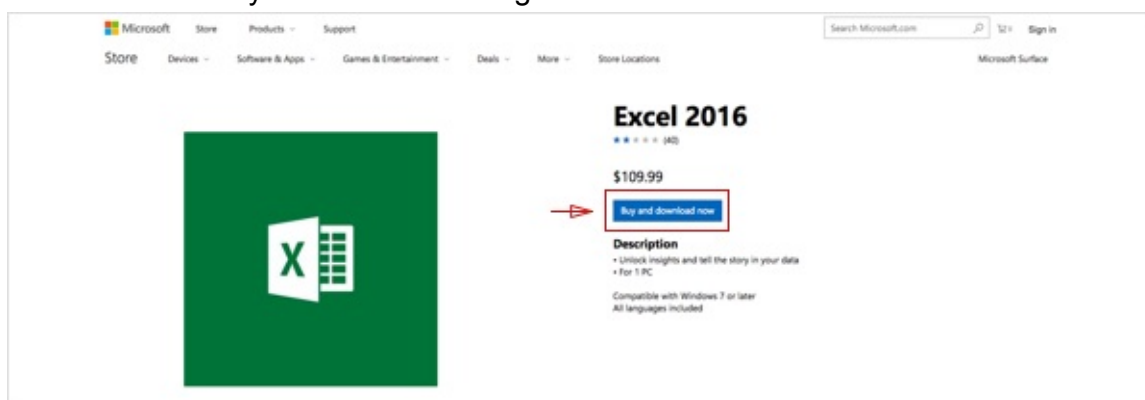




4. When you see the chatroom, you're finished.

## Microsoft Excel

1. Go to the [Microsoft Store](#) and click the Buy and Download button. Make sure it's Excel 2016 or later that you are downloading.



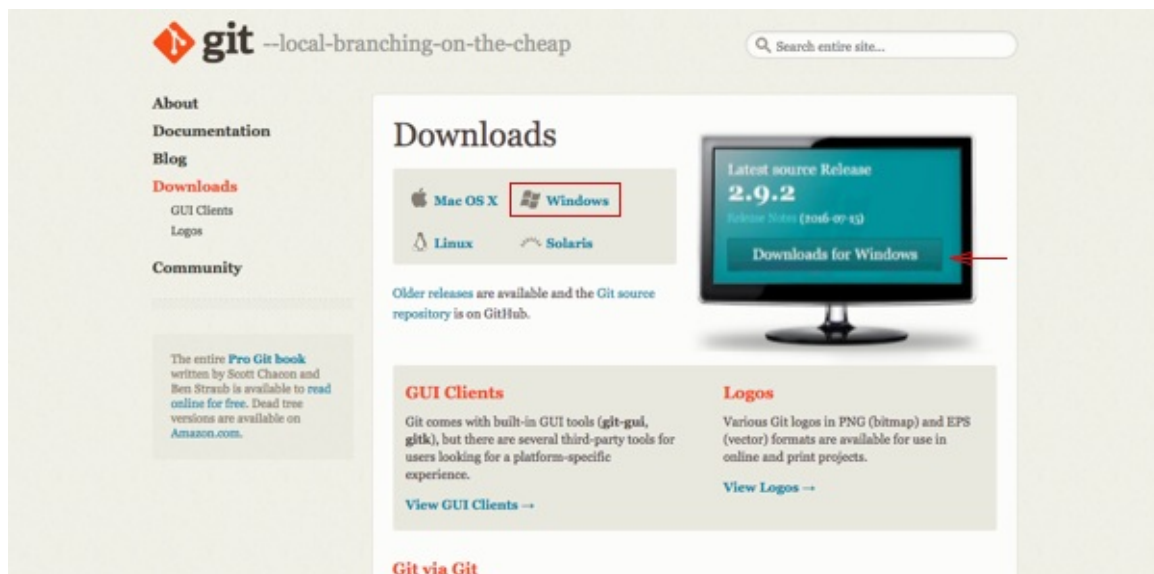
2. Follow the set up instructions after the download is completed.

## Git & Git Bash

Coders depend on this tool for version control; the process of logging the development of programs and applications. This comes in handy during collaborative programming, when teams of programmers change, add and remove code throughout a project's directory. This process would be chaotic without Git.

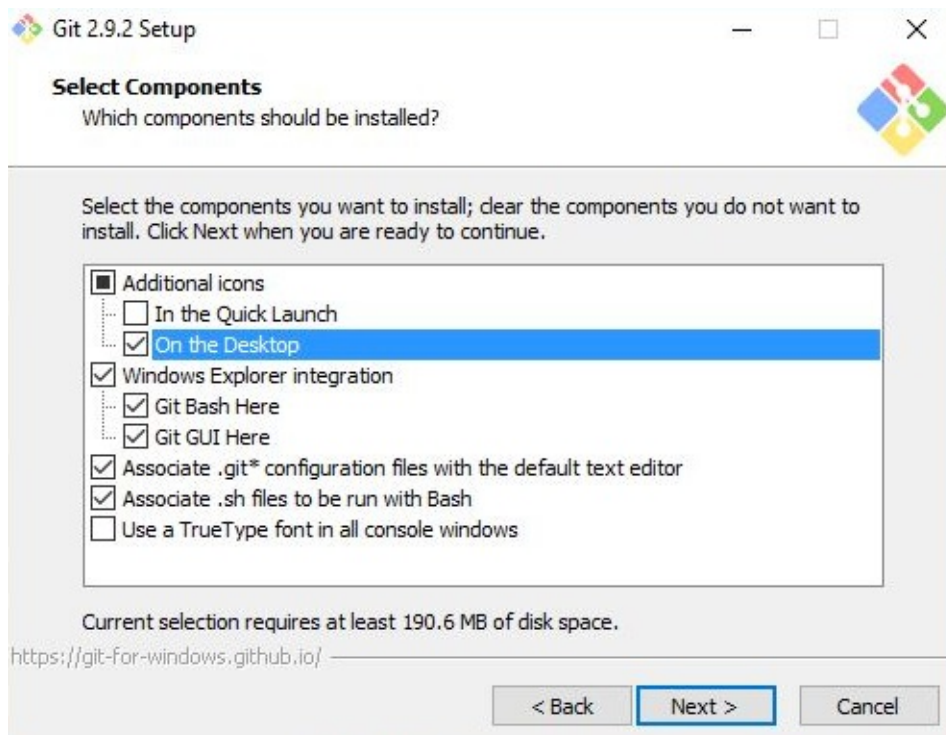
The installation also includes Git Bash or Bash for short. You'll be using this command line terminal throughout the course and during the rest of these instructions.

1. Go to Git download page: <https://git-scm.com/downloads>. Click on the download for your computer.

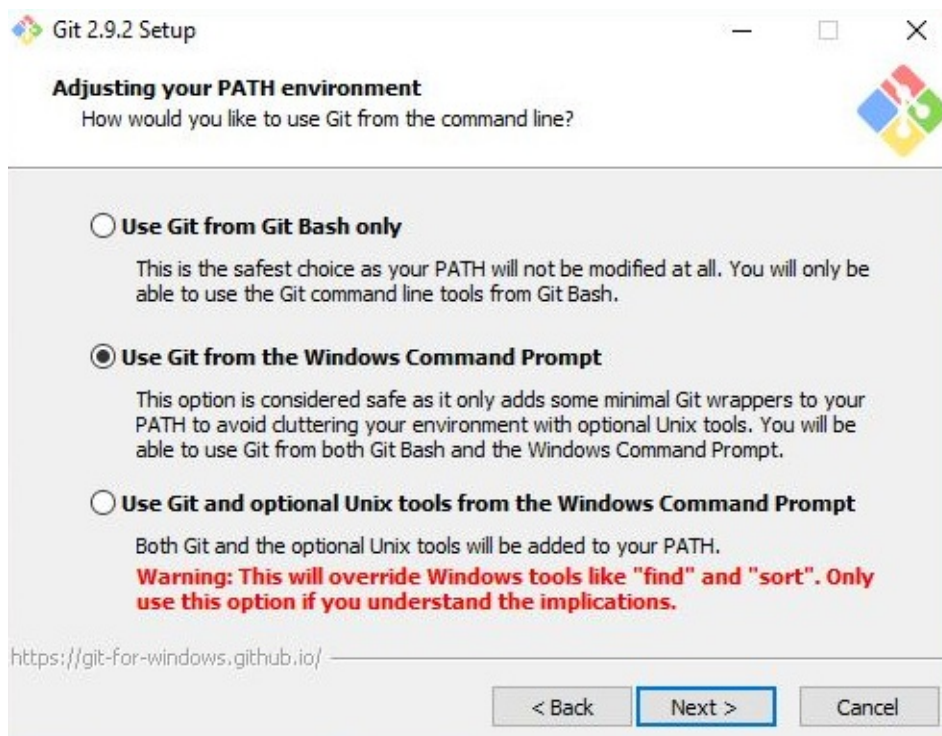


2. Run through the installation file. Make sure you check off the right boxes as shown in these four images.

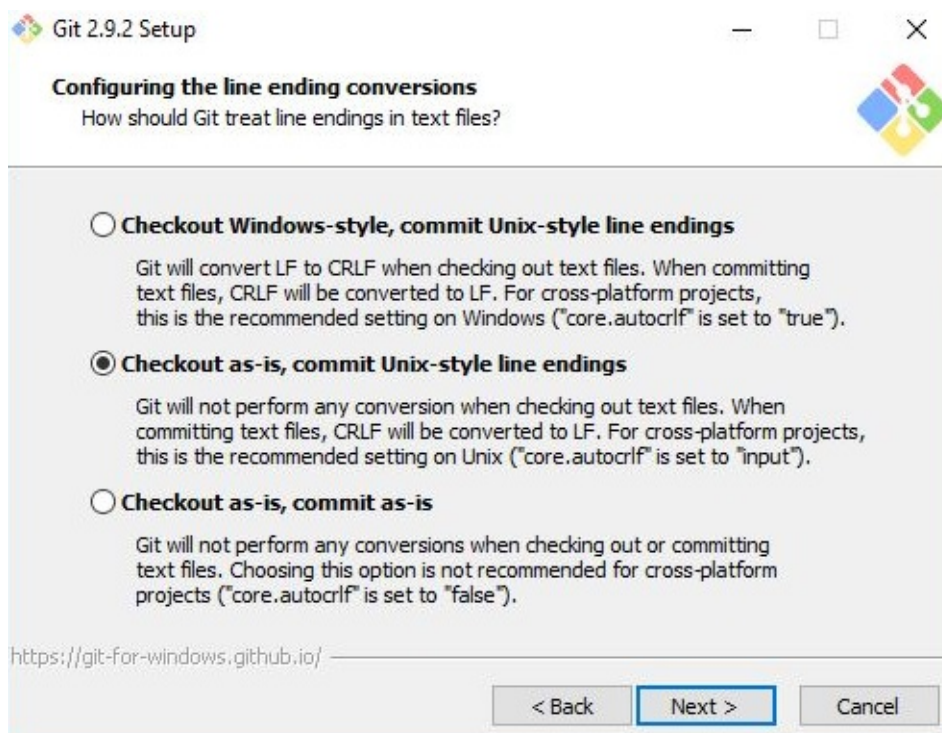
- Save Git to the desktop (this should save Git Bash to your desktop too).



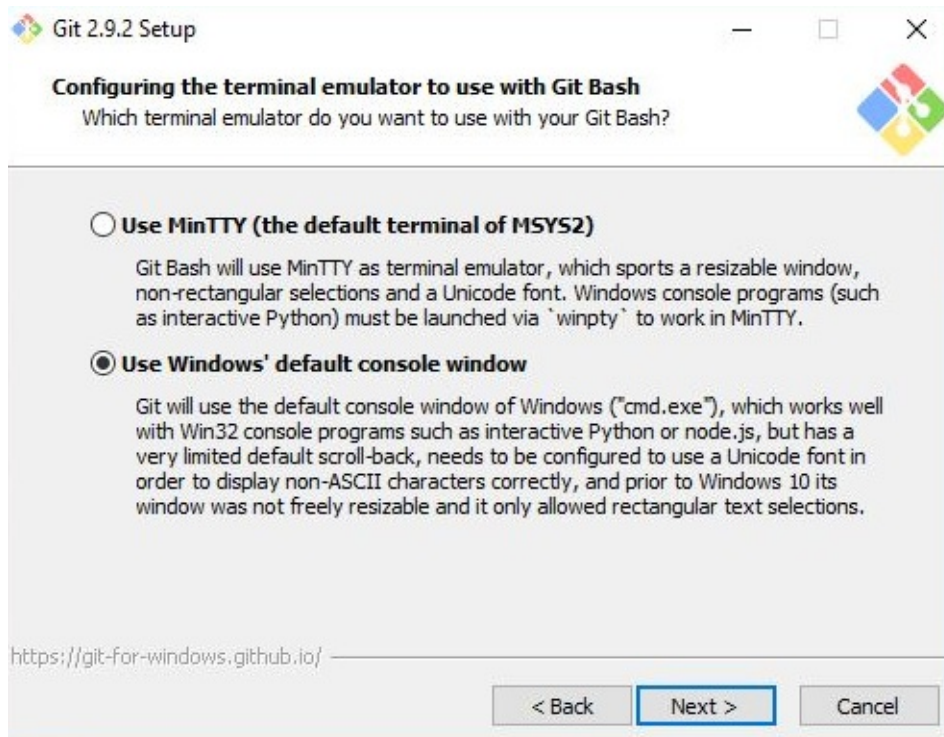
- Use Git from the Windows Command Prompt.



- Checkout as-is.



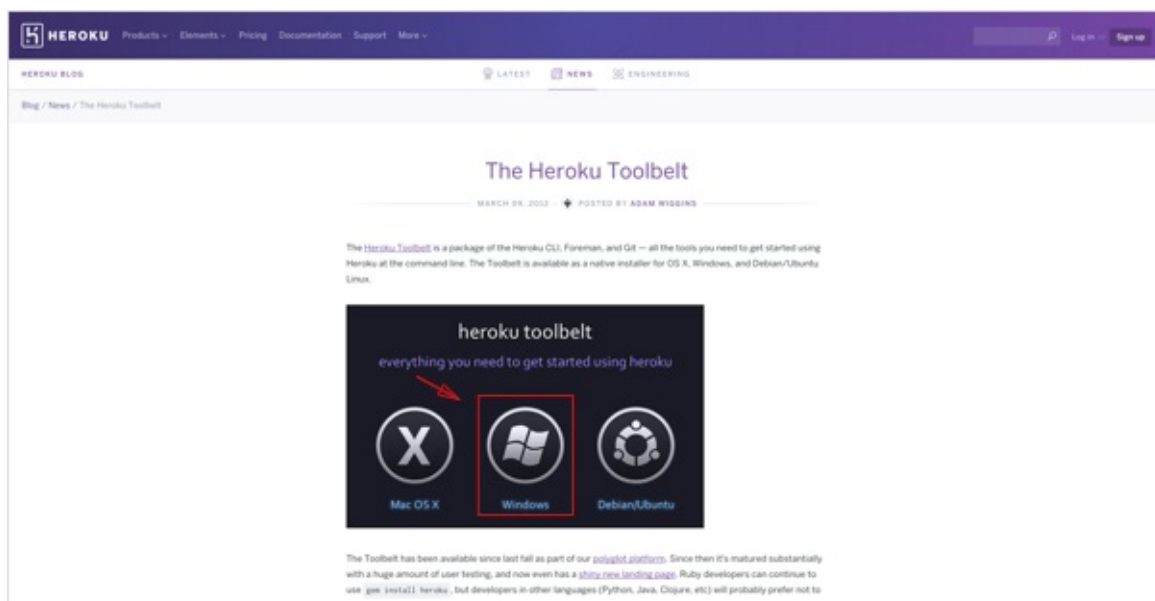
- Use Windows' default console window.



## 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. Go to <https://toolbelt.heroku.com>. Download the installer.



3. Go through the installation guide then open Command Prompt on your computer (not Bash).
  - Command Prompt or cmd.exe, comes preloaded on Windows operating systems

but it will be located in different locations depending on your version of Windows.

Use your OS's search feature for Command Prompt and it should pop up shortly.

4. Type `heroku login` into the command line, press enter. When prompted, enter the credentials from your Heroku account and then close Command Prompt.

## Mongo DB

Since installing MongoDB is somewhat involved and we won't be using MongoDB until rather late in the course, it's not necessary to have it installed on Day 1 - there will be time later for us to set it up together.

However, if you'd like to get a head start, [installation instructions for Windows user are here](#) and [installation instructions for OS X are here](#).

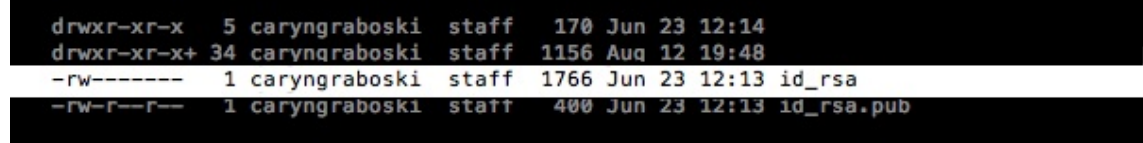
## SSH Key

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. If you haven't signed up for a GitHub account yet, you'll need to do so before moving on with these steps. Visit <https://github.com>.
2. Open up Bash.
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 Bash window(**copying and pasting will not work**):

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



```
drwxr-xr-x  5 caryngraboski staff  170 Jun 23 12:14
drwxr-xr-x+ 34 caryngraboski staff 1156 Aug 12 19:48
-rw-----  1 caryngraboski staff 1766 Jun 23 12:13 id_rsa
-rw-r--r--  1 caryngraboski staff  400 Jun 23 12:13 id_rsa.pub
```

- 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.

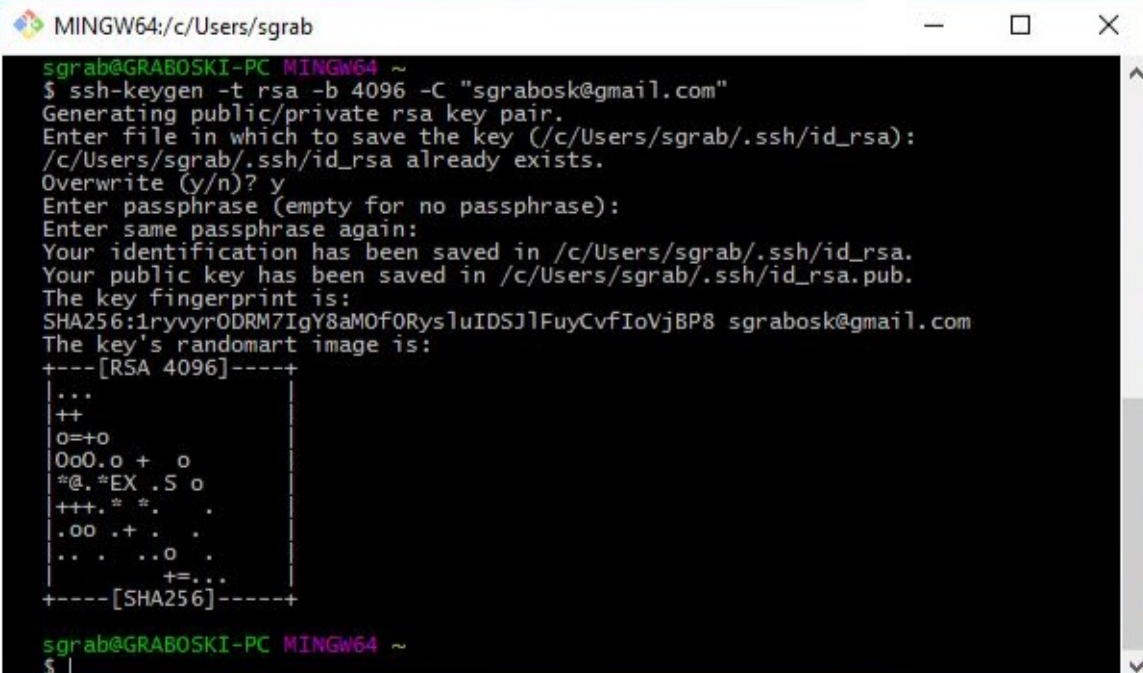
4. Type in this command along with your email to generate your keys

- `ssh-keygen -t rsa -b 4096 -C "YOURGITHUBEMAIL@PLACEHOLDER.NET"`

5. When asked to enter a file to save the key, just hit enter.

- 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:



```
MINGW64:/c/Users/sgrab
sgrab@GRABOSKI-PC MINGW64 ~
$ ssh-keygen -t rsa -b 4096 -C "sgrabosk@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/sgrab/.ssh/id_rsa):
/c/Users/sgrab/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/sgrab/.ssh/id_rsa.
Your public key has been saved in /c/Users/sgrab/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:1ryvyrODRM7IgY8aM0f0RysluIDSJlFuyCvFioVjBP8 sgrabosk@gmail.com
The key's randomart image is:
+----[RSA 4096]-----+
|...
|++
|o=+o
|OoO.o + o
|*@.*EX .S o
|+++.* *.
|.OO .+ . .
|.. . ..O .
|      +=...
+----[SHA256]-----+
sgrab@GRABOSKI-PC MINGW64 ~
$
```

7. For the next step, we need to use a tool called ssh agent to link our key with our machine. Let's test whether ssh-agent is working. Run this command in Bash:

- `eval "$(ssh-agent -s)"`
- If your Bash window looks like the below image, move onto the next step.



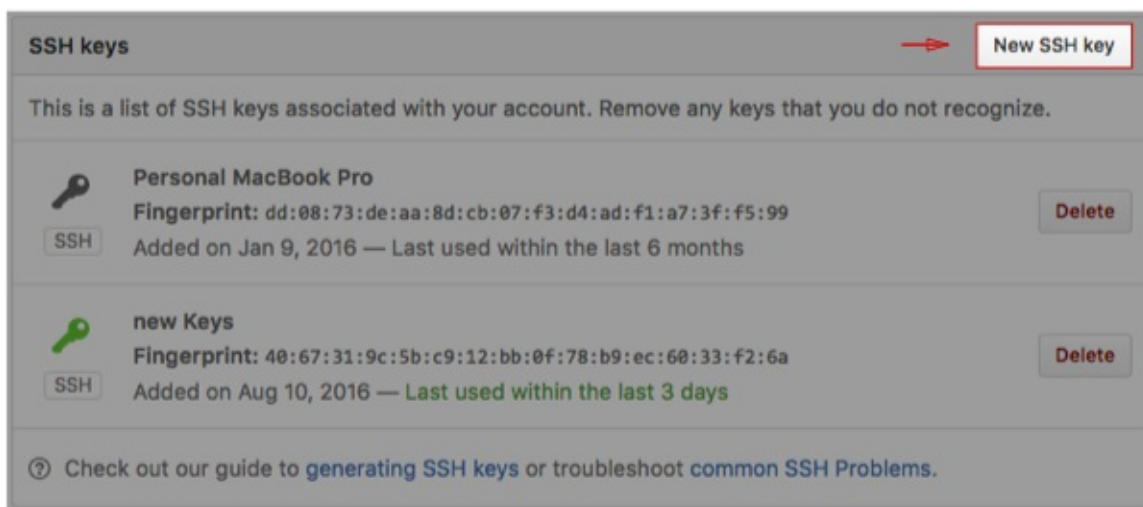
```
sgrab@GRABOSKI-PC MINGW64
$ eval "$(ssh-agent -s)"
Agent pid 12644
sgrab@GRABOSKI-PC MINGW64
$
```

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 password, just create a new one, starting with step 4.
10. We need to add the key to GitHub. Copy the key to your clipboard by entering this command:
- `clip < ~/.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 to your clipboard until you finish the next instructions. Otherwise, you'll have to repeat this step 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 Bash 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.

```
sgrab@GRABOSKI-PC MINGW64 ~  
$ ssh -T git@github.com  
The authenticity of host 'github.com (192.30.253.112)' can't be established.  
RSA key fingerprint is SHA256:nThbg6kXUpJWG17E1IG0CspRomTxdCARLviKw6E5SY8.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added 'github.com,192.30.253.112' (RSA) to the list of known hosts.  
Hi SGraboski! You've successfully authenticated, but GitHub does not provide shell access.
```

## 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 Git Bash.

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

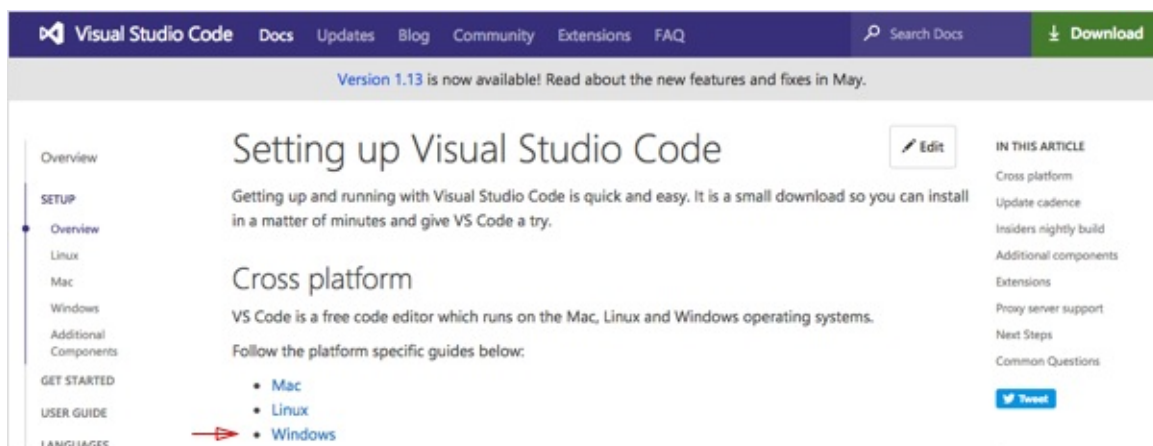
- 1 Open Git Bash.
- 2 Set an email address in Git. You can use your GitHub-provided `no-reply` email address or any email address.  

```
$ 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.

## VS Code

1. Head to the [Visual Studio Code](#) website and select the appropriate installation for your machine.
2. Download the Visual Studio Code installer for Windows.



3. Once it is downloaded, run the installer (VSCodeSetup-version.exe). This will only take a minute
4. By default, VS Code is installed under C:\Program Files (x86)\Microsoft VS Code for a 64-bit machine.

## **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**

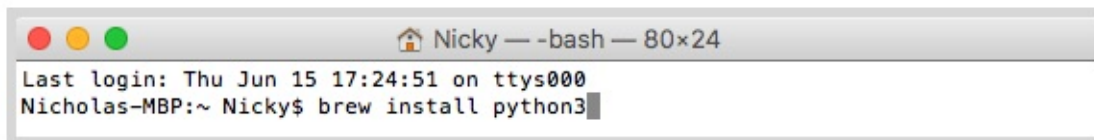
Data Boot Camp © 2017. All Rights Reserved.

# Assignment #1 - Curling Up With Python

---

## Installing Python Mac

1. Just run this command in your Terminal window: `brew install python3`



```
Nicky — -bash — 80x24
Last login: Thu Jun 15 17:24:51 on ttys000
Nicholas-MBP:~ Nicky$ brew install python3
```

## Installing Python Windows

1. Head to the [Python Home Page](#), click the latest Python 3 release - Python3.6.1.
2. Follow the install instructions.

## Installing Anaconda

1. Head to the [Anaconda Home Page](#) and click on the logo for your operating system to download the appropriate installer. Then, simply open the installer and follow the on-screen instructions. If you run into complications, the [detailed installation guide](#) will walk you through the process step-by-step.

## Creating and Managing Python Virtual Environments



### Virtual Environments - What and Why?

The Python docs tell us that Python applications will often use packages and modules that don't come as part of the standard library. Applications will sometimes need a specific version of a library because they may require that a particular bug has been fixed or the application may be written using an obsolete version of the library's interface.

This means it may not be possible for one Python installation to meet the requirements of every application. If application A needs version 1.0 of a particular module but application B needs version 2.0, then the requirements are in conflict and installing either version 1.0 or 2.0 will leave one application unable to run.

The solution for this problem is to create a virtual environment, a self-contained directory tree that contains a Python installation for a particular version of Python, plus a number of additional packages. Check out the [Python Virtual Environment Docs](#) for more info!

## Conda

There are other multiple ways to create virtual environments, but in this course we will be using Conda.

Conda is a package manager application that quickly installs, runs and updates packages and their dependencies. It allows you to easily set up and switch between environments on your local computer. Conda is included in all versions of Anaconda and Miniconda. Check out the [Conda Docs](#) for more info!

## Creating Virtual Environments

Verify that Conda is installed by checking its version with `conda --version`. Next let's update Conda using the update command `conda update conda`. If a newer version is available, go ahead and accept the update.

```
Nicholas-MBP:~ Nicky$ conda --version
conda 4.3.21
```

```
Nicholas-MBP:~ Nicky$ conda update conda
Fetching package metadata .....
Solving package specifications: .

# All requested packages already installed.
# packages in environment at /Users/Nicky/anaconda:
#
conda                4.3.21                  py36_0
Nicholas-MBP:~ Nicky$
```

1. Create your first virtual environment with the command `conda create --name dogs`. This will create a brand new environment with the name `dogs`.

```
Nicholas-MBP:~ Nicky$ conda create --name dogs
Fetching package metadata .....
Solving package specifications:
Package plan for installation in environment /Users/Nicky/anaconda/envs/dogs:

Proceed ([y]/n)? y

#
# To activate this environment, use:
# > source activate dogs
#
# To deactivate this environment, use:
# > source deactivate dogs
#
```

2. Activate your new environment with the command (Mac) `source activate dogs` , (Windows) `activate dogs` . You can also deactivate environments with (Mac) `source deactivate dogs` , (Windows) `deactivate dogs` . Note that your environment will live in `/envs/dogs` .

```
Nicholas-MBP:~ Nicky$ source activate dogs
(dogs) Nicholas-MBP:~ Nicky$
```

3. Let's create and activate another environment. Type `conda create --name cats` followed by (Mac) `source activate cats` or (Windows) `activate cats` .

```
Nicholas-MBP:~ Nicky$ conda create --name cats
Fetching package metadata .....
Solving package specifications:
Package plan for installation in environment /Users/Nicky/anaconda/envs/cats:

Proceed ([y]/n)? y

#
# To activate this environment, use:
# > source activate cats
#
# To deactivate this environment, use:
# > source deactivate cats
#

Nicholas-MBP:~ Nicky$ source activate cats
(cats) Nicholas-MBP:
```

4. Let's list out our new environments with the command `conda info --envs` . You should see both of your new environments plus your root `/home/username/miniconda` .

```
Nicholas-MBP:~ Nicky$ source activate cats
(cats) Nicholas-MBP:~ Nicky$ conda info --envs
# conda environments:
#
cats                * /Users/Nicky/anaconda/envs/cats
dogs                /Users/Nicky/anaconda/envs/dogs
root                /Users/Nicky/anaconda
```

5. Conda puts an asterisk next to your active environment. Again, type `conda info --envs` to verify which environment is active and then type (Mac) `source activate dogs` to change back to `dogs` or (Windows) `activate dogs` .

```
(cats) Nicholas-MBP:~ Nicky$ conda info --envs
# conda environments:
#
cats                * /Users/Nicky/anaconda/envs/cats
dogs                /Users/Nicky/anaconda/envs/dogs
root                /Users/Nicky/anaconda

(cats) Nicholas-MBP:~ Nicky$ source activate dogs
(dogs) Nicholas-MBP:~ Nicky$
```

## Additional Commands

Copy: `conda create --name fish --clone dogs` . This will make a copy of `dogs` named `fish` .

Delete: `conda remove --name fish --all` . This will delete `fish` .

---

## Copyright

Data Boot Camp © 2017. All Rights Reserved.



## Module #3 - Time to get Employable

---

### Career Services

Our Career Services team is built on the principle of empowerment. We are here to **empower you** to find a job using the skills you have learned in our program. We can't do the work for you, but we can motivate, guide, and equip you to reach beyond your comfort zone to find the right opportunity for you.

Don't forget that we can only help you based on the information you share with us; the more engaged you are, the more successful you will be!

---

### What to expect on the job hunt:

When transitioning into a new career, a reasonable timeline from starting applications to accepting a company's offer is 30-90 days. You'll spend time preparing your professional narrative and network just as you spend time honing your coding skills.

### What you DO need to succeed:

1. A professional narrative that showcases your strengths & passions.
2. Ability to showcase your previous skills as adding to your value as a technology worker.
3. A friendly demeanor - people hire people they like!
4. Ability to talk about technology - even technology you aren't familiar with - by relating it to concepts you do know. Yes, jargon matters. We need to call it an HTML element, and not a "thing".
5. Practice, practice, practice - the more networking events and interviews you complete, the more prepared you'll be for your next opportunity.

### What you DON'T need to succeed:

1. **"Perfect" looking code.** No one's code is perfect! Does it work as expected? You're golden!
2. **Mastering all technologies taught in the course.** You'll learn so much in this boot camp, it'll make your head spin. The skill you're really trying to pick up: learning how to teach yourself new technologies using online resources.
3. **Previous coding experience.** Many companies will let you learn on the job, and will

appreciate the unique skill set you bring to the table.

4. **Every skill listed on the job description.** Often, the job description isn't written by anyone familiar with technology, and it can be misleading. Apply anyway! Talk about your strengths, and the ability to learn quickly based on your boot camp performance!
5. **Certificate of completion from the boot camp.** Once we've approved your BootCampSpot (BCS) career profile - you're ready to start applying! Don't be afraid to start early, practice makes perfect, and you'll only get better the more interviews you complete.

---

## Create accounts (Required):

They're all free! Here are some resources that will get you started on the pathway to a new career:

1. [LinkedIn](#)

Join LinkedIn to get the latest news, insights, and opportunities from over 3 million companies. It'll act as a professional profile, and it has an activity feed full of wisdom from companies & individuals you connect with and/or follow.

2. [Stack Overflow](#)

Stack Overflow is the Wikipedia of code - question & answer style. It's like yahoo answers, except it has good answers. You can create questions, and answer them too!

3. [Angel List](#)

Where the world meets startups. Angel List helps investors, startups, and job seekers connect!

4. [Meetup](#)

Meetup brings people together in thousands of cities to do more of what they want to do in life. This site is chock full of opportunities to meet with others who are passionate about learning code - just head to the technology category!

5. [Medium](#)

Medium is home to the world's curious minds. It's a blogging platform full of technology content - start following [Free Code Camp](#), and browse other technology recommendations today!

6. [Gmail](#)

Gmail is email that's intuitive, efficient, and useful. 15 GB of storage, less spam, and mobile access. It comes with free storage on Google Drive which you'll use to submit some of your homework.

## 7. Master the Google Document

You will be using Google documents to submit the first several career homework assignments. We want to ensure you know how to make your document editable for your Career Services team to be able to provide timely and meaningful feedback.

Please watch this short [video](#) to become familiar with Google documents. Don't forget to try to create a shared document right after watching the video!

## Assignment (Recommended):

- [Connect Professional Dots!](#)

## Other Resources:

- [Stack Overflow Meta](#) - Stack Overflow, for beginners! The moderators are kinder and the community is more forgiving of newbies. Signing up for Stack Overflow does not sign you up for Stack Overflow Meta - they're separate!
- [Eventbrite](#) - Bringing the world together through live experiences. A great tool for finding recurring or single events.

## Final Notes:

Job hunting is stressful, and trust us, *no one likes it*. Don't worry - we'll be here to support you through learning the hard skills, and prepare you to succeed in the interview process. We need your participation and hard work - **but we'll get through this together**.

Good luck!

---

## Copyright

Data Boot Camp © 2017. All Rights Reserved.

## Assignment #2 - Connect Professional Dots!

---

### Overview:

This assignment is designed to dip your toe in the networking water, and talk to others in the technology field! Although we recommend that you complete this mission, it is not a requirement for completing the pre-work or course. It could however be very beneficial to you. Networking is one of the best ways to find your next opportunity!

### Instructions:

1. Read through the "[Time to get employable](#)" chapter of Pre-Work.
2. Browse [Eventbrite](#) and/or [Meetup](#) for an event that occurs **before the boot camp starts** that sparks your interest!
3. Attend the event, alone or with friends and talk to at least two people **you did not know previous to the event**.

### Note:

- No trick assignment here. We want you to meet people, have fun and reflect.
- 

### Copyright

Data Boot Camp © 2017. All Rights Reserved.

## Module #4 - Get Your Mind Right (Required)

As every successful person will tell you, "Attitude is everything."

For this reason, as you embark on the challenging endeavor ahead, it's critical that you align your attitudes and expectations with the realities to come. Don't underestimate the amount of mental and emotional discipline this bootcamp will require. Take the time to prepare yourself by reading through the following tips and suggestions.

P.S. Don't breeze through these! They are important!



### Major Keys to Success

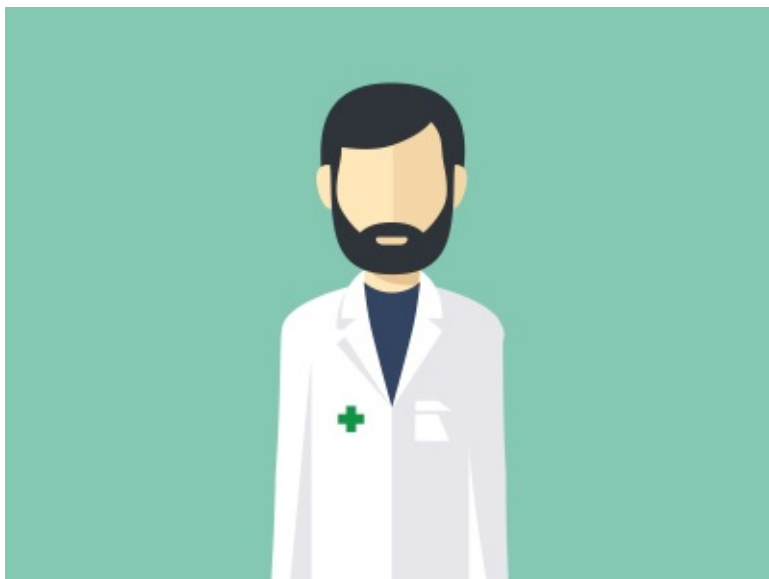
#### 1. A Little Humility Never Hurt Nobody...

Whether you believe it or not, this bootcamp will be one of the most challenging learning experiences of your life. For most of you, the skills being taught in the program will be fundamentally different from any you've been exposed to before. In fact, your past skills, schooling, and experiences are unlikely to help you become a successful Data Analyst. Instead, in order to succeed, you will need to channel your inner toddler and *learn to learn* again.

This will be a very challenging identity for many of you to take on. If you've had past academic or career successes in another trade or industry, you may find yourself resisting the idea that you can *fail* so dramatically at Data Analytics.

Yet, heed this advice now: "A humble learner is the most ready learner." Take each class as a new opportunity to be proven a fool and learn anew. Six months from now, you will marvel at the mastery you gained.

## 2. There is No Data Analytics Pill...



The greatest self-lie you can tell is that any program will teach you to analyze data. Instead, it's important to realize that, in order to succeed, *YOU* must own your learning experience.

Data Analysis, like any craft, requires diligent effort, constant experimentation, and a relentless desire to self-improve. These characteristics cannot be taught in any classroom. From Day 1 through Day 180 and beyond, remind yourself that *YOU* are responsible for your ultimate success. We will be here to guide, to encourage, to facilitate, to point out your mistakes, and to show you the path, but it's up to you to put in the effort.

Don't get discouraged! We know you can do it.

## 3. Put in the Hard Hours (At Least 20 Hours!)

According to author Malcolm Gladwell, successful people put in a minimum of 10,000 hours of deliberate effort to master their craft. As newcomers to the field of Data Analysis, be ready to put in your share of hours.

While the bare minimum to *survive* this program is 10 hours of outside class time, we've consistently found that those most successful put in closer to 20 hours of outside effort per week. At times, this number might even need to go upwards of 30 or 40 hours per week



week during more challenging topics.

Simply said, there is no substitute for long, hard hours.

Consider this time to be an investment in yourself, and know that for every hour you spend, you are guaranteeing yourself a better opportunity after graduation.

## 4. Patience Makes Perfect...

Tied to the previous suggestion is a second piece, often forgotten. Not only will learning to analyze data require many hours, it will also require many months (and frankly years) to master. Don't rush success!

For almost all of you, the first few months of the program will be particularly challenging. In fact, for many of you, this period will be one in which you doubt your eventual success through the program. Resist the urge to give in or to become hopeless.

Realize that learning this craft will require consistent effort, which will iteratively build your skills and understanding. What will seem challenging, confusing, and distressing in Week 3—will seem completely second nature by Week 24. Intensity is no *substitute* for *consistency*.

## 5. Not a Spectator Sport...

As every musician, painter, or craftsperson will tell you, one cannot learn a skill by simply reading, watching, or following along. Instead, it requires hours of deliberate practice. In the same way, data analysis, just as any other skill, will require you to step away from the spectator stands and to enter the fray yourself.

Know this now: going to class, watching tutorials, reading articles, or participating in other passive activities will only get you so far. You must spend significant hours actually *working* to succeed in this bootcamp. As a benchmark, consider spending 70–80% of your outside class time completing data related activities and homework. Only the remaining 20% is for other passive activities.

*But I don't feel ready!*

Often, we've seen that students who struggle the most are the ones most intent on *learning everything* before they start analyzing data themselves. Unfortunately, this is a recipe for stagnation. You will not be successful in this program unless you force yourself to work through confusion: making errors, chasing them down, and learning from your mistakes.

This bootcamp is a mudslide. There's just no way to avoid getting your hands dirty.

## 6. Crumple the Paper Tiger...

### *Am I doing this right...?*

The five most hated words of every instructor.

These words aren't hated because your instructor is unwilling or is unable to help. Rather, they're hated because they suggest that a student is afraid to *try* something on their own. As you enter the classroom, learn to abolish this phrase from your vocabulary.



Instead, learn to try things on your own, to test what you can, and to do a bit of debugging first. *Then* turn to your instructor and say, "I tried such and such. It didn't work so I tried such and such. But it still didn't work. What should I try now?"

Notice the change in tone. Instead of timidly asking for assurance, give yourself permission to *just try*.

The best way to fail is on your own terms.

## **7. Find a Squad...**

While in this program, make every effort to find friends, to form study groups, and to work together in and out of class. Sometimes, the fastest way to overcome a bug or to understand a challenging concept is to have another pair of eyes or to have another perspective.

## **8. Master the Art of Google Fu...**



One of the greatest surprises to students entering the program is the amount of *Googling* they are asked to do. This isn't because your teacher is lazy or the curriculum is unplanned—far from it!

As every professional Data Analyst will tell you, Data Analysis isn't about *memorization*. Instead, it's about curating bits and pieces of knowledge, scattered across the web and accumulated in various documentations, forums, and QA websites.

To become a good data analyst requires an ability to work through problems and to quickly research solutions ascertained by others. The field is constantly changing and new tools are always on the horizon. As you will find, every good data analyst is a great friend of Google (and Stack Overflow).

## 9. Plan Often...

The best first step in *every* data challenge is to formulate a plan. Break down the complex task in front of you into discrete, bite-size challenges. Once you have a plan, write it out and always refer back to it.

Fundamentally, every task in Data Analysis can and should be, broken into smaller tasks. Don't try to bite off everything at once! You'll get lost in your own mind games.

## 10. Fixing Things Takes Time...

One of the most frustrating aspects to new students of Data Analysis is the sheer amount of time it takes to troubleshoot (or debug) issues. At times, it might even feel like *fixing* an issue is taking 3–4 times as long as conceiving the original solution.

*Know in advance that this is completely normal.*

Instead of seeing these spent hours as a *distraction*, learn to see them as a critical part of the learning process. Each bug you pursue is a lengthy lesson that adds to your arsenal of understanding.

## 11. Self-Care is Key...

While we've probably traumatized you with all the talk of challenges, of difficulties, and of effort, we do \_want \_you to take care of yourself. Throughout the program, be sure to sleep, to exercise, and to eat nutritional meals. These moments of self-care are extremely important for your mind to be healthy. Taking breaks is encouraged!

In fact, you will find that that some of the best problem-solving happens during breaks. "Sleeping on a problem" is often a very real solution to your most challenging issues. Try to walk into class each day, ready and refreshed for new learning. We want happy, energized people in our classrooms. Not dead robots.

## 12. Be a stellar student

As a student, we'll encourage you to collaborate with your classmates, and participate in study groups in and out of the classroom. We'll ask you to look for examples in the curriculum and online as you're learning. This is good practice for a Data Analyst as you'll do the same on the job.

Just make sure you're doing your own work, and submitting applications you can be proud of. If you feel like you are borrowing too much, reach out to your SSM for guidance.

There's value in the work you're doing, it translates to hard skills, and you truly get out what you put into this program.

## 13. You Can Do This!

This last piece of advice is the most important. Remind yourself each day that you *can* do this. We've seen through many classrooms, students from all backgrounds, experiences, and personal situations persevere and succeed through this program.

During the tough times, dig deep into your own personal motivation and remind yourself why you entered the program. Let this be the fuel that you use to keep on.



You have *everything* it takes to learn this craft and to gain the opportunities that come with it. It may feel challenging at times, but remind yourself of past challenges you've overcome. Your future self will thank you for all that you endured.

---

## Video (Recommended)

- [Grit: The Power of Passion and Perseverance](#)
  - [James Clear: Successful Habits](#)
- 

## Copyright

Data Boot Camp © 2017. All Rights Reserved.

# Module #5 Let's Get To Work (Assignment Required)

## So You Wanna Be A Data Analyst...

Well, you've come to the right place! Over the course of the next six months, you will be working on developing the necessary skills to become a Data Analytics Pro.



## Instructions:

1. In Google Drive, create a file titled "Data Analytics and I".  
\* If you haven't created a Google Drive account, please go back to Module # 2 and complete the steps.
2. Write 200 words about what you want to learn in this class and what you are most excited to learn.
3. Write 100 words about what these skills will do for your career! If you don't have a career in this field yet- no problem! Write about how these skills will help you reach your goal.
4. **Once you complete this assignment- copy the Google document link into BootcampSpot!**

## Copyright

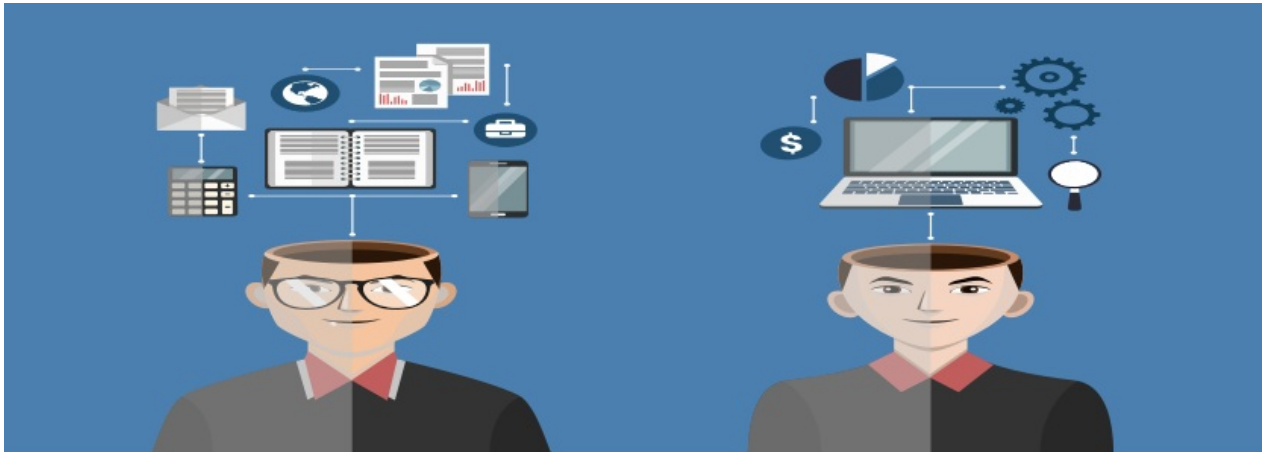
Data Boot Camp © 2018. All Rights Reserved.



## Want More?

---

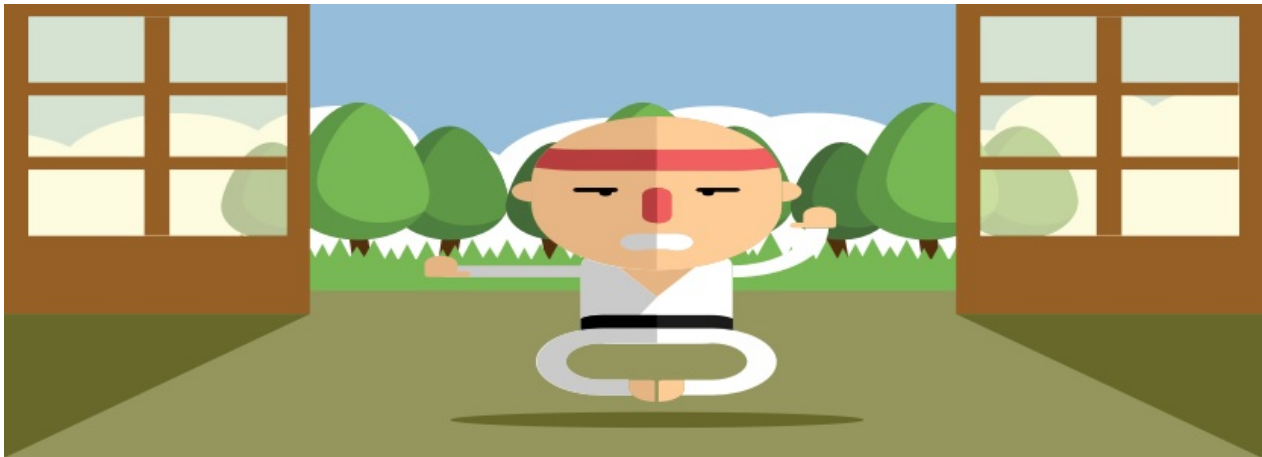
### Supplemental Resources for the Curious of Mind



Knowledge is power. If you are interested in getting your feet wet, check out some of these free online courses.

- [Intro to HTML and CSS](#)
- [Intro to JavaScript](#)
- [JavaScript Basics](#)
- [Data Visualization in Tableau](#)
- [Introduction to Python](#)
- [Intro to Computer Science](#)
- [Intro to Data Analysis](#)
- [How to Use Git and GitHub](#)
- [Intro to Relational Databases](#)
- [Database Systems Concepts and Designs](#)
- [Data Wrangling with MongoDB](#)
- [Object-Oriented JavaScript](#)
- [Asynchronous JavaScript Requests](#)
- [Google Maps APIs](#)
- [Data Visualization and D3.js](#)
- [Intro to Hadoop and MapReduce](#)
- [Intro to Machine Learning](#)
- [HTTP & Web Servers](#)

## The Art of Google-Fu



Undoubtedly through your career journey you will have questions, and there may not always be a more experienced person there to help you with the answers. This is why you need some Google-Fu in your life.

Google-Fu is defined as “skill in using search engines (especially Google) to quickly find useful information on the internet”. You may be thinking this sounds easy enough.....however there are some "tricks to the trade" that will make your life much easier.

For Instance:

- Simple is better than complex - Google searches should be simple but broad enough to bring you the anticipated results. For example “JavaScript remove key from object”.
- Spending time going through to page 10 of a search is not necessarily a waste of time - a search may not always readily give you the answer you are looking for but it may just give you a keyword that will help you get there. Finding these little gems help lead you to the mine.

For some other really cool tips, check out [this article](#)!

---

## Copyright

Data Boot Camp © 2017. All Rights Reserved.