**Module #5 - Machine Ready (Required)**

**More Than Programming...**

As every developer knows, becoming proficient at web and software development requires more than just a grasp of programming languages alone. Instead, successful developers must also possess a comfortable familiarity with a wide assortment of tools and technologies. These tools enable software to be created, to be collaborated over, and to be shared with users around the world.

In this class, you can expect to be exposed to a wide range of tools with names you've never heard—tools like *Heroku*, *Git*, *Emmet*, and *Robomongo*. 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 a number of tools already installed. This will ensure that you are ready to start coding right from the start. The purpose of this module is to walk you through the process of installing all your tools and to give you a brief primer on the roles they play.

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

* Google Chrome
* Slack
* Sublime Text Editor
* Sublime Package Manager
* Emmet Sublime Package
* Git
* Git Bash (Windows)
* Terminal (Mac, Pre-Installed)
* Homebrew (Mac)
* Heroku Toolbelt
* Node.js
* MySQL Workbench
* MAMP

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

* LinkedIn
* GitHub
* Stack Overflow
* Slack

**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.

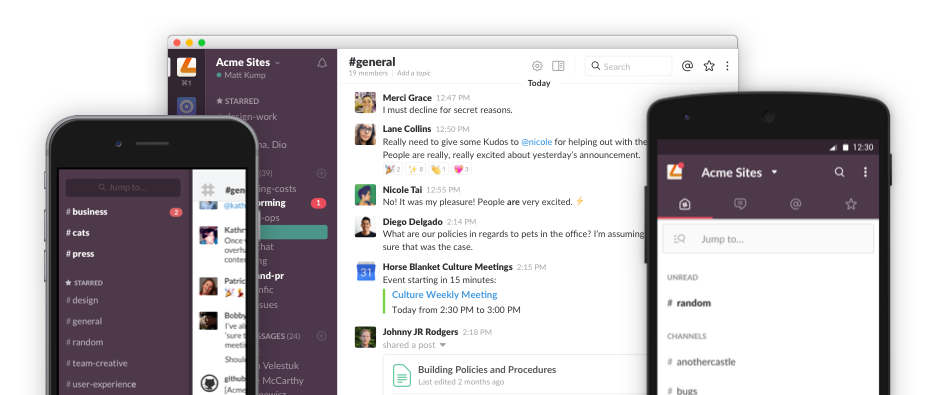
**Google Chrome:**

This one is pretty 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 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.)



**Sublime Text:**

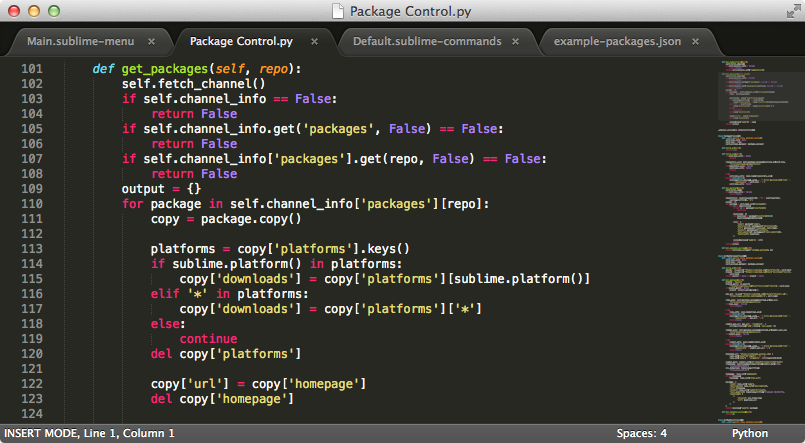
Oh, the power of Sublime Text! A little program that does so much!

Sublime is a free, open-source *text-editor*. 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...



...what we've really done is just created a block of text. There are funny symbols in there like those angle brackets and whatnot, but at its 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 Sublime Text, 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 Sublime can give us a more visually intuitive understanding of the code through indicative coloring, through *smart* tabs, and through autocomplete functionality. The end result is that creating the above block of HTML becomes a more natural process and becomes one that can be debugged more quickly.



Additionally, as you will find in this class, Sublime Text is also powerful in its ability to be extended. This means that we can easily incorporate free add-ons that enable Sublime to make the process of coding even easier than before. Which brings us to....

**Package Manager:**

As you proceed through life as a developer, you will be incorporating a number of Sublime extensions on the fly. And *because* there are so many packages, you will actually need a package to manage these other packages.

In this case, Package Manager is the answer. This Sublime package makes it easy to find, to download, and to install other packages into your Sublime editor quickly and painlessly, allowing you to get back to coding without strain.

**Emmet:**

To many developers, Emmet is the single greatest Sublime package of all time. As you will find in this course, typing out code can be a rather tedious process, and often you are retyping the same bits of HTML, over and over again. Emmet adds a number of supremely useful keyboard shortcuts to Sublime, enabling you to create HTML content in a tenth of the time. You'll have a chance to revisit Emmet later in the course.

**Git Bash / Terminal:**

Git Bash (PC) and Terminal 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.

**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.

**Homebrew:**

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 / 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.

**Node.js:**

Oh, Node! (Get the pun?)

Node.js is a Javascript library that we'll be using extensively for the back-end of our applications. Don't worry if that doesn't make much sense yet. We'll be spending effectively half the program working on Node. You'll become very proficient in its use and in its utility by the end of the course.

**MySQL Workbench:**

MySQL is a popular database used in many web applications. In essence, a database serves as a warehouse for information. Throughout the class, the applications you create will be retrieving, creating, updating, and deleting records from these databases. We'll be covering MySQL in more depth during Week 10.

**MAMP:**

MAMP is a popular *stack* of tools that is used by many web developers. In our case, we'll install it as a back-up, in case your machine runs into any issues with MySQL. (It happens. Tools are finnicky sometimes.)

**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! As future coders, 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.

**Assignment (Required):**

* [Get Yo' Tools Installed on Windows](https://the-coding-bootcamp.gitbooks.io/pre-work-book/content/assignment5_win.html)
* [Get Yo' Tools Installed on Mac](https://the-coding-bootcamp.gitbooks.io/pre-work-book/content/assignment5_mac.html)

**Copyright**

Coding Boot Camp © 2016. All Rights Reserved.

**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](https://slack.com/) as well. If you haven't received an email from us yet with the section's domain name, contact your Student Success Manager to assist you. You should receive an invite 2 weeks prior to class start.

**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 at <https://www.google.com/chrome/browser/desktop/>.
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 hereso set this program up as soon as you can.

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.

**Sublime Text**

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

1. Download the version listed with “OSX” in its name.
2. Run the dmg file. A window with the Sublime Text program should appear in a window, next to a shortcut of your Applications folder. Drag Sublime into the folder.

**Sublime Package Control**

With this plugin, you can incorporate new features into Sublime without opening your web browser; just enter in the name of the tool you want to add, and Sublime will install it.

1. Open Sublime Text. Open the console by pressing command + `
2. Go to <https://packagecontrol.io/installation>.
3. Copy the code from the Sublime Text 3 box, and paste it into the console. Hit return to run the code.

**Emmet**

This Sublime plugin lets developers drop boilerplate code into their programs with certain key combinations; this will cut out a chunk of the monotony from your work.

1. Close and reopen Sublime Text Editor.
2. Hit command + shift + p to open the options menu. Then type Package Control Install, and click the option highlighted below:
3. In the Package Control menu, type Emmet. Click on the highlighted option to install the plugin.
4. Let’s make sure you installed the tool correctly.
   * Close and reopen Sublime.
   * Then hit command + shift + p to open the options menu again.
   * If you type Emmet and you see a slew of Emmet options, then congrats! You’re ready to move onto Git.

**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 Sublime 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.
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.

**MySQL**

This database management system allows developers to create incredibly dynamic web applications for a variety of purposes—user-specific content, data analytics, so on and so forth.

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

**Sequel Pro**

Developers use this MySQL GUI client to see their databases in a human-readable form. This sort of tool simplifies abstract concepts by providing developers with a *graphical user interface*, or GUI for short (pronounced *gooey*, by the way).

1. Head to <https://sequelpro.com/download>. Click the "Download Now" button.
2. Open the dmg file. A Window should pop up on your computer with the Sequel Pro program. Drag this into your Mac’s Applications folder.

**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:

**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 © 2016. All Rights Reserved.

**Assignment #5 - # Get Yo' Tools Installed (Windows)**

**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](https://slack.com/) as well. If you haven't received an email from us yet with the section's domain name, contact your Student Success Manager to assist you. You should receive an invite 2 weeks prior to class start.

**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 at <https://www.google.com/chrome/browser/desktop/>.
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.

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.

**Sublime Text**

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

1. Go to <https://www.sublimetext.com/3>. Download the 64-bit Windows version, and run through the installation program.

**Sublime Package Control**

With this plugin, you can incorporate new features into Sublime without opening your web browser; just enter in the name of the tool you want to add, and Sublime will install it.

1. Open Sublime Text. Open the console by pressing control + `.
2. Go to <https://packagecontrol.io/installation>.
3. Copy the code from the Sublime Text 3 box, and paste it into the console. Hit enter to run the code.

**Emmet**

This Sublime plugin lets developers drop boilerplate code into their programs with certain key combinations; this will cut out a chunk of the monotony from your work.

1. Close and reopen Sublime Text Editor.
2. Hit control + shift + p to open the options menu. Then type Package Control Install, and click the option highlighted below:
3. In the Package Control menu, type Emmet. Click on the highlighted option to install the plugin.
4. Let’s make sure you installed the tool correctly.
   * Close and reopen Sublime.
   * Then hit control + shift + p to open the options menu again.
   * If you type Emmet and see a slew of Emmet options, then congrats! You’re ready to move on to Git.

**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 install 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 into the command line, and then press enter. When prompted, enter the credentials from your heroku account, and then close Command Prompt.

**Node.js**

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

1. Go to the Node.js site: <https://nodejs.org/en>. Click the download button, and run through the installation file.

**MySQL**

This database management system allows developers to create incredibly dynamic web applications for a variety of purposes—user-specific content, data analytics, so on and so forth.

1. Go to MySQL for Windows: <https://dev.mysql.com/downloads/installer/>. Click the button in this image to download MySQL’s installer.
2. Open and run through the installer, making sure you follow these images:
   * Server only.
   * Click "Execute" here.
   * Set your root password. Store it somewhere safe too; you'll need it later in the course.

**MySQL Workbench**

Developers use this MySQL GUI client to see their databases in a human-readable form. This sort of tool simplifies abstract concepts by providing developers with a *graphical user interface*, or GUI for short (pronounced *gooey*, by the way).

1. Make sure you’ve installed MySQL before setting up the MySQL Workbench.
2. Go to the MySQL Workbench site: <https://dev.mysql.com/downloads/workbench>.
3. Scroll down, and select the 64-bit Windows version of the software. Download it, and run through the installer.

**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 Windows, and click download.
2. Run through the installer.
   * When prompted, make sure you deselect MAMP PRO; that’s a paid service, and you won’t need it.

**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:
     + 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:
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.
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.

**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 © 2016. All Rights Reserved.