* [Docs](https://coding-bootcamp-fsf-prework.readthedocs-hosted.com/en/latest/) »

* Module 2 - Installing Your Tools (Required)

**Module 2: Install Your Tools**

Proficiency in web and software development requires more than just a grasp of programming languages. Developers must also be familiar with a wide assortment of tools and technologies that allow software to be created and shared with users around the world.

This course will expose you to a wide range of tools, some (or all) of which you may be unfamiliar with—tools with names like Heroku, Git, Emmet, and Robomongo. Although they may be overwhelming at first, these tools will soon become as familiar to you as a scalpel is to a surgeon.

The purpose of this section is to acquaint you with each of these tools and walk you through the process of setting up all necessary accounts and installing the required programs on your machine.

**Ready for Action!**

Below is a list of the tools you should install before your first class to ensure you're ready to dive in on Day 1.

* Google Chrome
* Screencastify
* Slack
* Visual Studio Code
* "Open in Browser" Visual Studio Code extension
* Git Bash (Windows only)
* Terminal (Mac only; preinstalled)
* Git and GitHub
* Xcode (Mac only)
* Homebrew (Mac only)
* Heroku Toolbelt
* Node.js
* MySQL Workbench
* SSH keys

You should also create accounts for the following:

* LinkedIn
* GitHub
* Stack Overflow
* Slack

If you encounter any issues during setup, don't worry. Your instructional staff will help you troubleshoot any errors and answer your questions on the first day of class. Just sit tight until then.

**An Overview of Your Tools**

Before installing your tools, take a moment to examine each of them in more detail to better understand the role they will play in the course.

**Google Chrome**

This is the web browser we'll be using to quickly determine whether our code is working. Google Chrome has a number of tools that make it an ideal platform for coding, so if you are currently using a different browser, we encourage you to switch to Chrome.

**Screencastify**

Screencastify is a free, lightweight extension for Google Chrome. We will be using this screen recorder to easily capture our desktop, browser, webcam, or certain applications.

**Slack**

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

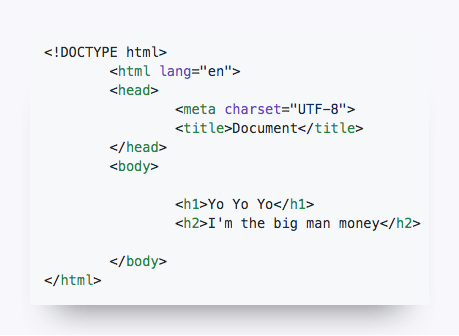
We will use Slack to send code snippets during class, relay important announcements, and facilitate group exercises. You will receive the link to your class-specific channel during orientation. Though there is a Slack web client, for this course you should have the program installed on your machine.

**VS Code**

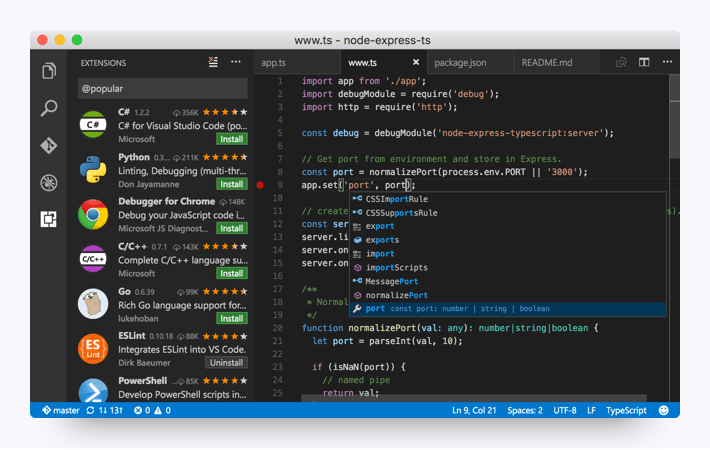
Oh, the power of VS Code! A little program that does so much!

Visual Studio (VS) Code is a free text editor that runs on Mac, Linux, and Windows operating systems. For developers, text editors are like the cozy pillow on which they rest their heads.

At a fundamental level, 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 created a block of text. There are some funny symbols in there, but at its most basic level, it's just text.



For a simple text editor, this is where the comprehension stops: our block of HTML remains a block of text. But for more powerful text editors like VS Code, these blocks of text are immediately recognized as code (as long as we include the right file extension). VS Code can provide a more visually intuitive understanding of the code through indicative coloring, smart tabs, and autocomplete functionality. As a result, creating HTML like the block above is a more natural process and can be debugged more quickly.



**Open in Browser (VS Code Extension)**

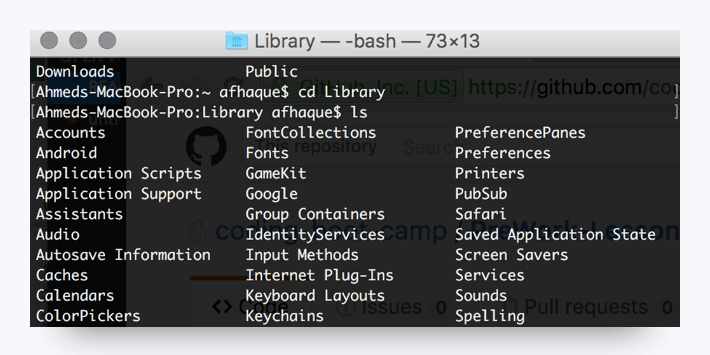
As you will find in this course, VS Code is powerful in its ability to be extended through the use of plug-ins. This means that we can easily incorporate free add-ons that enable VS Code to make the process of coding even easier than before.

For now, the only extension we recommend installing with VS Code is the Open in Browser extension. This will allow you to open HTML files you are editing in VS Code in your web browser without having to go through File Explorer (Windows) or Finder (Mac).

**Git Bash and Terminal**

Git Bash (for Windows users) and Terminal (for Mac users) 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 that there are no pictures or visuals. It's just a box with text.

*Uh, why would I want that?* Over time, you'll come to understand that, in many situations, using a command-line interface can be faster and more effective than relying on the operating system's graphical user interface (GUI). You'll get plenty of exposure to the command line at the beginning of the course.

**Git and GitHub**

Code files are largely collaborative, as developers are constantly building on each other's work. Git is a version-control system that offers a specialized set of strategies for orchestrating this collaboration, and GitHub stores these collaborative actions online. You can think of GitHub as a sort of Dropbox for coders. It offers a central place for developers to upload their code, view revision history, and 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.

**Xcode (Mac Only)**

Xcode is a development suite exclusive to Mac. We will primarily be using Visual Studio Code in this course but installing Xcode will set up some of the other required boot-camp programs including Git, which coders depend on for logging the development of programs and applications.

**Homebrew (Mac Only)**

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

**Heroku and Heroku Toolbelt**

In this course you will be creating websites on your computer, and you will use Heroku to put that website online for the world to see.

Hosting platforms like Heroku in effect serve as a dumping ground for web applications. These platforms take your web applications, activate their code, and then assign them a URL so that every internet user has access to them. You'll learn a lot about how this works toward the tail end of the course.

As part of this prework, 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**

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 utility by the end of the course.

**MySQL and 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 cover MySQL in more depth later in the course.

MySQL Workbench is a MySQL GUI client developers use 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).

**SSH Keys**

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.

**Ready, Set, Go!**

Now it's time to collect your tools and begin. Setup guides for both Mac and Windows users are provided in the links below. Follow the instructions closely and do your best with the information you have. (Yes, we know there is a lot to install.)

One bit of advice: Throughout the course, you will frequently need to install and use unfamiliar tools. Resist the urge to stop and ask, "Am I doing this right?" Instead, trust your instincts and take your best stab at it. If you get lost, we'll get you the help you need right away. You can do this!

**Activity**

* [Get Your Tools Installed (Windows)](https://coding-bootcamp-fsf-prework.readthedocs-hosted.com/en/latest/modules/assignment5_win/)
* [Get Your Tools Installed (Mac)](https://coding-bootcamp-fsf-prework.readthedocs-hosted.com/en/latest/modules/assignment5_mac/)

[Next](https://coding-bootcamp-fsf-prework.readthedocs-hosted.com/en/latest/modules/chapter3/)[Previous](https://coding-bootcamp-fsf-prework.readthedocs-hosted.com/en/latest/modules/chapter1/)

# Get Your Tools Installed (Windows)

You will be installing the required tools and software for the course. There's a lot to get through, so buckle in and get ready!

## Before You Begin

First, create accounts for the following services, which you'll need throughout the course. Don't just create logins; job recruiters often scour these sites in search of job candidates, so be sure to provide at least a headshot and contact information.

* [LinkedIn](https://www.linkedin.com/)

**Note:** You should create a full profile highlighting your skills and work experience, and include a headshot.

* [GitHub](https://github.com/)
* [Stack Overflow](http://stackoverflow.com/)

In addition, be sure to accept the invite for your section on [Slack](https://slack.com/). You will receive the link to your class-specific channel during orientation. If you don't receive this email, contact your Student Success Manager for assistance.

## Tool and Software Installations

Follow the instructions below to complete the installation process for all of the required tools.

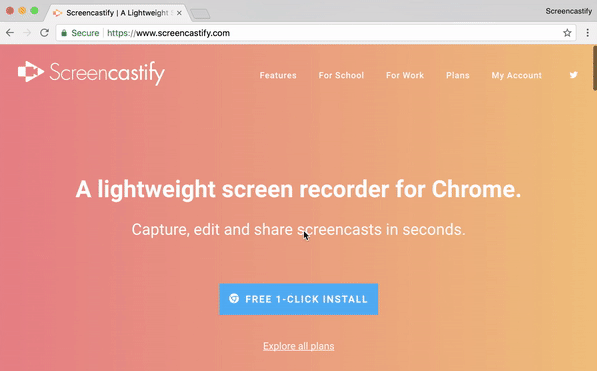
### Google Chrome

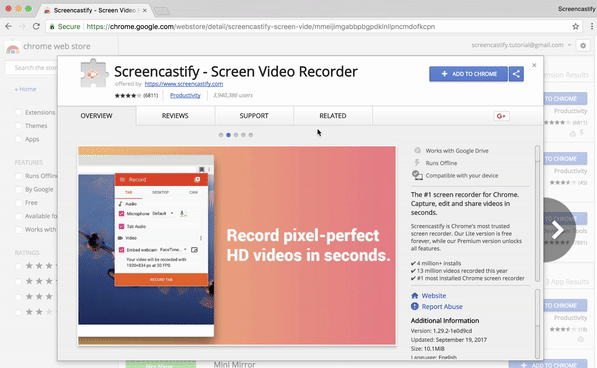
1. If you don’t already have Chrome installed, visit the download page [here](https://www.google.com/chrome/browser/desktop/index.html).
2. Download, open, and run the installation file.

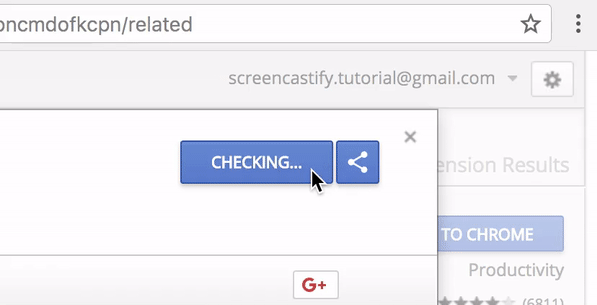
### Screencastify

**Note:** You must be using Google Chrome in order to install and use Screencastify.

1. Open Google Chrome.
2. Go to the [Chrome Web Store](https://chrome.google.com/webstore/category/extensions) and type **Screencastify** in the **Search Extensions** bar in the top left of the page. **Note:** You can also start by going to the [Screencastify website](https://www.screencastify.com/) and clicking the **Add to Chrome**button, which will take you directly to its listing in the Chrome Web Store.
3. Click the **Add to Chrome** button.
4. Click **Add extension**.
5. When the download is complete, click the gray film strip icon in the top-right corner of your browser screen to launch Screencastify.



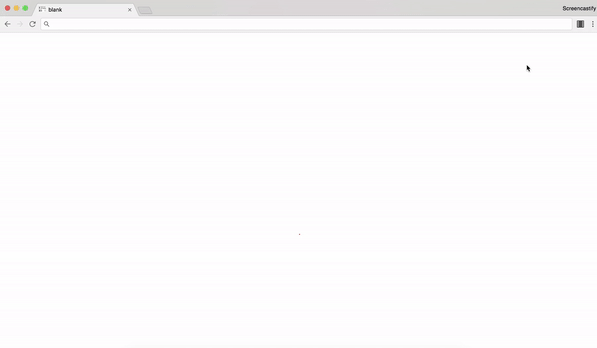




#### Allow Webcam and Microphone Access

The first time you click on the Screencastify icon, you'll be asked to give Screencastify permission to access your webcam and microphone, which enables you to narrate over your recordings as well as record your webcam. You are required to allow access in order for the Screencastify extension to function properly. You do not need to include audio or video in your recordings, and Screencastify never records audio or video unless you expressly ask it to.

1. Click the **Setup Camera Access** button.
2. Click **Allow** in the pop-up window.



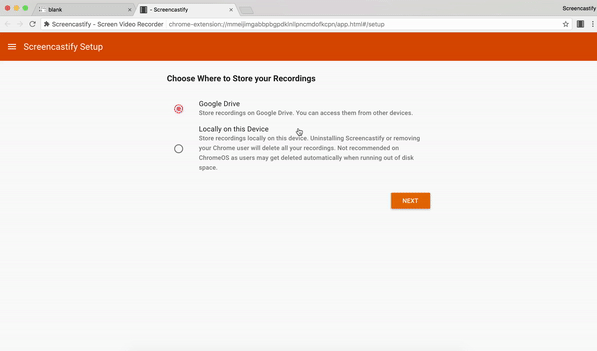
**Note:** If Screencastify is unable to detect your microphone, click [here](https://help.screencastify.com/article/206-i-dont-hear-any-audio-from-my-microphone). If it's unable to detect your webcam, click [here](https://help.screencastify.com/article/205-my-webcam-is-not-detected).

#### Connect Your Google Drive to Screencastify

It is recommended that you save all of your Screencastify recordings to Google Drive, which is safer than storing them locally on your computer. By connecting your Google Drive to Screencastify, your recordings will be saved there automatically. The length of this process depends on how long a particular recording is. **Note:** Screencastify never views, modifies, stores, or in any way interacts with the other files in your Google Drive.

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

1. Confirm that Google Drive is selected, and then click **Next**.
2. Sign in to your Google account. **Note:** You can sign in with any Google-based account, not just an @gmail.com address.
3. Allow Screencastify permission to access your Google Drive.



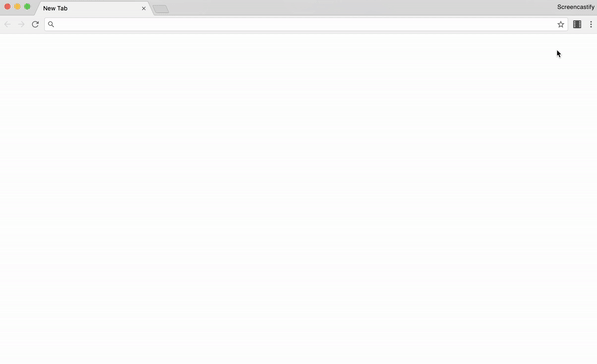
And that's all you need to do! As soon as you finish your first recording, a new folder named Screencastify will be created in your Drive. All of your recordings will be saved in this folder.

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

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

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

Want to know how to record, save to your Google Drive and share your recordings? Check it out [here](https://coding-bootcamp-fsf-prework.readthedocs-hosted.com/en/latest/modules/screencastify-record-save-to-your-google-drive-and-share-your-video/)!

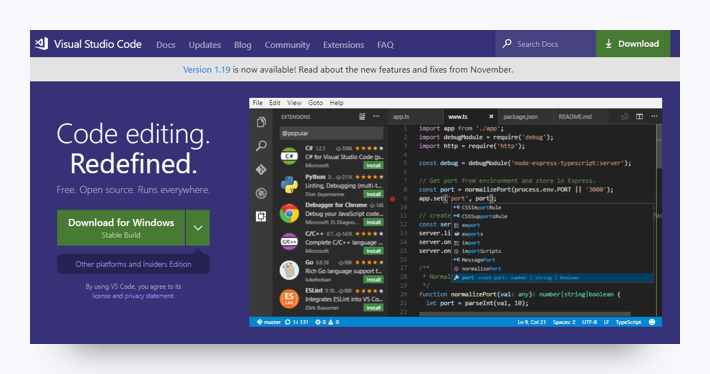


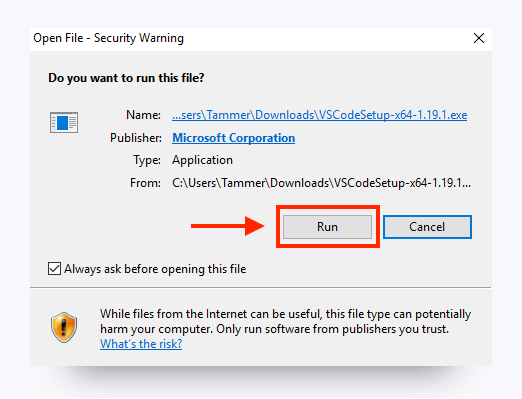
### Slack

1. Go to [Slack for Windows](https://slack.com/downloads). Click **Download**.
2. When the installation is complete, add our channel to your application.
3. Click the header of your current Slack channel.
4. Select **Sign in to another team**.
5. Enter the Slack domain provided to you during orientation.
6. Enter your email (the one that the Slack invite was sent to) and password. When you see the chatroom, you're finished.

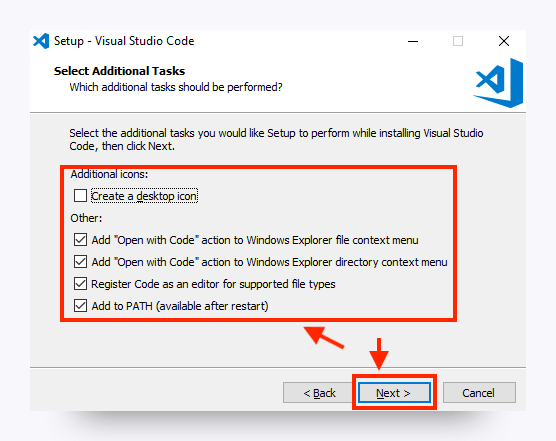
### VS Code

1. Go to the [setup page](https://code.visualstudio.com/docs/setup/setup-overview) on the VS Code website and select Windows as your platform.
2. Click **Visual Studio Code installer** in Step 1 under the Installation heading.
3. Open and run the installer file.





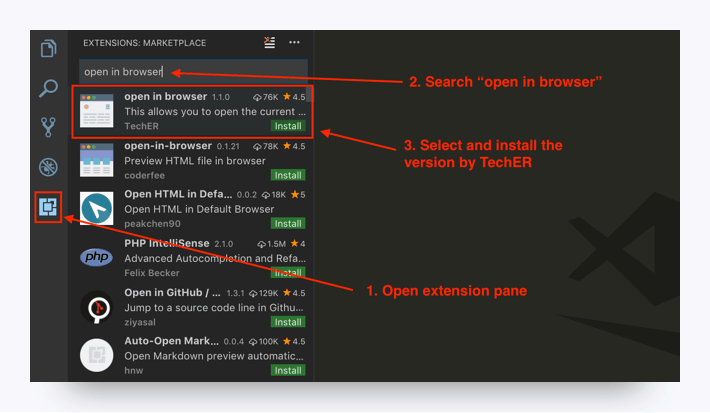
1. Follow the installer prompts. When you reach the following screen, select the following settings and click **Next**.



**Note:** When the installation is complete, you should be able to access VS Code from your Start menu.

### Open in Browser (VS Code Extension)

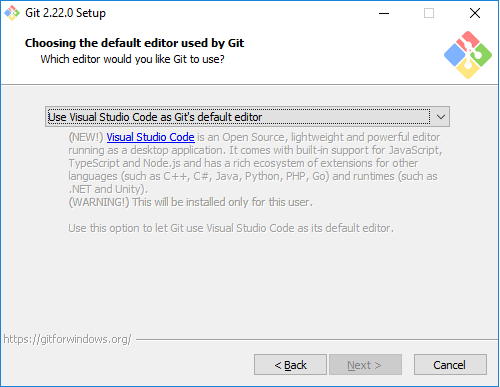
1. Open VS Code.
2. Open the extensions pane and search for **open in browser**.
3. Select the version written by TechER and click **Install**.



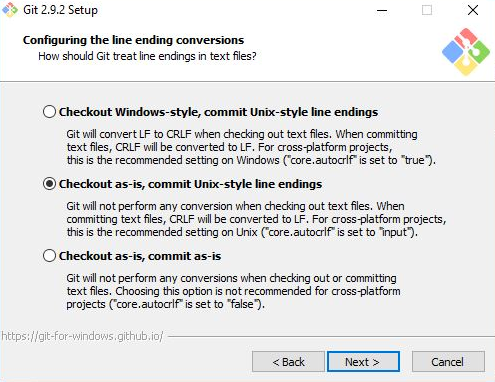
#### Git & Git Bash

**Step 1.** Go to the Git [downloads](https://git-scm.com/downloads) page. Select the download for Windows. It should automatically download the most up to date version, which is totally fine!

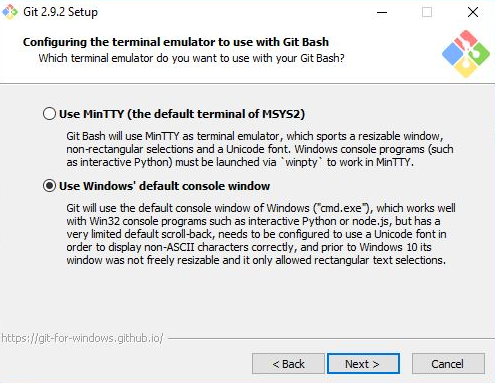
**Step 2.** Use Next to progress through the installation until you get to the screen that asks you to choose a default editor for Git. Select Use Visual Studio Code as Git's default editor.



**Step 3.** When you see a prompt like this, select Checkout as-is, commit Unix-style line endings.



**Step 4.** Finally, select Use Windows' default console window.



Any settings not mentioned here can be left in their default mode.

### Heroku Toolbelt

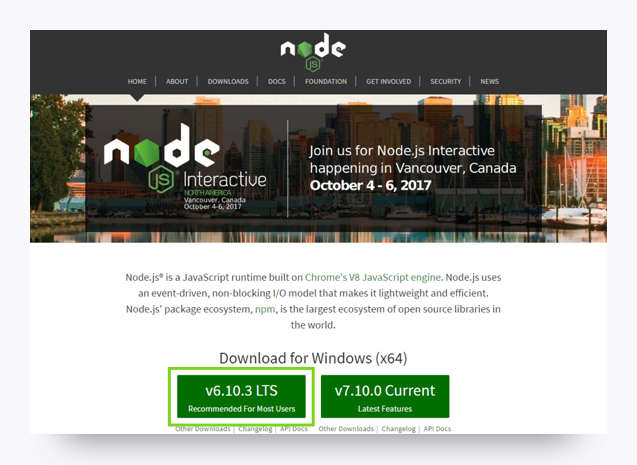
1. Sign up for a free [Heroku](https://signup.heroku.com/) account.
2. Download the [installer](https://toolbelt.heroku.com/) and read the installation guide.
3. Open Command Prompt (not Bash).

**Note:** Command Prompt, or cmd.exe, comes preloaded on Windows operating systems, but the exact location will vary depending on your version of Windows. You can locate Command Prompt using your OS's search feature.

1. Type heroku login into the command line. When prompted, press **Enter**and type in your Heroku credentials. Close Command Prompt.

### Node.js

1. Go to the [Node.js website](https://nodejs.org/en).
2. Select the Windows installer and download. Follow the prompts to complete the installation.



### MySQL Workbench

**Note:** You must install MySQL before you can set up the MySQL Workbench.

1. Go to the [MySQL Workbench](https://dev.mysql.com/downloads/workbench) website.
2. Scroll down and select the 64-bit Windows version of the software.
3. Click to download it and run through the installer.

### SSH Keys

To complete these steps, you will need to sign up for a [GitHub](https://github.com/) account if you haven't already.

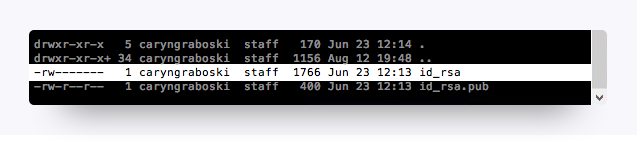
**Step 1.** Open Bash.

**Step 2.** To make sure you don’t already have a set of keys on your computer, type the following in your Bash window.

* **Note:** Copying and pasting will not work!

​ ls –al ~/.ssh

* If no keys pop up, move on to **Step 3**.
* If keys do pop up, check that none of them are listed under id\_rsa, like in this image:



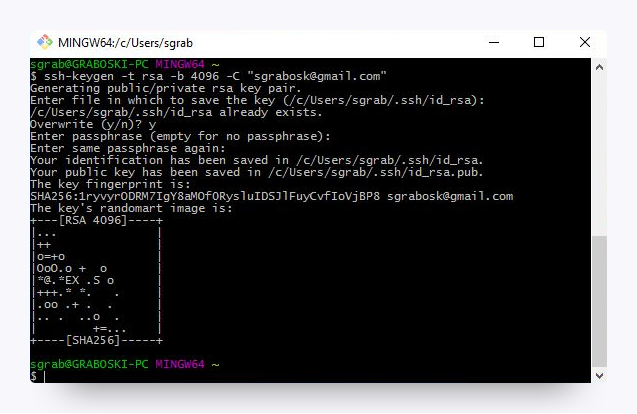
* If you find a key with a matching name, you can either overwrite it by following the next steps, or you can use the same key referenced in **Step 8**.
* If you decide not to overwrite it, you will need to remember the password tied to your key.

**Step 3.** Enter the following command along with your email to generate your keys.

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

**Step 4.** When prompted to enter a file to save the key, press **Enter**, and then enter a passphrase for your key.

* **Note:** You shouldn’t see any characters appear in the window while typing the password. When you’re finished, your window should look like this:



**Step 5.** Link your key to your machine using a tool called the ssh-agent. Run the following command in Bash to test whether the ssh-agent is running on your machine: eval "$(ssh-agent –s)". Your Bash window should look like the following:



**Step 6.** Run the following command: ssh-add ~/.ssh/id\_rsa

**Step 7.** When prompted, enter the passphrase associated with the key.

* **Note:** If you’ve forgotten this key, go back to **Step 3**.

**Step 8.** To add the key to GitHub, copy the key to your clipboard by entering the following command:

clip < ~/.ssh/id\_rsa.pub

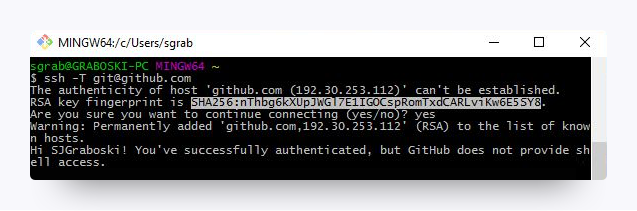
* You shouldn’t see any kind of message when you run this command. If you do, make sure you entered it correctly.
* **Note:** Do not copy anything else to your clipboard until all steps are completed. Otherwise, you’ll need to enter the copy command again.

**Step 9.** Go to GitHub's [SSH key settings](https://github.com/settings/ssh). Click **New SSH key**.

**Step 10.** 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 8.

**Step 11.** To add GitHub to your computer’s list of acceptable SSH hosts, type the following command in your Bash window: ssh –T git@github.com.

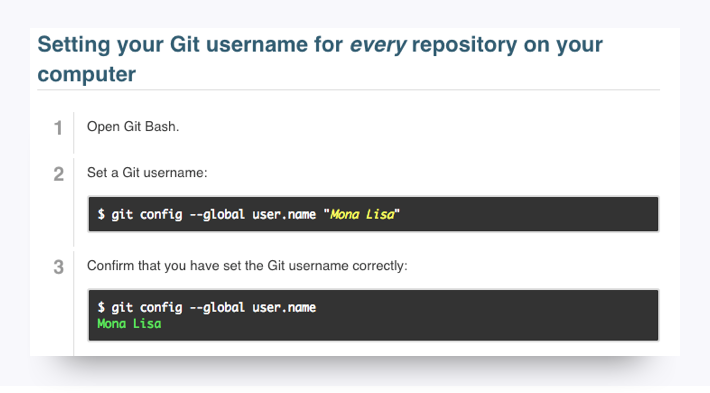
* You should see an RSA fingerprint in your window. Enter **yes** only if it matches the one highlighted in the image below:



### Setting Your Git Username for Every Repository on Your Computer

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

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

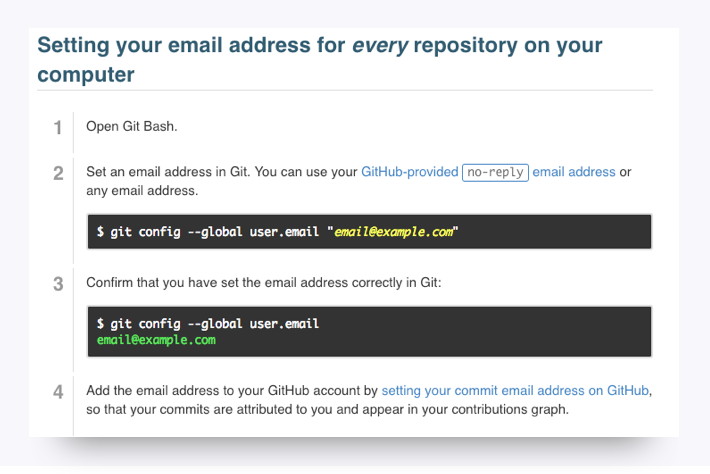


### Setting Your Email Address for Every Repository on Your Computer

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](https://help.github.com/articles/about-commit-email-addresses/)."



## You're Done!

That's all for the installations, so give yourself a pat on the back! Installations are never fun, but just like taxes, you gotta do them.

Be sure to take a break before continuing with the rest of the prework.

# Get Your Tools Installed (Windows)

You will be installing the required tools and software for the course. There's a lot to get through, so buckle in and get ready!

## Before You Begin

First, create accounts for the following services, which you'll need throughout the course. Don't just create logins; job recruiters often scour these sites in search of job candidates, so be sure to provide at least a headshot and contact information.

* [LinkedIn](https://www.linkedin.com/)

**Note:** You should create a full profile highlighting your skills and work experience, and include a headshot.

* [GitHub](https://github.com/)
* [Stack Overflow](http://stackoverflow.com/)

In addition, be sure to accept the invite for your section on [Slack](https://slack.com/). You will receive the link to your class-specific channel during orientation. If you don't receive this email, contact your Student Success Manager for assistance.

## Tool and Software Installations

Follow the instructions below to complete the installation process for all of the required tools.

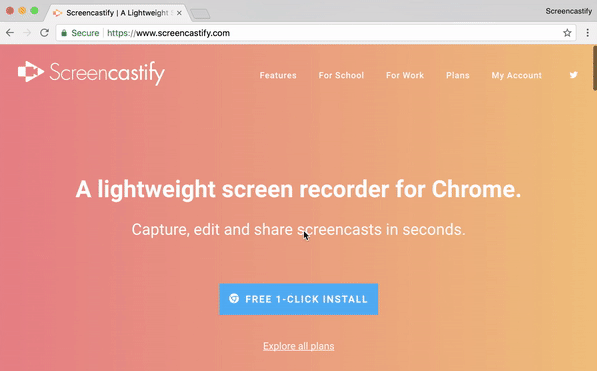
### Google Chrome

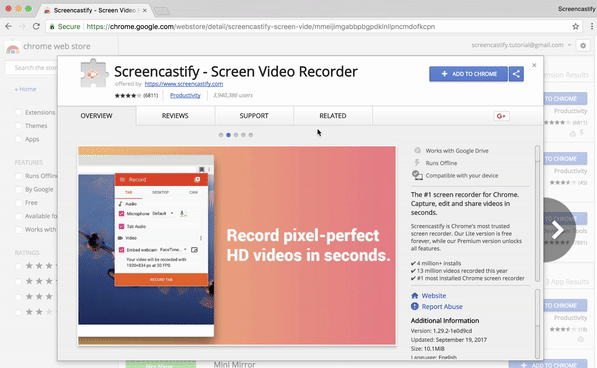
1. If you don’t already have Chrome installed, visit the download page [here](https://www.google.com/chrome/browser/desktop/index.html).
2. Download, open, and run the installation file.

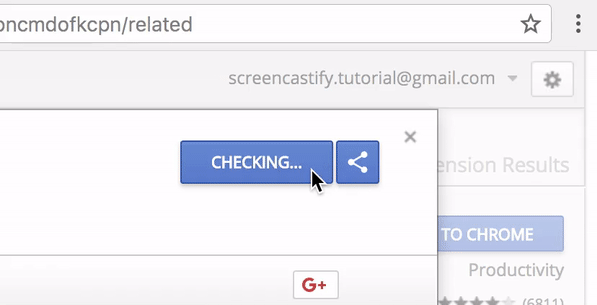
### Screencastify

**Note:** You must be using Google Chrome in order to install and use Screencastify.

1. Open Google Chrome.
2. Go to the [Chrome Web Store](https://chrome.google.com/webstore/category/extensions) and type **Screencastify** in the **Search Extensions** bar in the top left of the page. **Note:** You can also start by going to the [Screencastify website](https://www.screencastify.com/) and clicking the **Add to Chrome**button, which will take you directly to its listing in the Chrome Web Store.
3. Click the **Add to Chrome** button.
4. Click **Add extension**.
5. When the download is complete, click the gray film strip icon in the top-right corner of your browser screen to launch Screencastify.



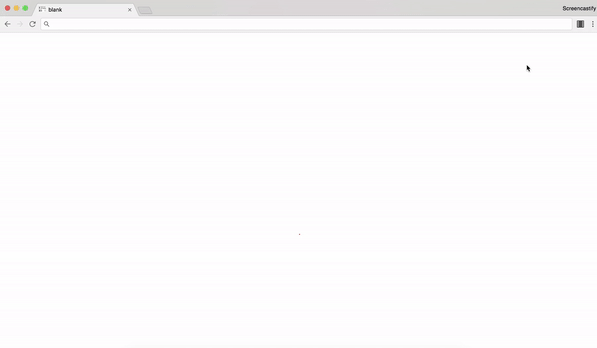




#### Allow Webcam and Microphone Access

The first time you click on the Screencastify icon, you'll be asked to give Screencastify permission to access your webcam and microphone, which enables you to narrate over your recordings as well as record your webcam. You are required to allow access in order for the Screencastify extension to function properly. You do not need to include audio or video in your recordings, and Screencastify never records audio or video unless you expressly ask it to.

1. Click the **Setup Camera Access** button.
2. Click **Allow** in the pop-up window.



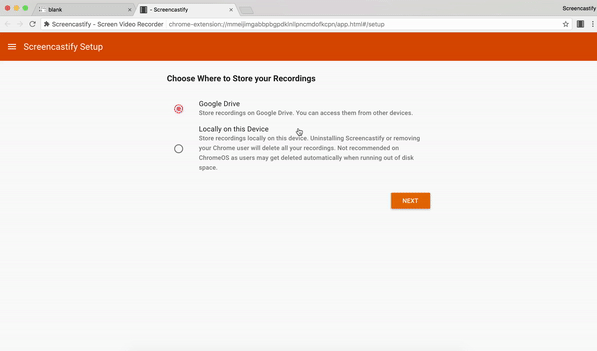
**Note:** If Screencastify is unable to detect your microphone, click [here](https://help.screencastify.com/article/206-i-dont-hear-any-audio-from-my-microphone). If it's unable to detect your webcam, click [here](https://help.screencastify.com/article/205-my-webcam-is-not-detected).

#### Connect Your Google Drive to Screencastify

It is recommended that you save all of your Screencastify recordings to Google Drive, which is safer than storing them locally on your computer. By connecting your Google Drive to Screencastify, your recordings will be saved there automatically. The length of this process depends on how long a particular recording is. **Note:** Screencastify never views, modifies, stores, or in any way interacts with the other files in your Google Drive.

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

1. Confirm that Google Drive is selected, and then click **Next**.
2. Sign in to your Google account. **Note:** You can sign in with any Google-based account, not just an @gmail.com address.
3. Allow Screencastify permission to access your Google Drive.



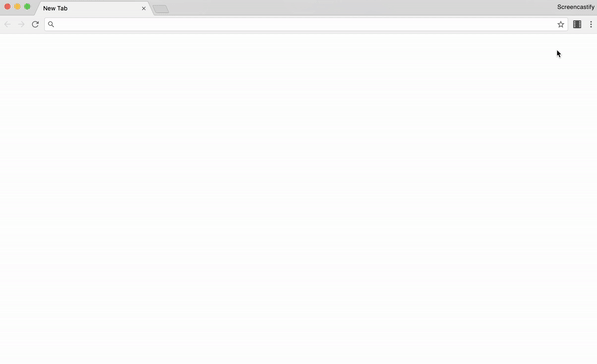
And that's all you need to do! As soon as you finish your first recording, a new folder named Screencastify will be created in your Drive. All of your recordings will be saved in this folder.

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

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

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

Want to know how to record, save to your Google Drive and share your recordings? Check it out [here](https://coding-bootcamp-fsf-prework.readthedocs-hosted.com/en/latest/modules/screencastify-record-save-to-your-google-drive-and-share-your-video/)!

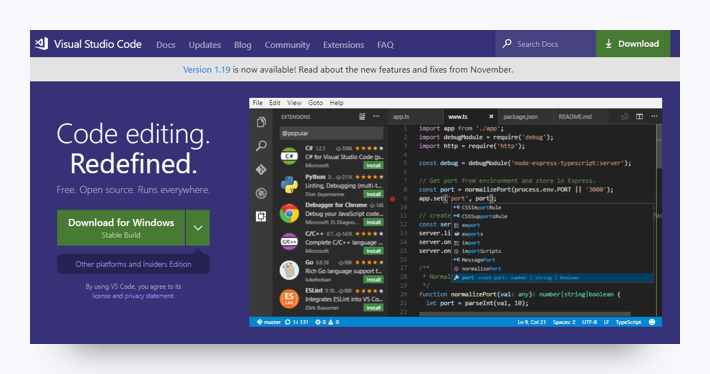


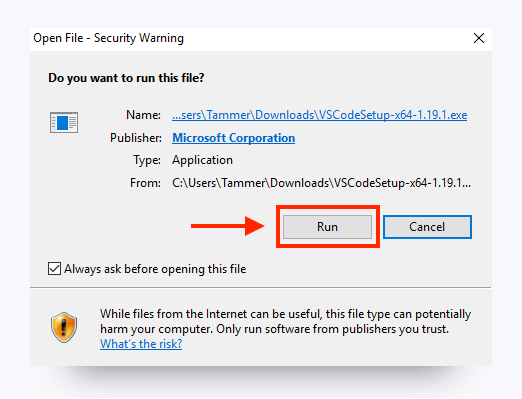
### Slack

1. Go to [Slack for Windows](https://slack.com/downloads). Click **Download**.
2. When the installation is complete, add our channel to your application.
3. Click the header of your current Slack channel.
4. Select **Sign in to another team**.
5. Enter the Slack domain provided to you during orientation.
6. Enter your email (the one that the Slack invite was sent to) and password. When you see the chatroom, you're finished.

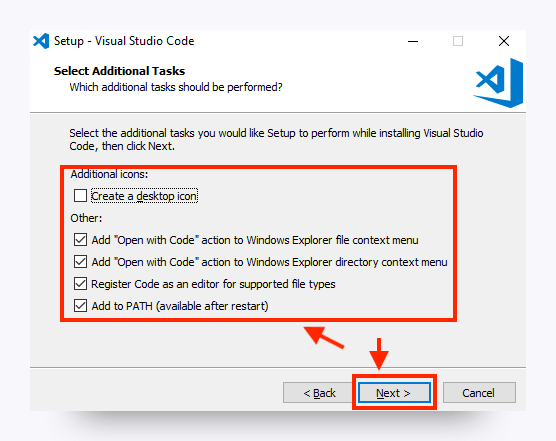
### VS Code

1. Go to the [setup page](https://code.visualstudio.com/docs/setup/setup-overview) on the VS Code website and select Windows as your platform.
2. Click **Visual Studio Code installer** in Step 1 under the Installation heading.
3. Open and run the installer file.





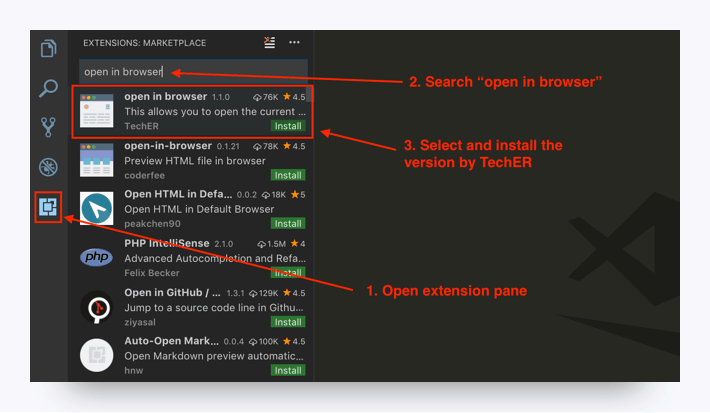
1. Follow the installer prompts. When you reach the following screen, select the following settings and click **Next**.



**Note:** When the installation is complete, you should be able to access VS Code from your Start menu.

### Open in Browser (VS Code Extension)

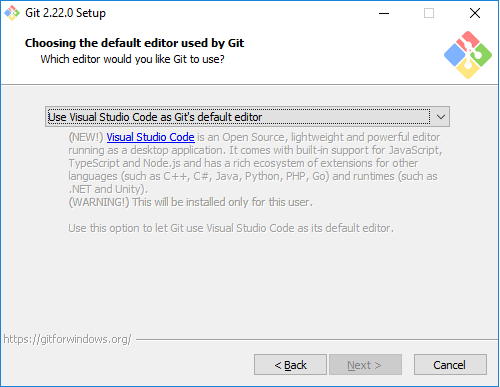
1. Open VS Code.
2. Open the extensions pane and search for **open in browser**.
3. Select the version written by TechER and click **Install**.



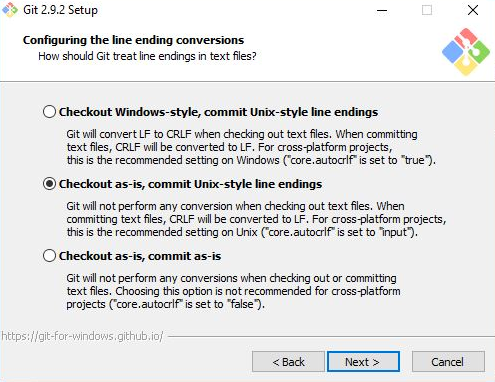
#### Git & Git Bash

**Step 1.** Go to the Git [downloads](https://git-scm.com/downloads) page. Select the download for Windows. It should automatically download the most up to date version, which is totally fine!

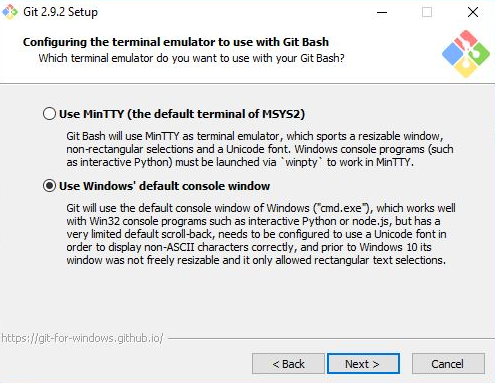
**Step 2.** Use Next to progress through the installation until you get to the screen that asks you to choose a default editor for Git. Select Use Visual Studio Code as Git's default editor.



**Step 3.** When you see a prompt like this, select Checkout as-is, commit Unix-style line endings.



**Step 4.** Finally, select Use Windows' default console window.



Any settings not mentioned here can be left in their default mode.

### Heroku Toolbelt

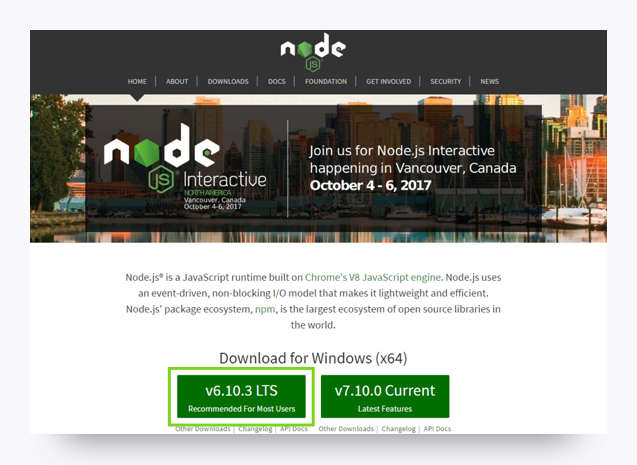
1. Sign up for a free [Heroku](https://signup.heroku.com/) account.
2. Download the [installer](https://toolbelt.heroku.com/) and read the installation guide.
3. Open Command Prompt (not Bash).

**Note:** Command Prompt, or cmd.exe, comes preloaded on Windows operating systems, but the exact location will vary depending on your version of Windows. You can locate Command Prompt using your OS's search feature.

1. Type heroku login into the command line. When prompted, press **Enter**and type in your Heroku credentials. Close Command Prompt.

### Node.js

1. Go to the [Node.js website](https://nodejs.org/en).
2. Select the Windows installer and download. Follow the prompts to complete the installation.



### MySQL Workbench

**Note:** You must install MySQL before you can set up the MySQL Workbench.

1. Go to the [MySQL Workbench](https://dev.mysql.com/downloads/workbench) website.
2. Scroll down and select the 64-bit Windows version of the software.
3. Click to download it and run through the installer.

### SSH Keys

To complete these steps, you will need to sign up for a [GitHub](https://github.com/) account if you haven't already.

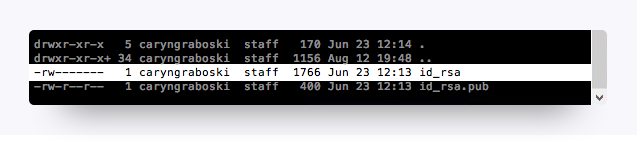
**Step 1.** Open Bash.

**Step 2.** To make sure you don’t already have a set of keys on your computer, type the following in your Bash window.

* **Note:** Copying and pasting will not work!

​ ls –al ~/.ssh

* If no keys pop up, move on to **Step 3**.
* If keys do pop up, check that none of them are listed under id\_rsa, like in this image:



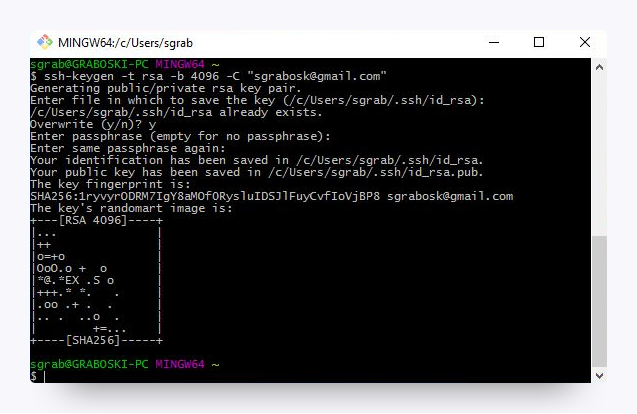
* If you find a key with a matching name, you can either overwrite it by following the next steps, or you can use the same key referenced in **Step 8**.
* If you decide not to overwrite it, you will need to remember the password tied to your key.

**Step 3.** Enter the following command along with your email to generate your keys.

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

**Step 4.** When prompted to enter a file to save the key, press **Enter**, and then enter a passphrase for your key.

* **Note:** You shouldn’t see any characters appear in the window while typing the password. When you’re finished, your window should look like this:



**Step 5.** Link your key to your machine using a tool called the ssh-agent. Run the following command in Bash to test whether the ssh-agent is running on your machine: eval "$(ssh-agent –s)". Your Bash window should look like the following:



**Step 6.** Run the following command: ssh-add ~/.ssh/id\_rsa

**Step 7.** When prompted, enter the passphrase associated with the key.

* **Note:** If you’ve forgotten this key, go back to **Step 3**.

**Step 8.** To add the key to GitHub, copy the key to your clipboard by entering the following command:

clip < ~/.ssh/id\_rsa.pub

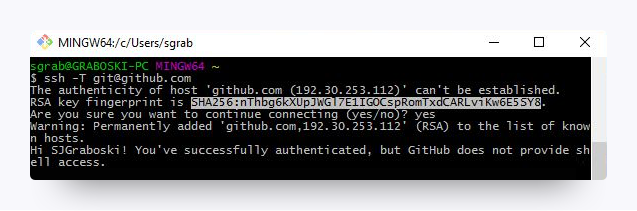
* You shouldn’t see any kind of message when you run this command. If you do, make sure you entered it correctly.
* **Note:** Do not copy anything else to your clipboard until all steps are completed. Otherwise, you’ll need to enter the copy command again.

**Step 9.** Go to GitHub's [SSH key settings](https://github.com/settings/ssh). Click **New SSH key**.

**Step 10.** 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 8.

**Step 11.** To add GitHub to your computer’s list of acceptable SSH hosts, type the following command in your Bash window: ssh –T git@github.com.

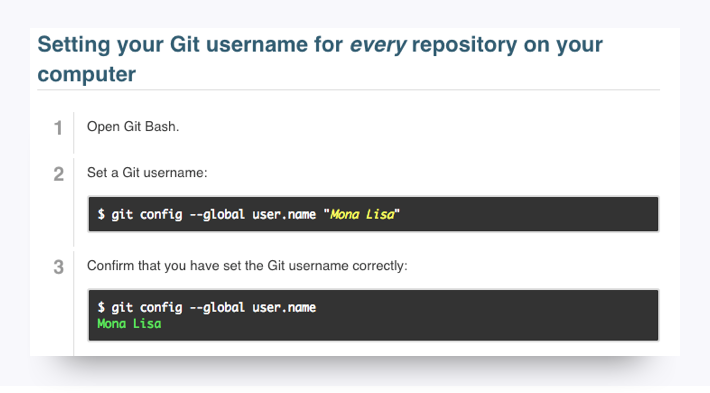
* You should see an RSA fingerprint in your window. Enter **yes** only if it matches the one highlighted in the image below:



### Setting Your Git Username for Every Repository on Your Computer

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

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

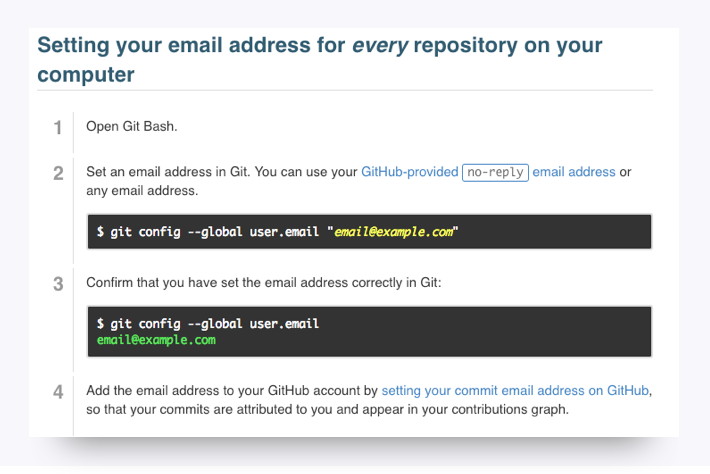


### Setting Your Email Address for Every Repository on Your Computer

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](https://help.github.com/articles/about-commit-email-addresses/)."



## You're Done!

That's all for the installations, so give yourself a pat on the back! Installations are never fun, but just like taxes, you gotta do them.

Be sure to take a break before continuing with the rest of the prework.