INFO/CS 2300: Lab 1

By Sharon Jeong

Attendance: **Due at the beginning of lab section.**
Participation: **Due Tuesday, 2/6 at 5:00pm**.

Overview

In this lab, we will walk you through how to install all the software required for this course as well as gain practice on how to submit an assignment through Git. Remember credit is dependent on **both** your presence in your **enrolled lab** (in student center) as well as submission of material. We expect that not everyone will be able to finish this lab in class due to differences in machinery and operating systems. You may need to finish this outside of class as homework.

Learning Objectives

- Become familiar with web development tools like Atom and PHP.
- Exposure to the basics of using a version control tool like Git.
- Learn how to submit assignments in this course.

Credit

To receive credit for this lab you must:

1. Attend your registered lab section.
    - For this lab, you will sign the attendance sheet after showing a TA your registered lab section in Student Center.
    - If you're on the wait-list, just let the TA know. You can still sign the sheet.
2. Fully participate in the lab activity.
    - Your participation is based on whether you complete this entire document. You will receive participation credit if you submit your lab to Git by **Tuesday, 2/6 at 5:00pm**.

What You Need

If you plan on using your personal laptop to work on your assignments this semester, you should bring it. Otherwise you'll configure the lab computer.

Tips

1. **Avoid skimming this document. Read everything carefully.**
2. If you get stuck, **go back and check that you completed the instructions before asking for help**.
3. If you see that your neighbors have successfully completed a step, **feel free to ask them for help**. We're a team

and we'll all do better in this class if we support one another.

A Note About Speed

This lab is long and complicated. Don't let this scare you. The teaching staff are here to help.

We understand that people work at different paces and some of you may find yourself ahead of everyone else in the class. We ask that if you are ahead and get stuck that you wait until the TAs have finished helping all the students who are behind or on the current step before stopping them for help.

If you feel like you are taking longer than everyone else, don't sweat it. The TAs will hold office hours to help you through this lab if you don't finish it during class.

Part I

Introductions

You'll be in this lab section with the same people all semester long. Take a brief moment to introduce yourself to those around you. Get to know one another so you can help each other out. You might want to even exchange contact information.

Check Your Section Number

In past semesters, students have made the mistake of attending the wrong lab section. Please go to Student Center and open your enrolled courses. Show this page to the TA to sign the attendance sheet; a TA must see and confirm that you are currently enrolled in the appropriate section.

Note: Other platforms such as CoursePad and Scheduler are not synced with the university and you should not be looking at these sites to check enrollment.

In the case you are not in the right section, please go to the correct lab section now.

If you are wait-listed, we cannot accommodate all the students due to the constrained sizes of lab classrooms and limited teaching staff. You can stay put for today. However, we ask that you leave the seats for the registered students today.

Part II

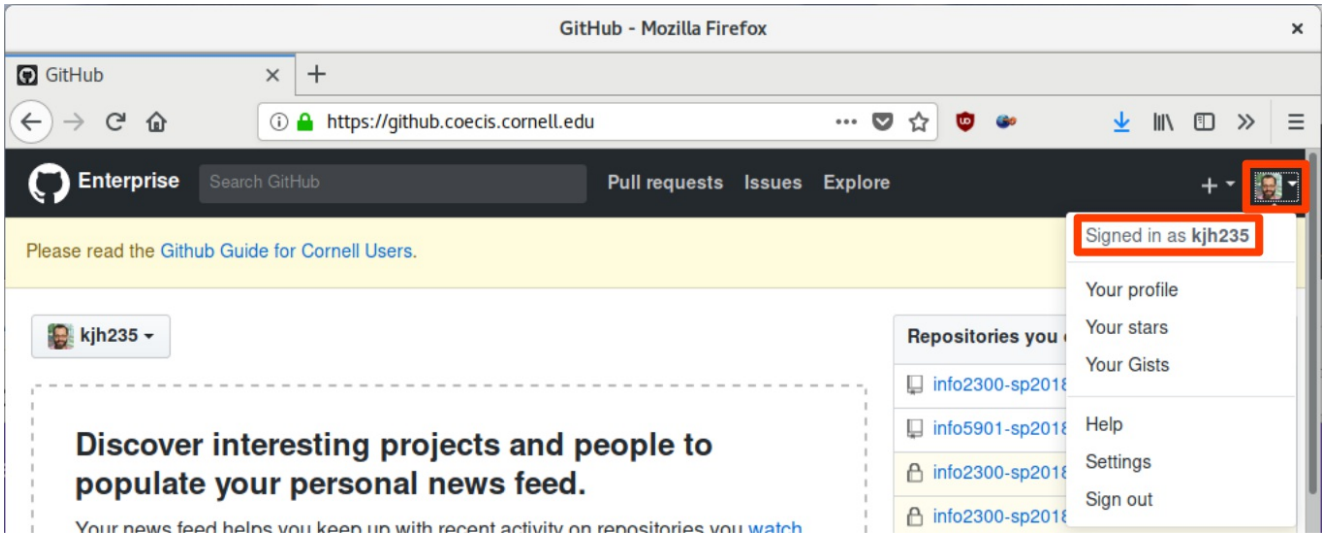
In this part, you'll install the official course software you'll need to use for assignments, labs, and projects in this course. All of the software that is listed in this document is supported by all of the teaching staff. **You are free to use any other tools that you like, however if you have an issue, we will only help you with the officially support course tools.**

See the syllabus for the list of officially supported tools. Please note that Sublime, Safari, and MAMP are **not** supported.

1. Request a Git Repository

In order to submit this lab you'll need a Git repository. We need to do this for you, but in order for us to create your repository, we need some information from you.

1. Go to https://github.coecis.cornell.edu/ and log in using your Cornell credentials (your NetID as the username and your password).
2. Click the avatar in the top-right corner. You should see **Signed in as *USERNAME***.



3. Fill out this survey: https://goo.gl/forms/DZdNTHT9lIHr4HY52

2. Install or Update Git

Git is a free and open source distributed version control system. In industry, it is the main tool for collaborative work and a good skill to learn. We will provide some experience working with it by using it to distribute and submit assignments throughout the semester.

To install or update Git, please download the install package at the following link: https://git-scm.com/.

1. On the bottom right corner, there will be an image of a computer screen with a link to download (the website will have identified your operating system (Mac OS X, Windows, Linux, etc.)).

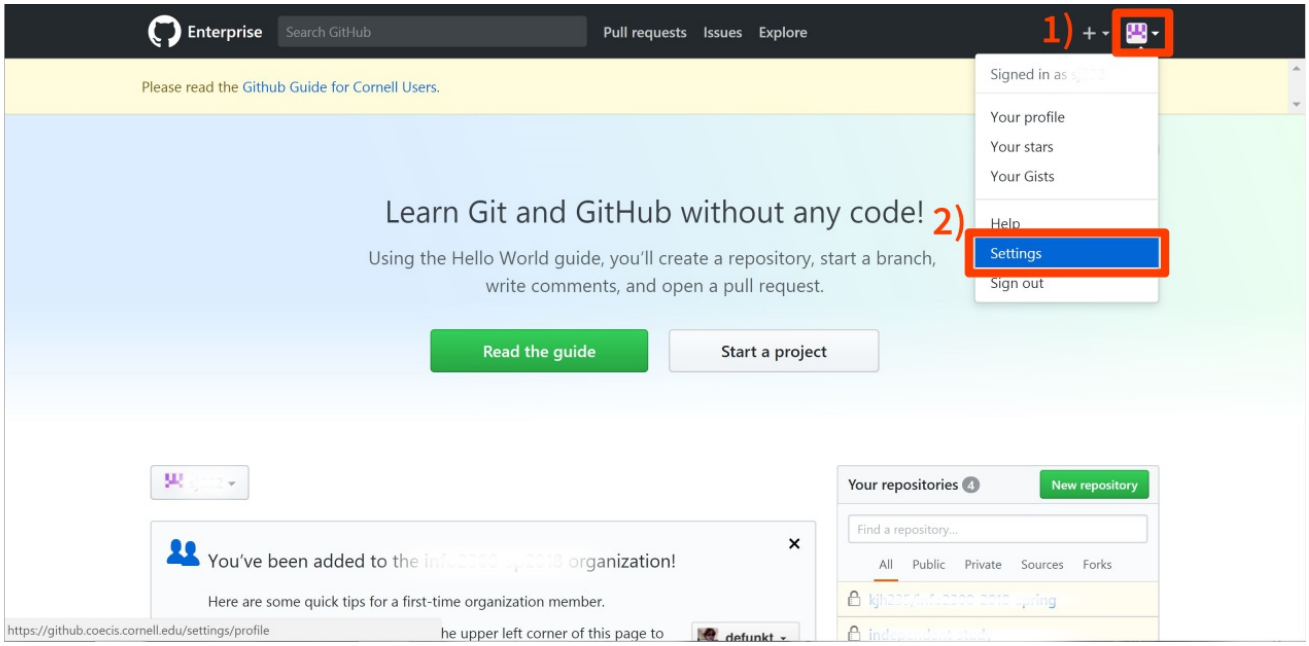
2. We will be using the default settings; just continuously click next until you reach **Finish**.

4. Adding a new SSH Key to your **Cornell** GitHub

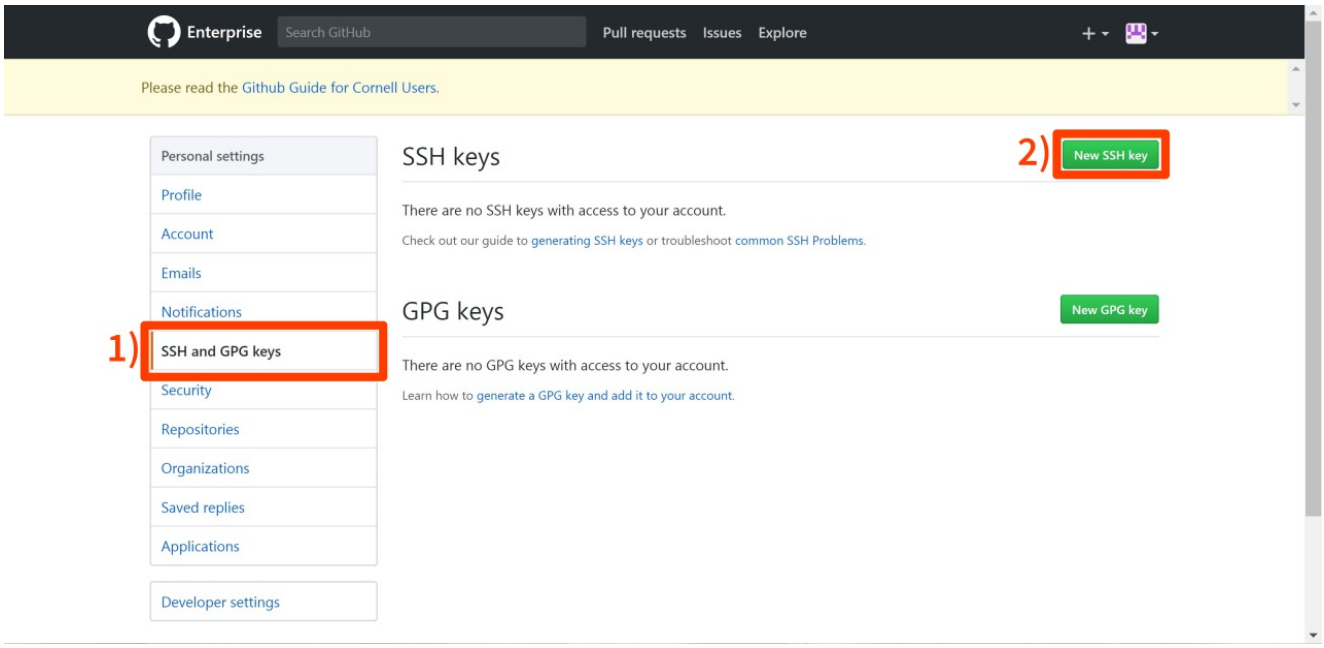
Course documents and assignments will be distributed and submitted through the Cornell GitHub. Adding a SSH Key to your GitHub is useful as it will authenticate your computer so that you no longer have to type your password every time you interact with the repository.

1. Go to https://github.coecis.cornell.edu/ and log in using your Cornell credentials (your NetID as the username and your password).

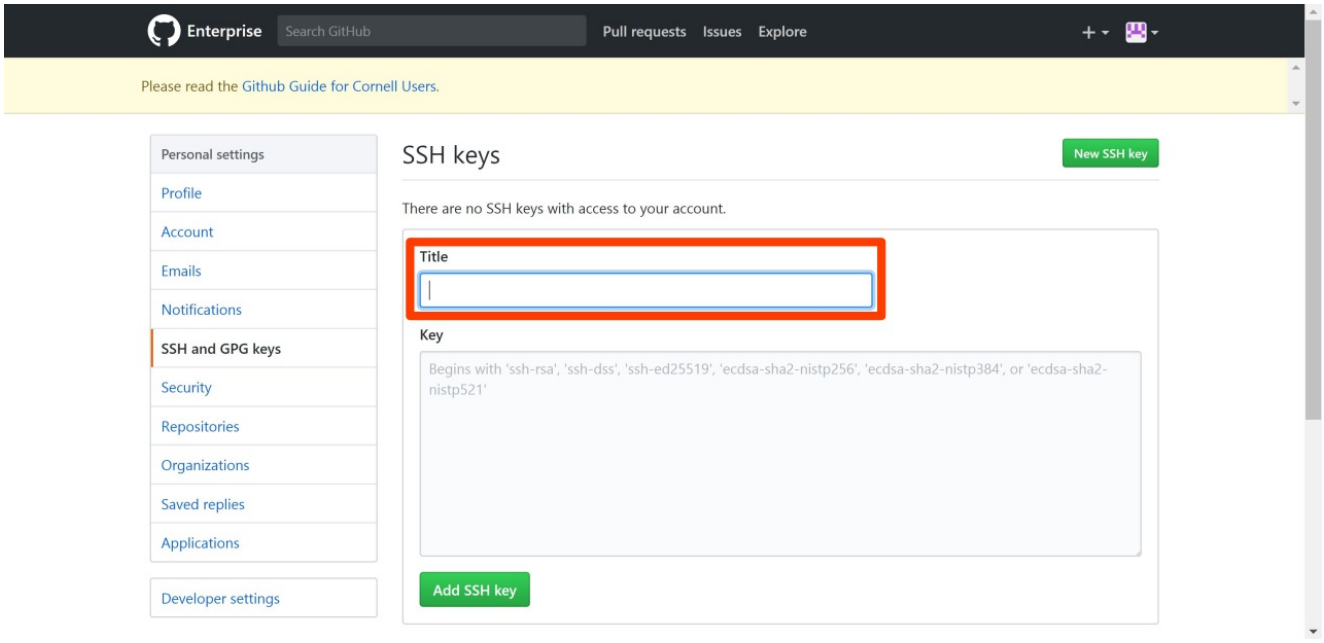
2. Go to the upper right corner of the screen to click your avatar. From the drop down, select "Settings".



3. On the left-hand panel, under "Personal settings", click "SSH and GPG keys" and then click "New SSH key" in the upper right corner.



4. Create whatever name you want to identify your repository such as "My Laptop" in "Title".



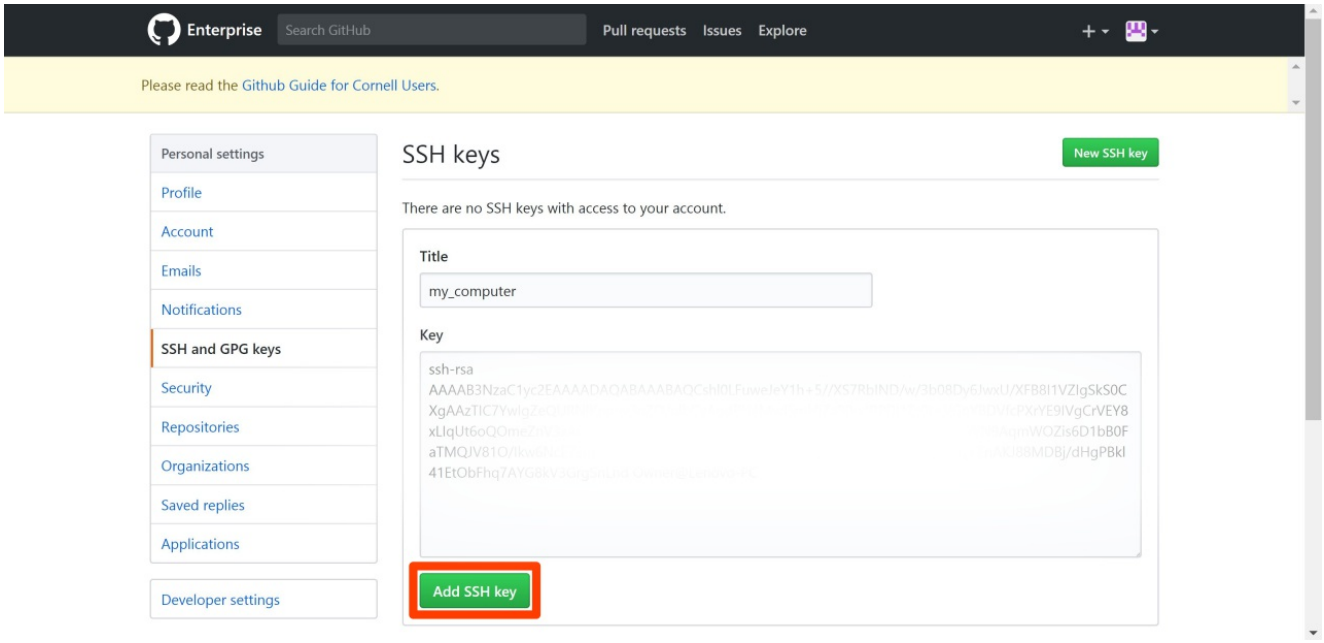
5. Minimize your web browser. **Do not close the GitHub SSH key page.
6. Now stop and open your command line
    - Windows: open Git Bash (**NOT** Command Prompt).
    - Mac: open Terminal.
7. Copy and paste into the command line:
    - Windows: `clip < ~/.ssh/id_rsa.pub`
    - Mac: `pbcopy < ~/.ssh/id_rsa.pub`

Once the above lines of code run **without** an error, **DO NOT** copy or cmd-c/ctrl-c anything as this will override the unique SSH key we just copied to the clipboard.

Note: Some of you may receive an error message that says something like `bash: /Users/USERNAME/.ssh/id_rsa.pub: No such file or directory`.

1. Enter this command at the command line: `ssh-keygen`
2. Press "Enter" or "Return" key three times.
3. Redo step 6. You should no longer have the *no such file error*.

8. Go pull up the browser again and paste into "Key" and then click "Add SSH key"



9. Close your command prompt, but do not log out of GitHub or close your browser window; we will need to access your GitHub again in a future step.

4. Install or Update Firefox and Chrome

Google Chrome and Mozilla Firefox are the two most popular browsers used. For example, look at W3Schools Statistics, together, 90% of their visitors utilize Chrome and Firefox. Since they currently dominate the browser usage share, it is important that we test our projects and designs on these two browsers throughout the semester.

This course uses Mozilla's Firefox as the official browser; however, **grading will be done in either or both Firefox and Chrome**. Your web pages will need to work in both.

Please go through the steps to install and update (if already installed) for both browsers on your system.

Firefox

Firefox is the official browser for this course.

Update Firefox

If you already have Firefox installed, update to the latest version.

1. Go to Menu (the hamburger button/triple bar) on the top right corner.
2. Click "Help" and then "About Firefox."
3. If Firefox is out of date, it will start downloading the latest update. Please the restart button when prompted.

Install Firefox

Follow the instructions at https://www.mozilla.org/en-US/firefox/

Chrome

You may install Google Chrome or the open source version, Chromium.

Update Chrome

1. Go to Menu (the vertical ellipsis ⋮) on the top right corner.
2. Click "Help" and then "About Google Chrome."
3. Google Chrome will automatically start updating to the newest version.

Install Chrome

Follow the instructions at https://www.google.com/chrome/

5. Install DB Browser for SQLite

We will be using DB Browser for SQLite for accessing our development SQL databases in this course. Please install the appropriate version for your operating system from http://sqlitebrowser.org/

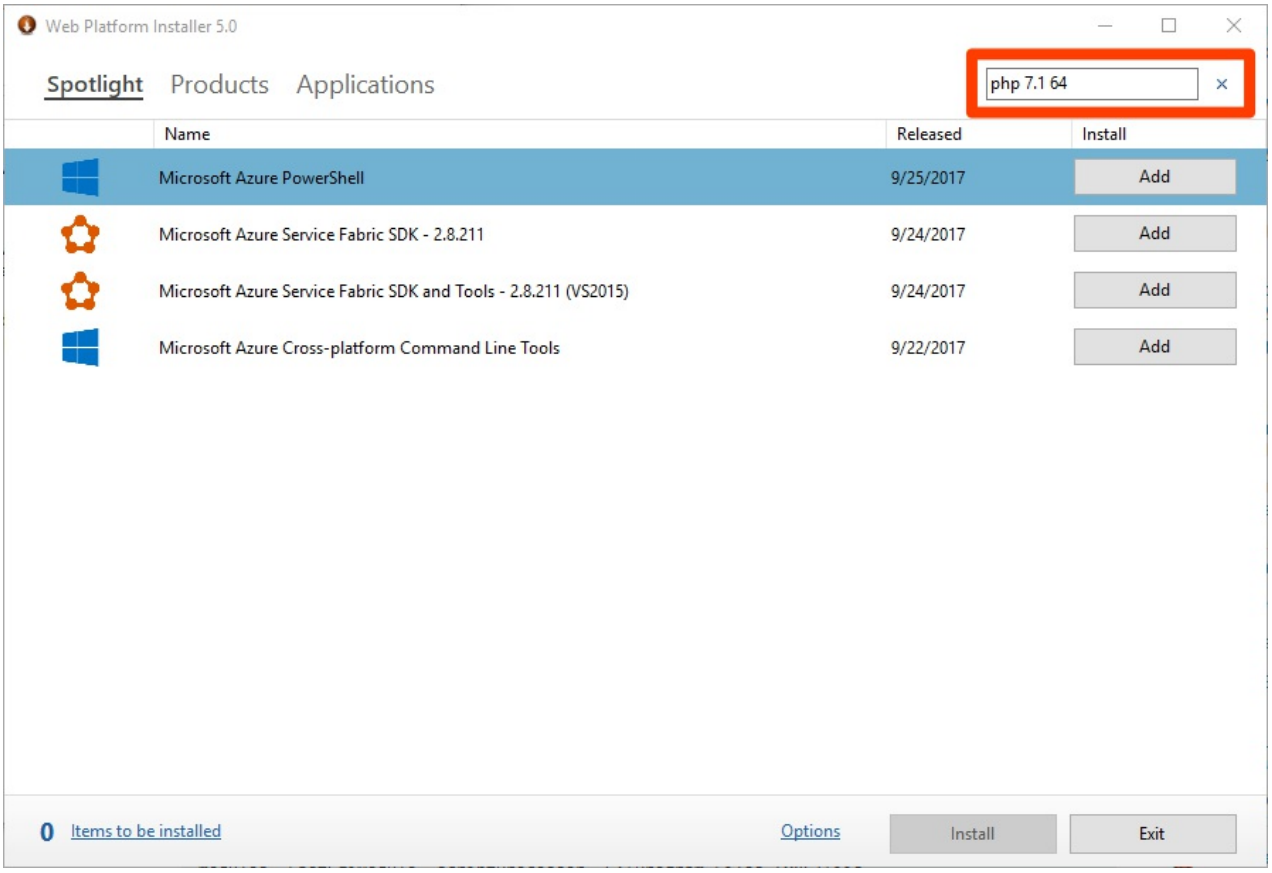
6. Install or Update to PHP 7.1 or 7.2

PHP enables us to create *dynamically generated* web pages. PHP stands for **P**HP: **H**ypertext **P**reprocessor (it's a recursive acronym).

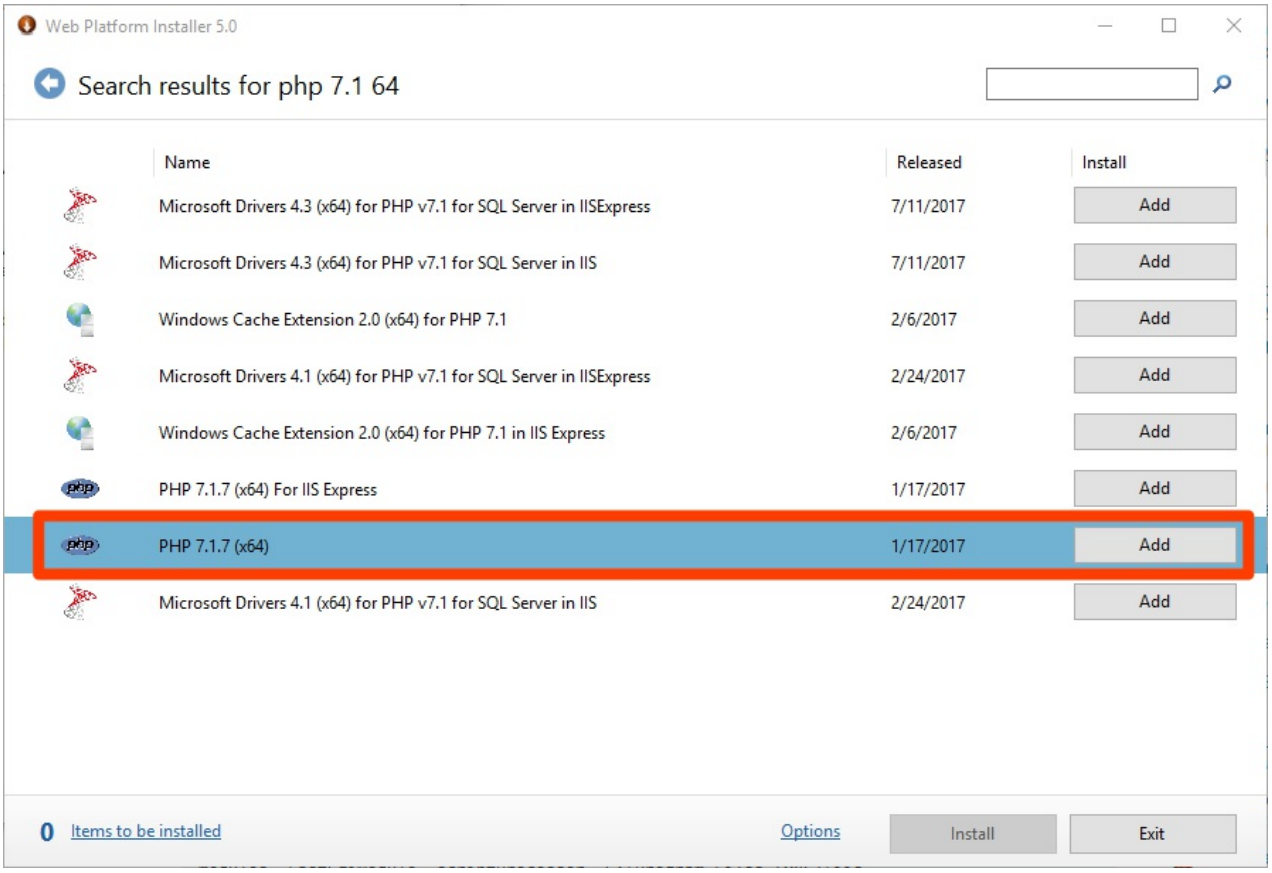
How you install PHP depends on your operating system. Read the section for your system and follow the instructions.

Windows

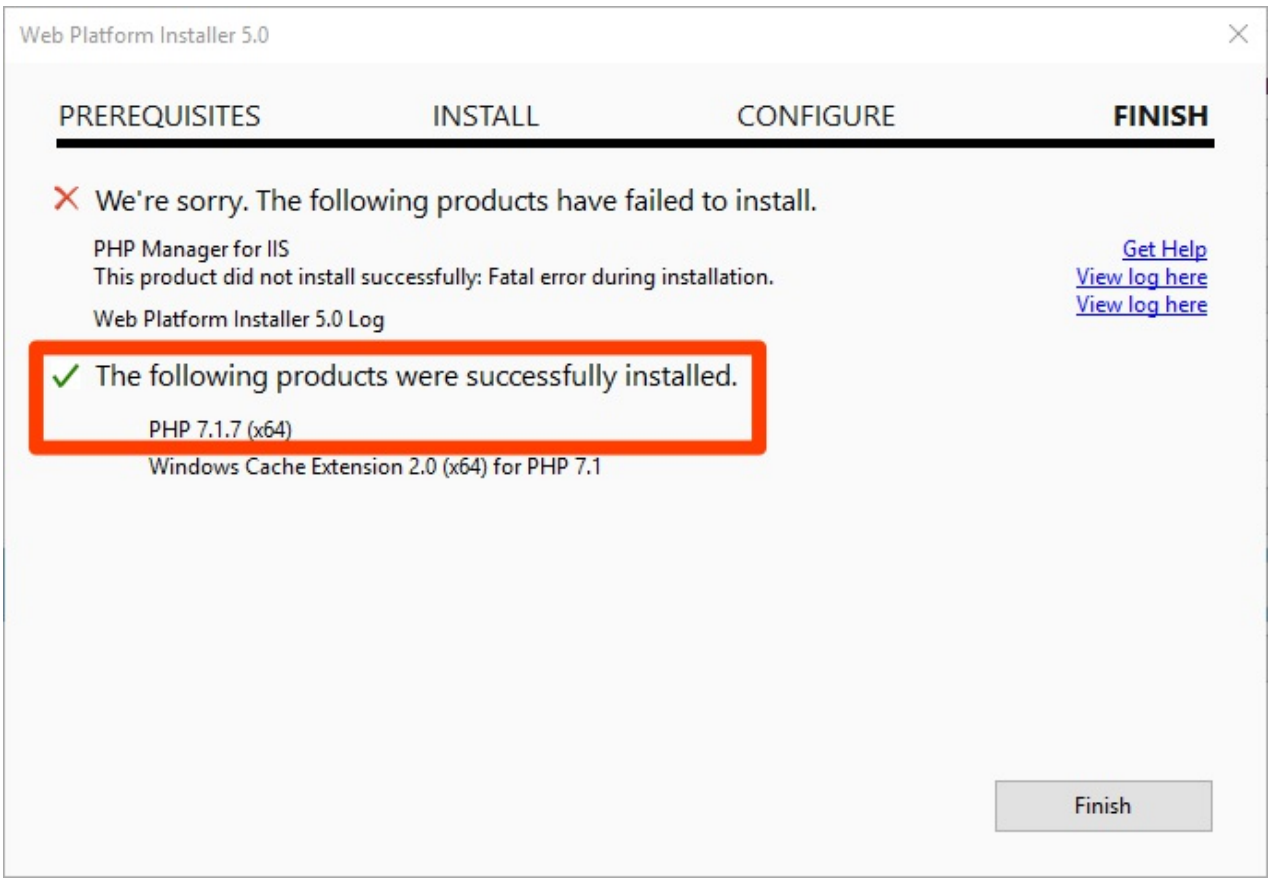
1. Go to Microsoft's Web Platform Install page at https://www.microsoft.com/web/downloads/platform.aspx
2. Click the green "Free Download" button on the right.
3. Once the program has finished downloading, launch it.
4. In the top right, search for "**php 7.1 64**":



5. Click the “Add” button next to **PHP 7.1.7 (x64)**.



6. Scroll to the bottom and click “Install.”
7. Click "I accept" and wait for installation to complete.
8. Some platforms (as pictured below) may fail to install; this is okay, as long as **PHP is installed successfully**.



Mac

1. Open Terminal (Applications/Utilities/Terminal)
2. Copy and paste `curl -s https://php-osx.liip.ch/install.sh | bash -s 7.1` into the command line and press "enter" or "return".

Linux

Install the the php-pdo package.

7. Install or update Atom

We will use Atom as the official code editor of INFO 2300. A lot of code editors, specifically integrated development environments (IDEs), are very complex to learn. They also make learning programming more difficult. We will use in Atom in the class because it's a classic code editor and you should have exposure to one before you graduate. Atom also integrates with GitHub so it will make the submission process simple; no more panic due to CMS going down!

Updating Atom

If you already have Atom installed, you should update it to the latest version.

Windows

1. Click "Help" on the top menu bar and then click "About Atom."
2. Update so that you see "Atom is up to date!"

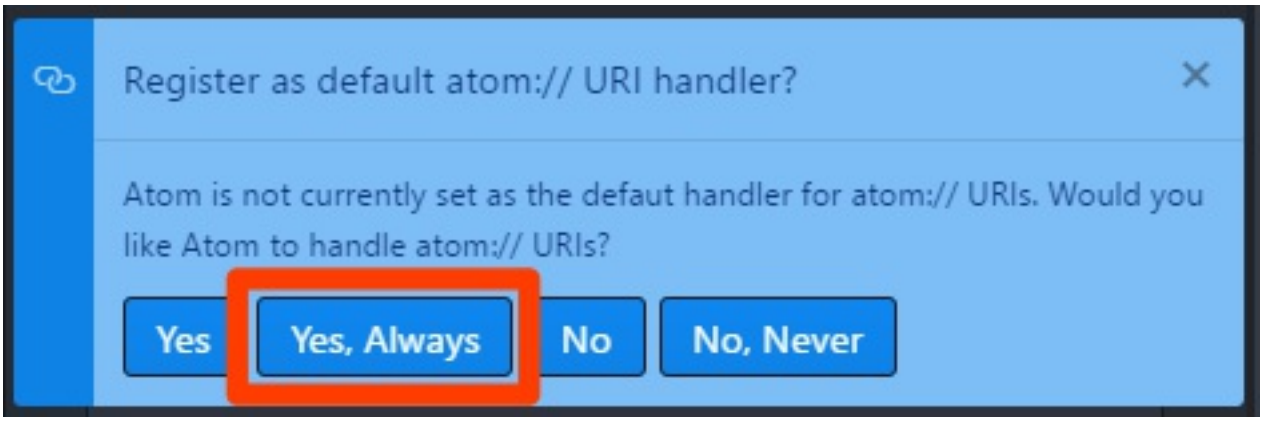
Mac

1. Click "Atom" in the top left menu bar and then click "About Atom."
2. Update so that you see "Atom is up to date!"

Installing Atom

If you don't have Atom installed, you'll need to install it.

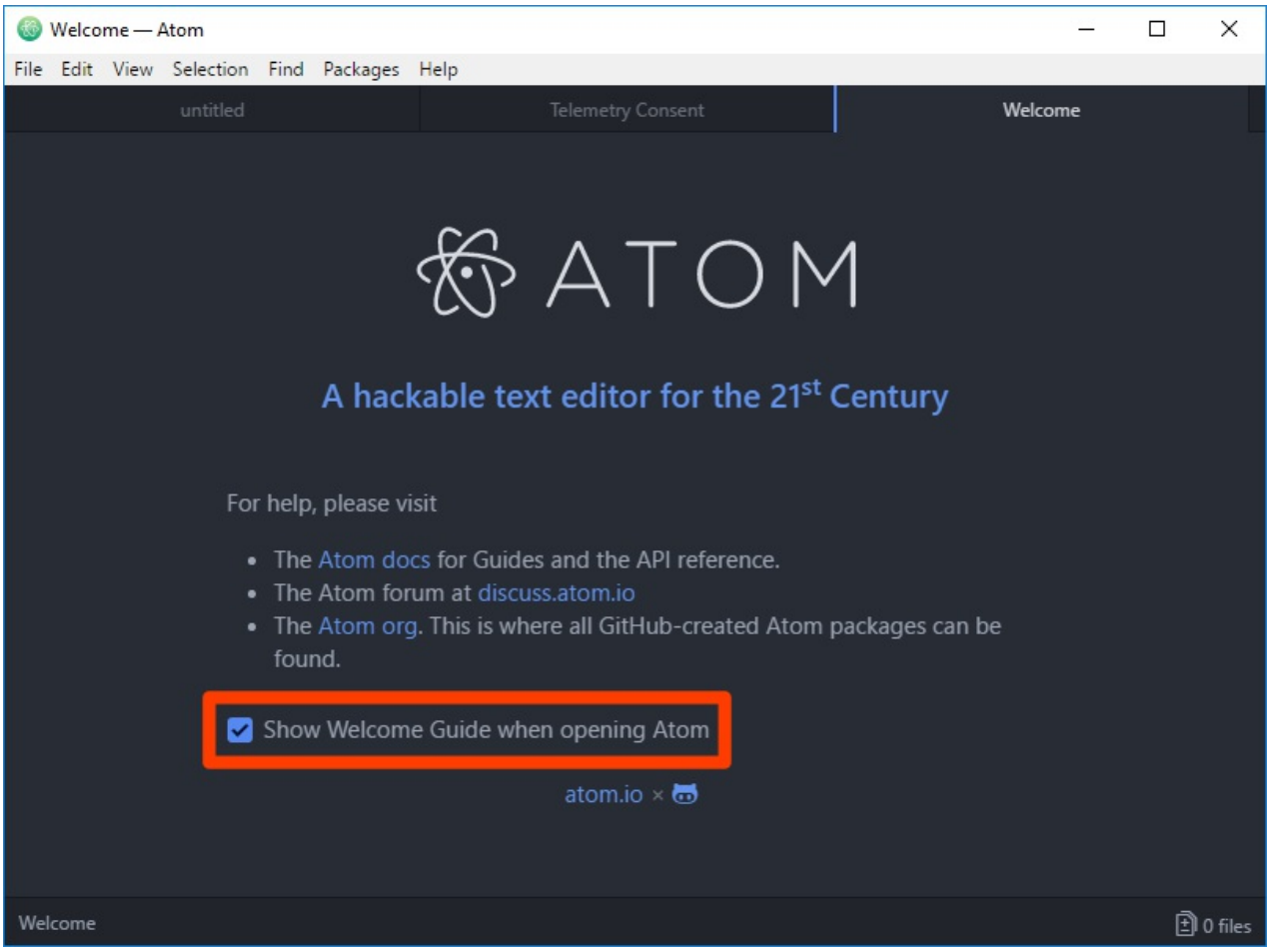
1. Download the install Atom from the following website, https://atom.io/
2. When you finish the installation, and open the application.
3. If Atom asks you to register a URI handler, we recommend you select "Yes, Always".



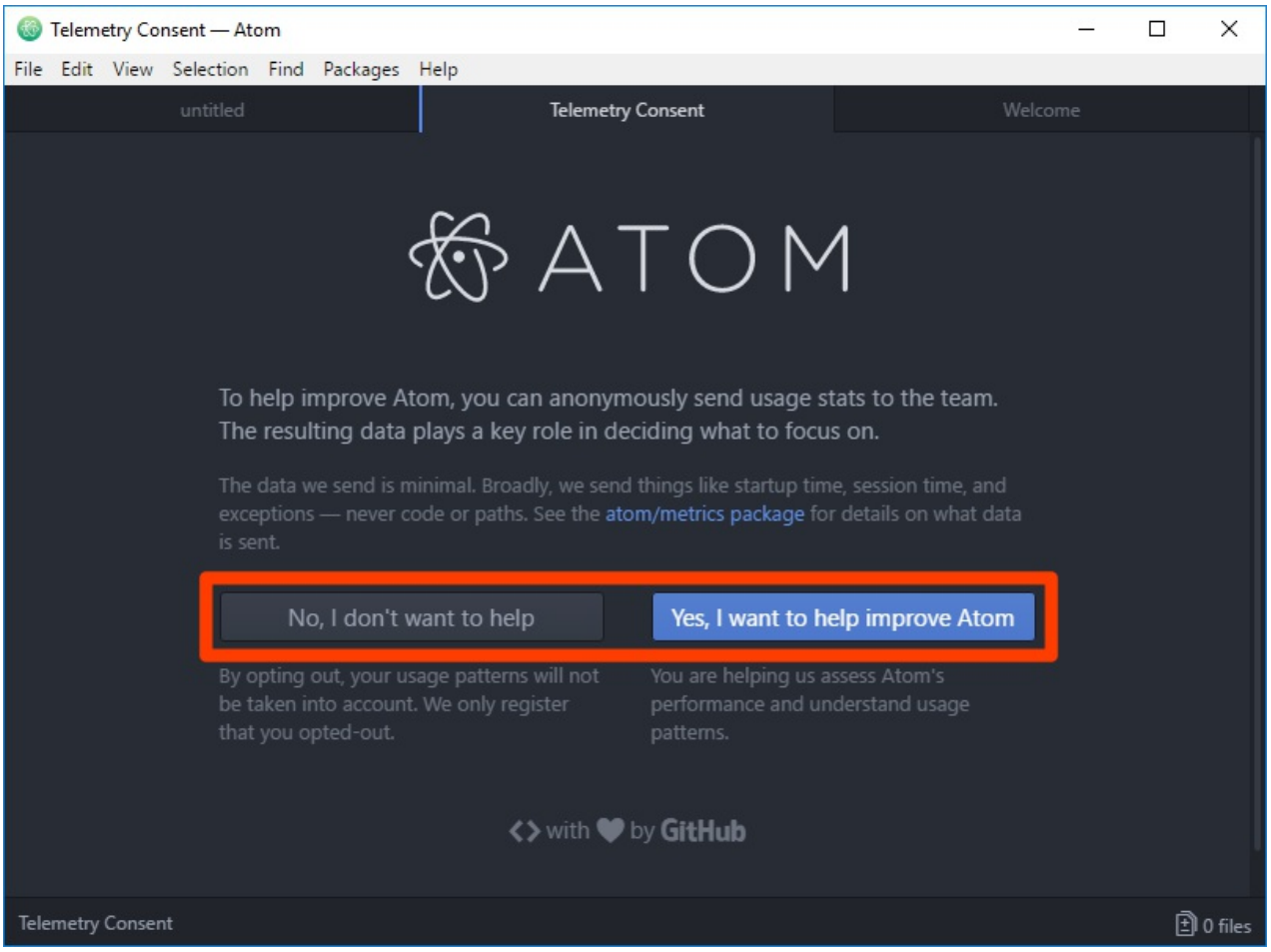
Less Annoying Atom Start-up

When Atom starts for the first time, it'll show you two screens. If you don't select an option you will see these two screens every time you open Atom for the entire semester. Save yourself some time and...

1. On the **Welcome** tab, uncheck "Show Welcome Guide when opening Atom."



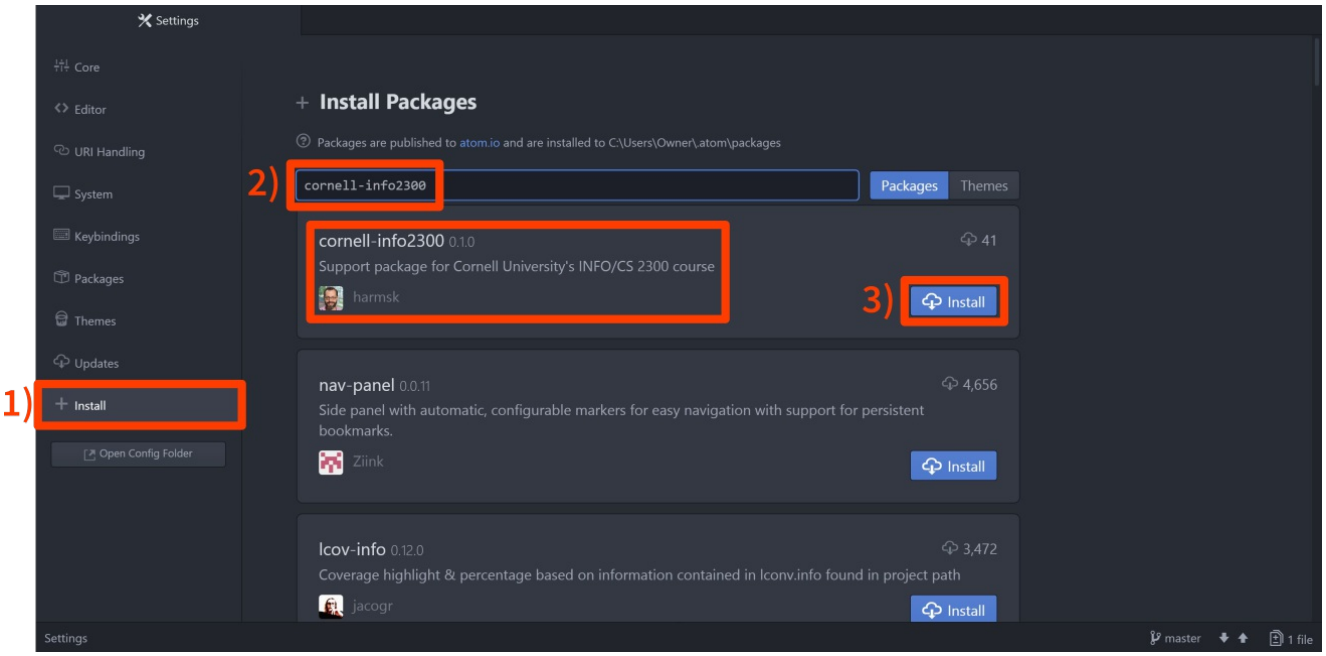
2. Close the welcome tab.
3. On the **Telemetry Consent** tab, make a choice. Select either "No" or "Yes".



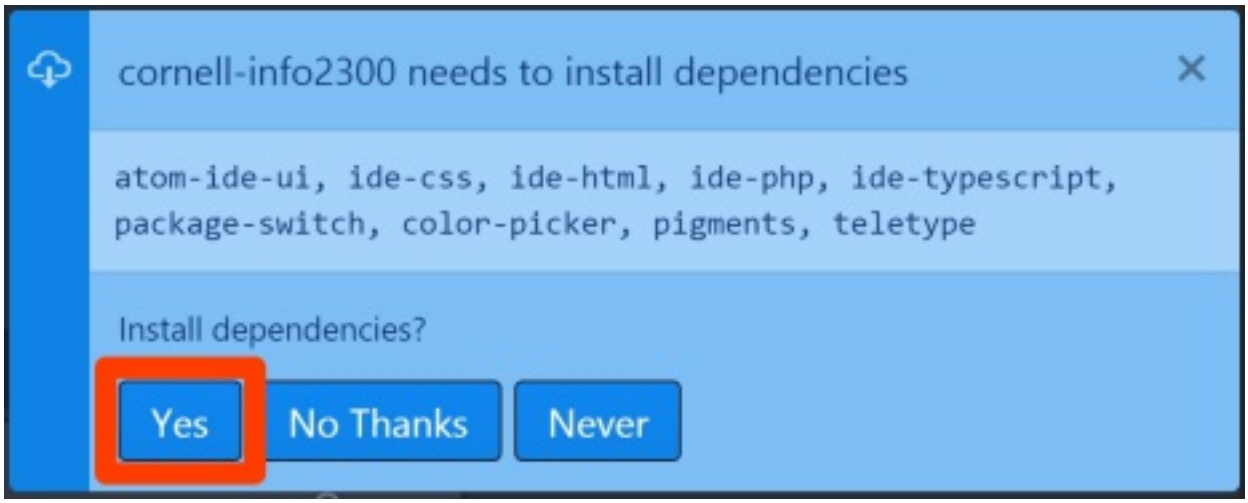
8. Install the Course Package in Atom and Update Other Packages

Installing the Course Package

1. Open the "Settings" panel in Atom (Shortcut *ctrl-comma* for Windows and *cmd-comma* for Mac).
2. Click "Install" and type "cornell-info2300" into the search bar and press enter. Click "Install".



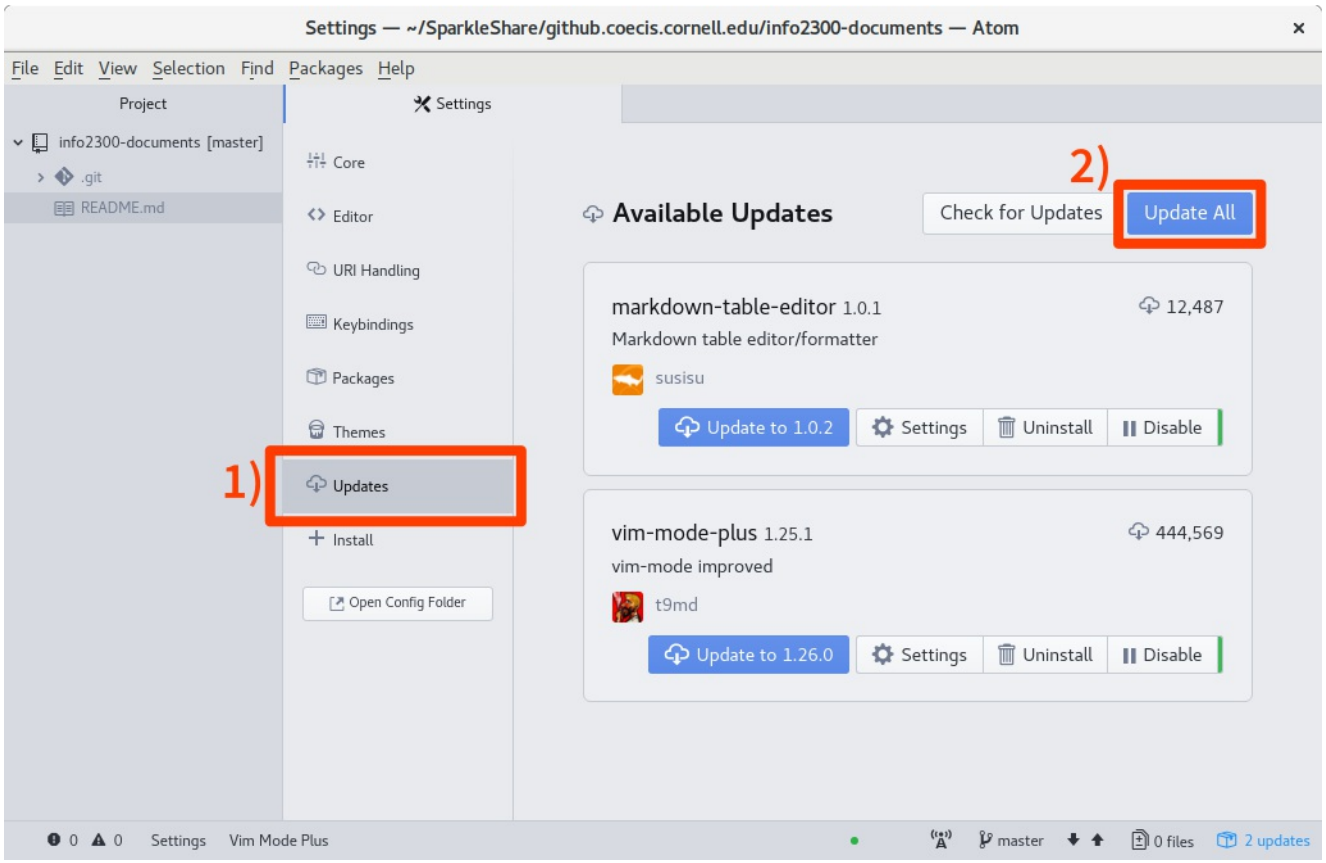
3. When installing, you will be prompted to install several dependencies. Select "Yes" and download all the dependencies. This may take a while, so please be patient.



Update Your Other Atom Packages

1. After the course package is installed, now click "Updates" in the Settings panel.

2. Atom should automatically check for updates. Click "Update All".


Settings — ~/SparkleShare/github.coecis.cornell.edu/info2300-documents — Atom
File Edit View Selection Find Packages Help
Project
info2300-documents [master]
.git
README.md
Settings
Core
Editor
URI Handling
Keybindings
Packages
Themes
Updates
Install
Open Config Folder
1)
2)
Available Updates
Check for Updates
Update All
markdown-table-editor 1.0.1
12,487
Markdown table editor/formatter
susisu
Update to 1.0.2
Settings
Uninstall
Disable
vim-mode-plus 1.25.1
444,569
vim-mode improved
t9md
Update to 1.26.0
Settings
Uninstall
Disable
0 0
Settings
Vim Mode Plus
master
0 files
2 updates

Part III

In part III, you use your newly installed or updated course software to learn how to submit an assignment in this course.

Overview

We'll be using Git to download and submit assignments. Specifically, we'll be using Cornell's GitHub and the info2300-sp2018 organization to distribute assignment files. You may visit main organization's page here: https://github.coecis.cornell.edu/info2300-sp2018.

Git is a version control system. It allows multiple coders to work on a project at the same time. It also enables you to store many versions of your code. This is great for keeping back-ups. We'll cover more about Git throughout the semester, so don't worry if you don't understand all of this right now.

Git uses its own vocabulary. You'll also learn this, but for now this is a helpful guide for translating between English and Git.

English	Git
assignment	repository
download assignment	clone
save a version of your code	stage (add) and commit
submit assignment	push
download assignment updates from the instructor	pull or fetch and merge

1. Download Assignment Files - Clone a Git Repository

As mentioned earlier, we will be using the Cornell's GitHub to distribute assignments as well as for students to submit assignments.

In this section we'll clone a copy of the assignment repository from the GitHub server to your local computer.

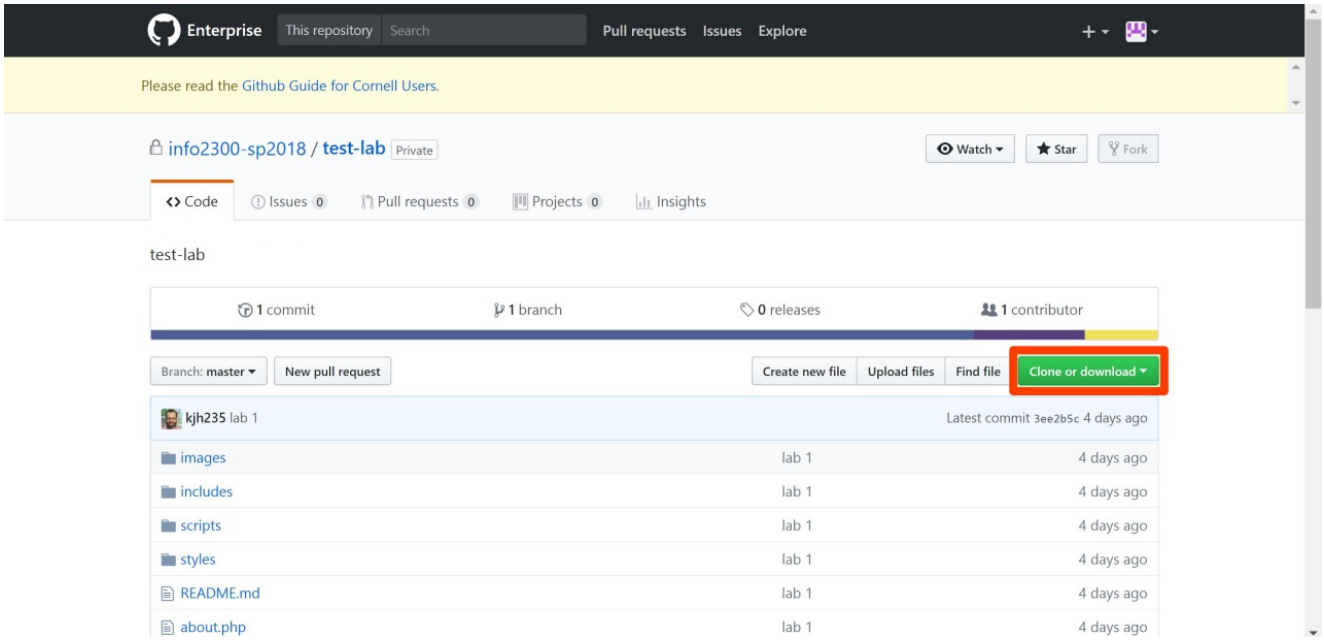
`git@github.coecis.cornell.edu:info2300-sp2018/YOUR_GITHUB_USERNAME-lab1.git`

https://github.coecis.cornell.edu/info2300-sp2018/YOUR_GITHUB_USERNAME-lab1

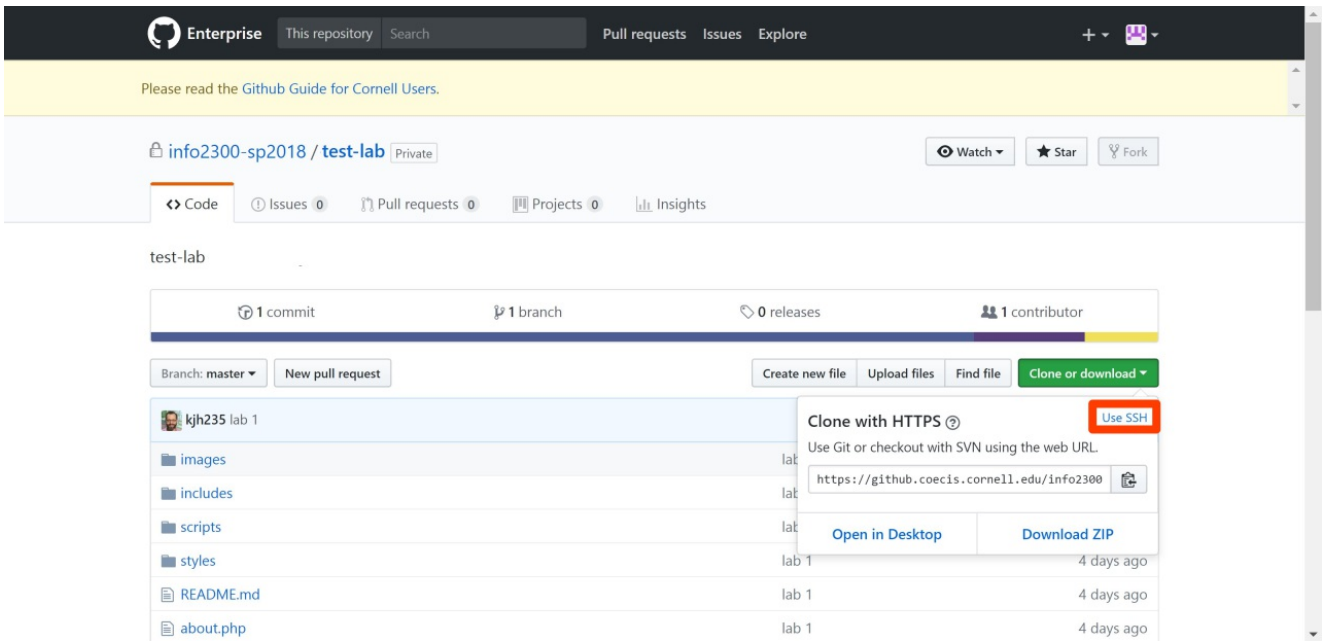
1. Visit the web page for your assignment repository. Visit the following URL in your web browser. Change **YOUR_GITHUB_USERNAME** in the URL to **your actual GitHub username** for most of you this should be your NetID.

2. We want to clone a copy of the assignment from the GitHub server to your local computer.

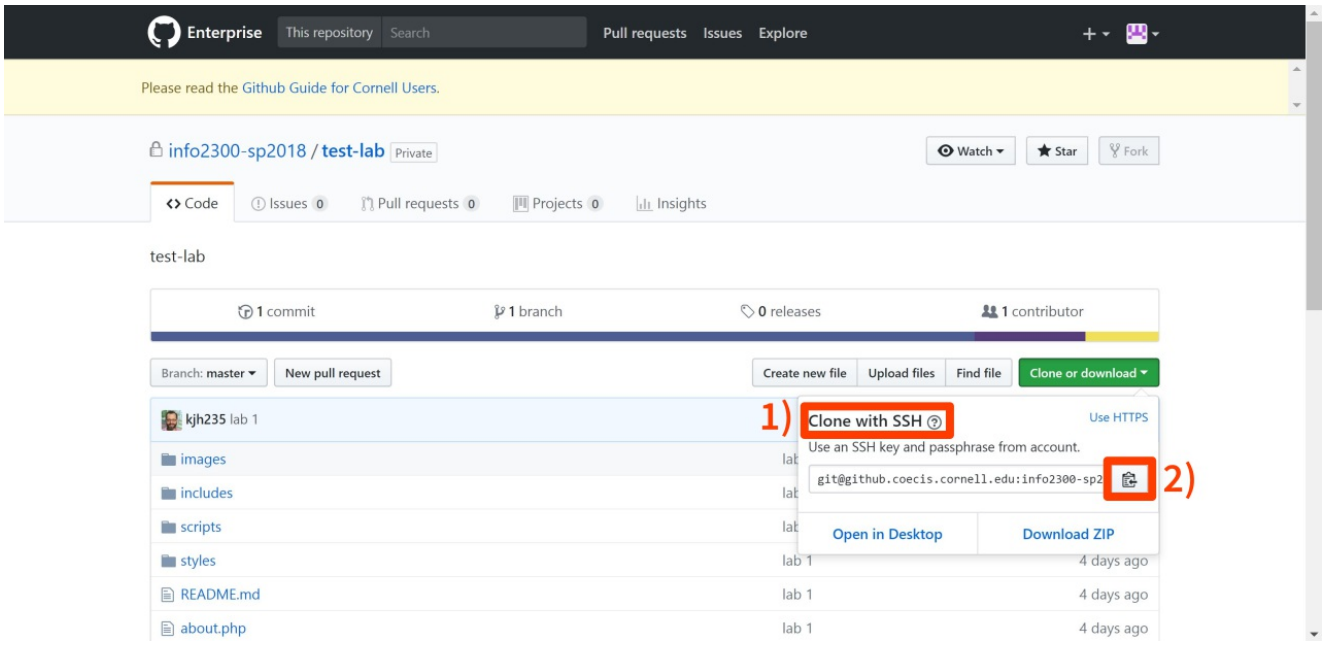
You will be able to see all the files in the repository. We want to clone a copy to your local computer so you can edit the file. Click the green button that says "Clone or download".



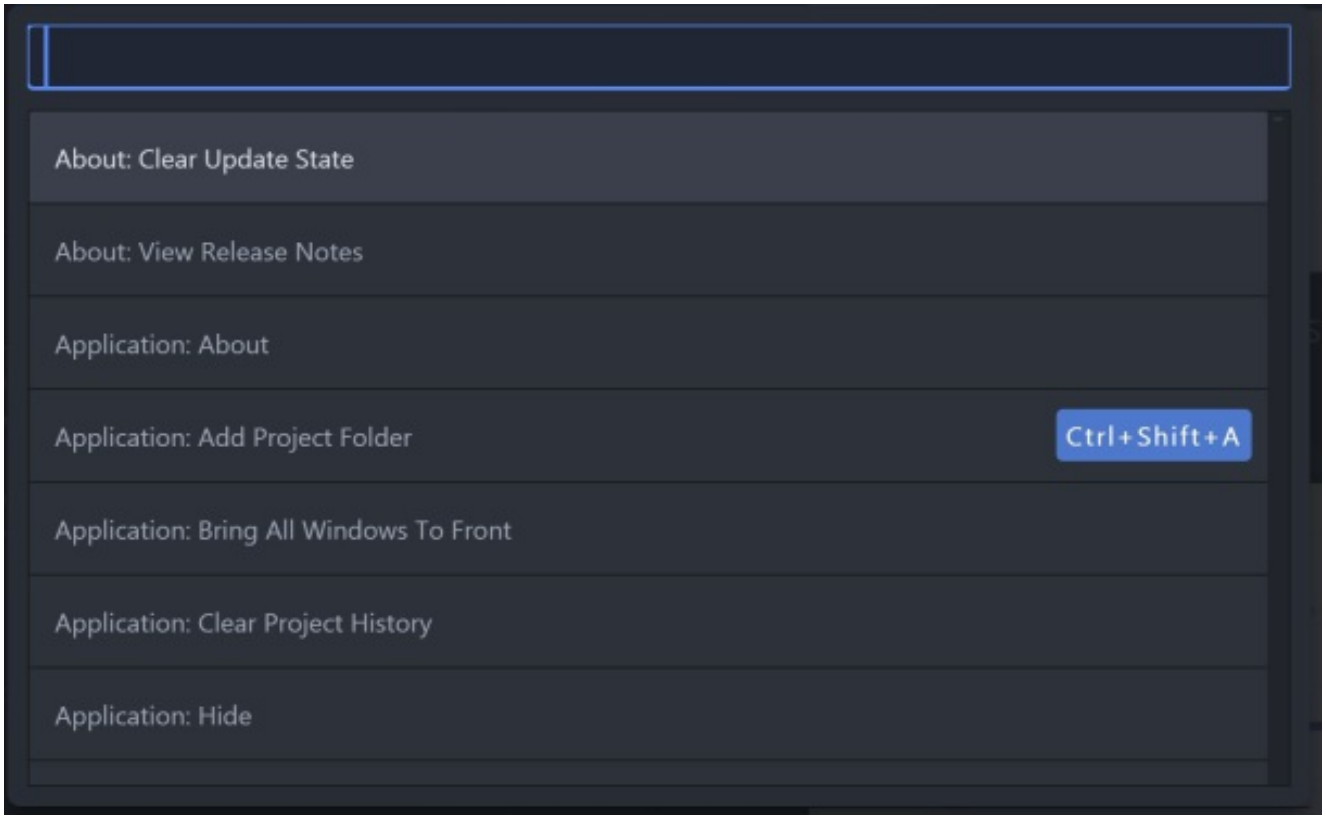
5. A little box will dropdown, click "Use SSH" in its top right corner.



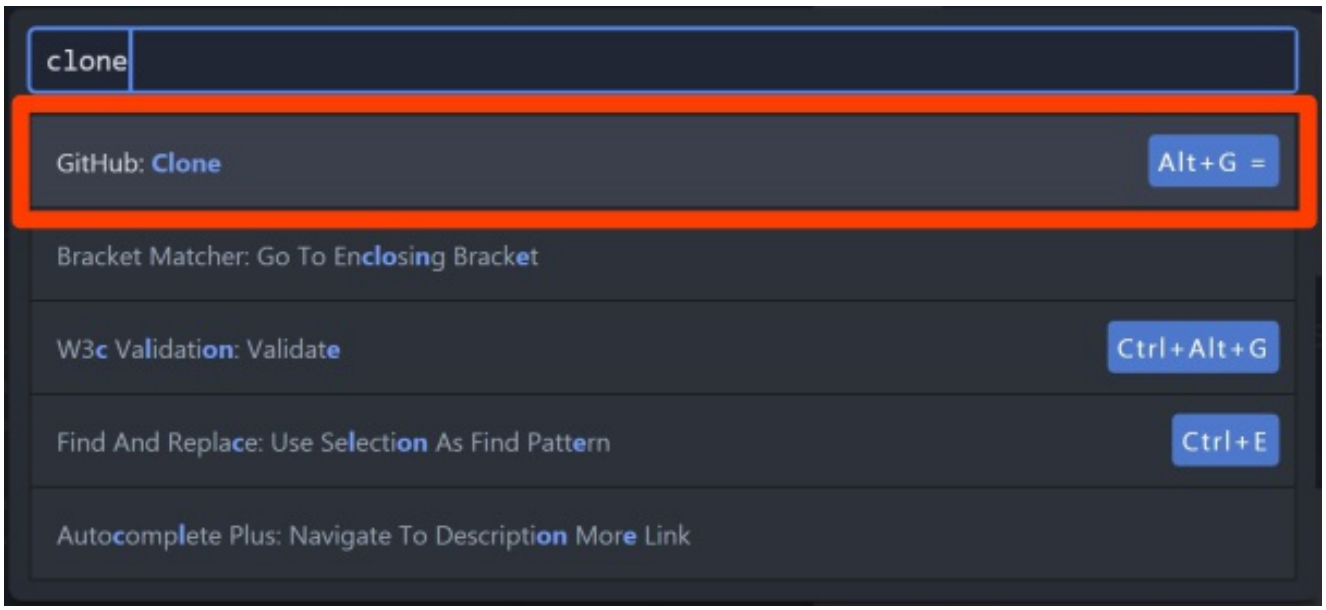
6. Make sure that you see "Clone with SSH" on bolded on top; if so, click the clipboard button to copy the link provided.



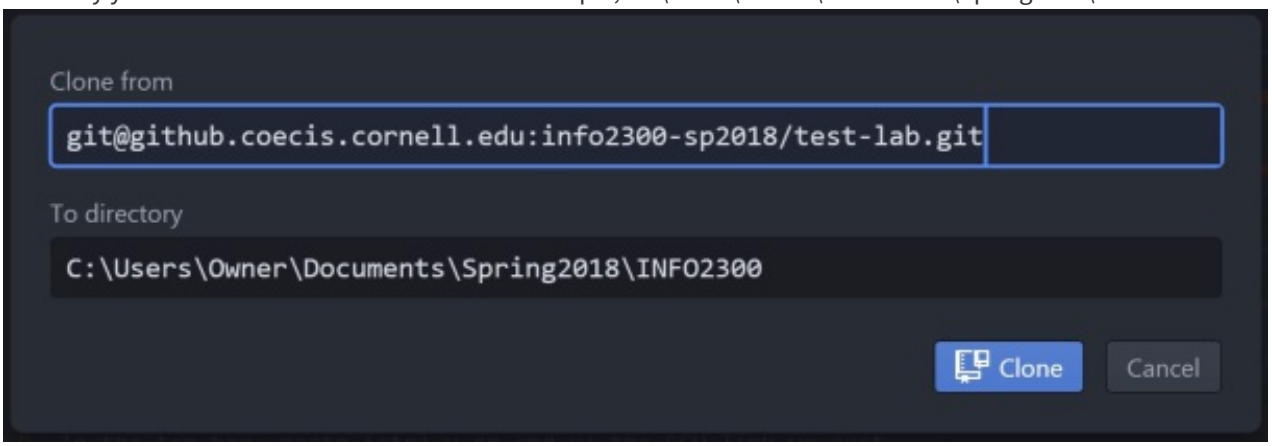
7. Open the Command Palette (ctrl-shift-p for Windows and cmd-shift-p for Mac).



8. In the Command Palette, type "clone" select the first choice, "GitHub: Clone".



9. In the top bar, paste (ctrl-v or cmd-v) the SSH link we just copied from GitHub. On the bottom bar, write the directory you want the files to be saved in. For example, "C:\Users\Owner\Documents\Spring2018\INFO2300".



10. Click "Clone" and you are good done!

10. How to open a folder as a project in Atom

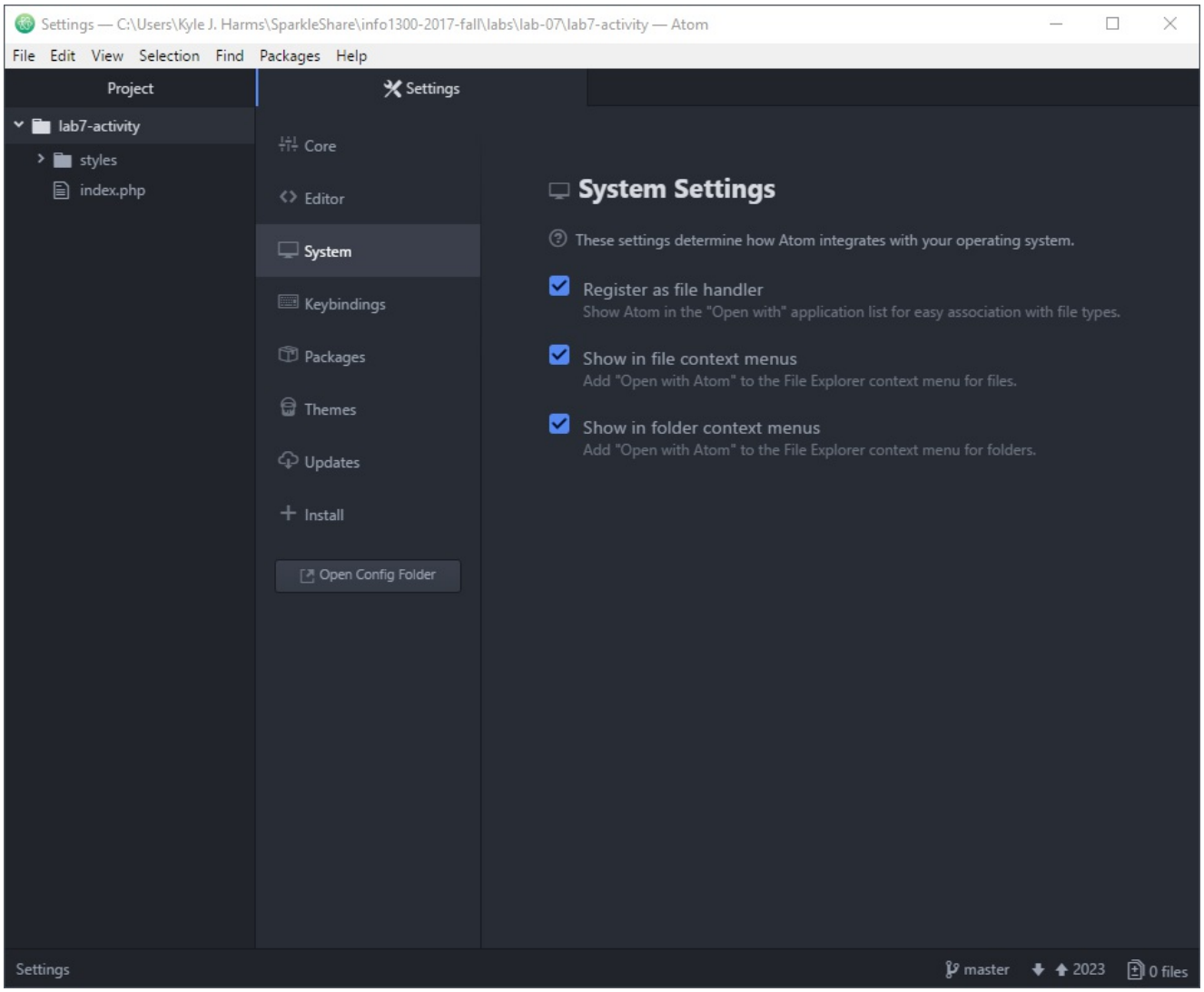
Atom works best when we work on *projects* rather than *just opening files* in Atom. You will want to follow this advice particularly while working with PHP otherwise things may not work as expected.

Keep in mind that a **folder** on your computer (in your file manager) is a **project** in Atom.

Now that we have cloned the assignment files from GitHub, let's open them up to work on them. To open the folder [TODO: Kyle: Name of folder] in Atom, right click on the folder in your file manager and select **"Open in Atom"**. If this works, go to the next section.

If you do not see this option, then you might need to turn on these menus in Atom. On Windows, open Atom, go to the Preferences or Settings, and select the **System** tab. Check the register boxes as necessary (see the figure below). For your Mac, check out this web page: https://blog.brettski.com/2015/07/09/adding-open-in-atom-in-secondary-menu-

on-a-mac/



11. How to run a PHP server through Atom

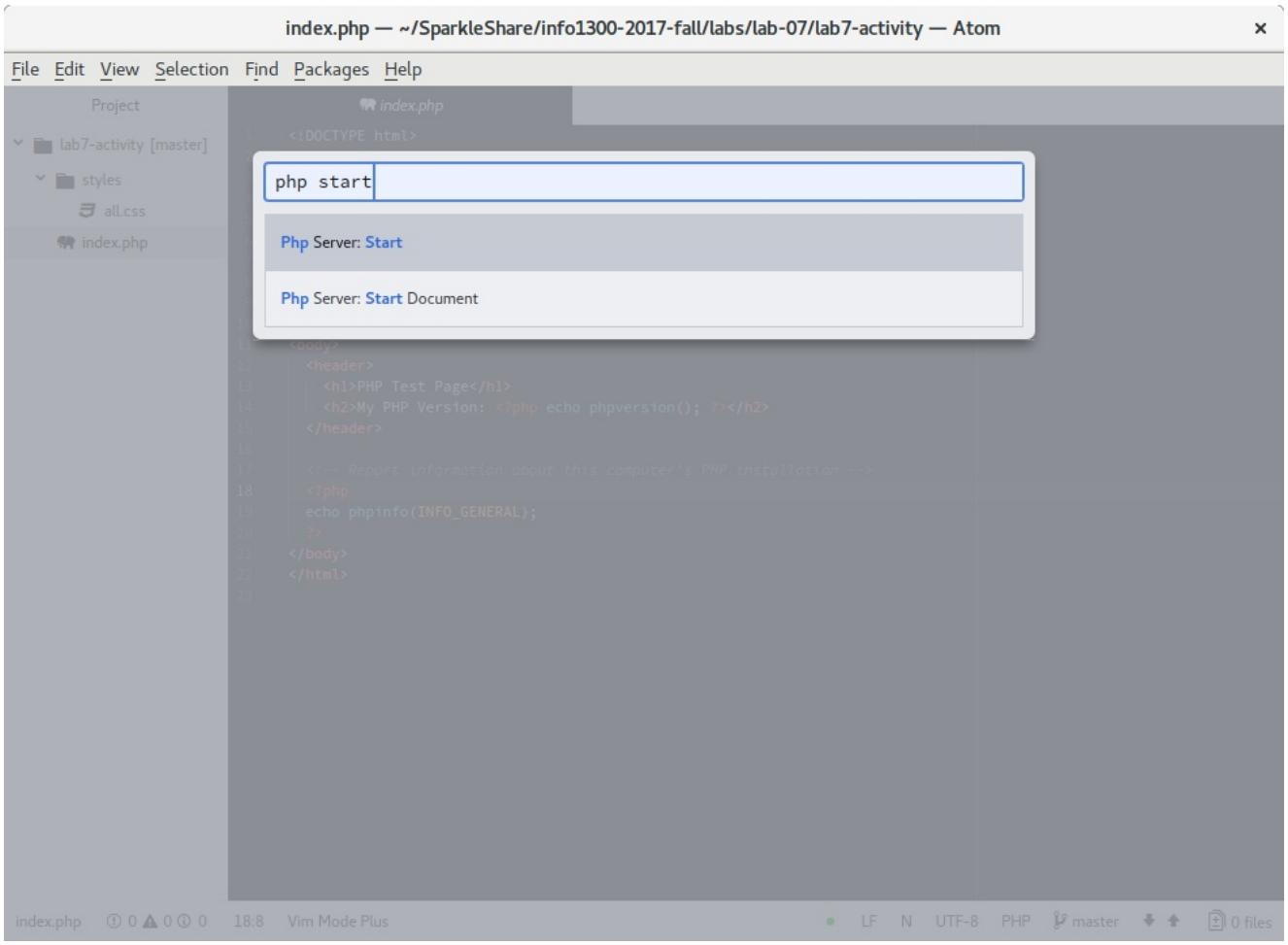
Now that you have PHP installed on your machine, we are going to configure atom to run PHP web sites.

In this class, the teaching staff will *only* support Atom. You are free to use any other text editor (Sublime, Brackets, etc.) and mechanism to run PHP (MAMP, etc.) however, **we will *not* help you if run into issues**.

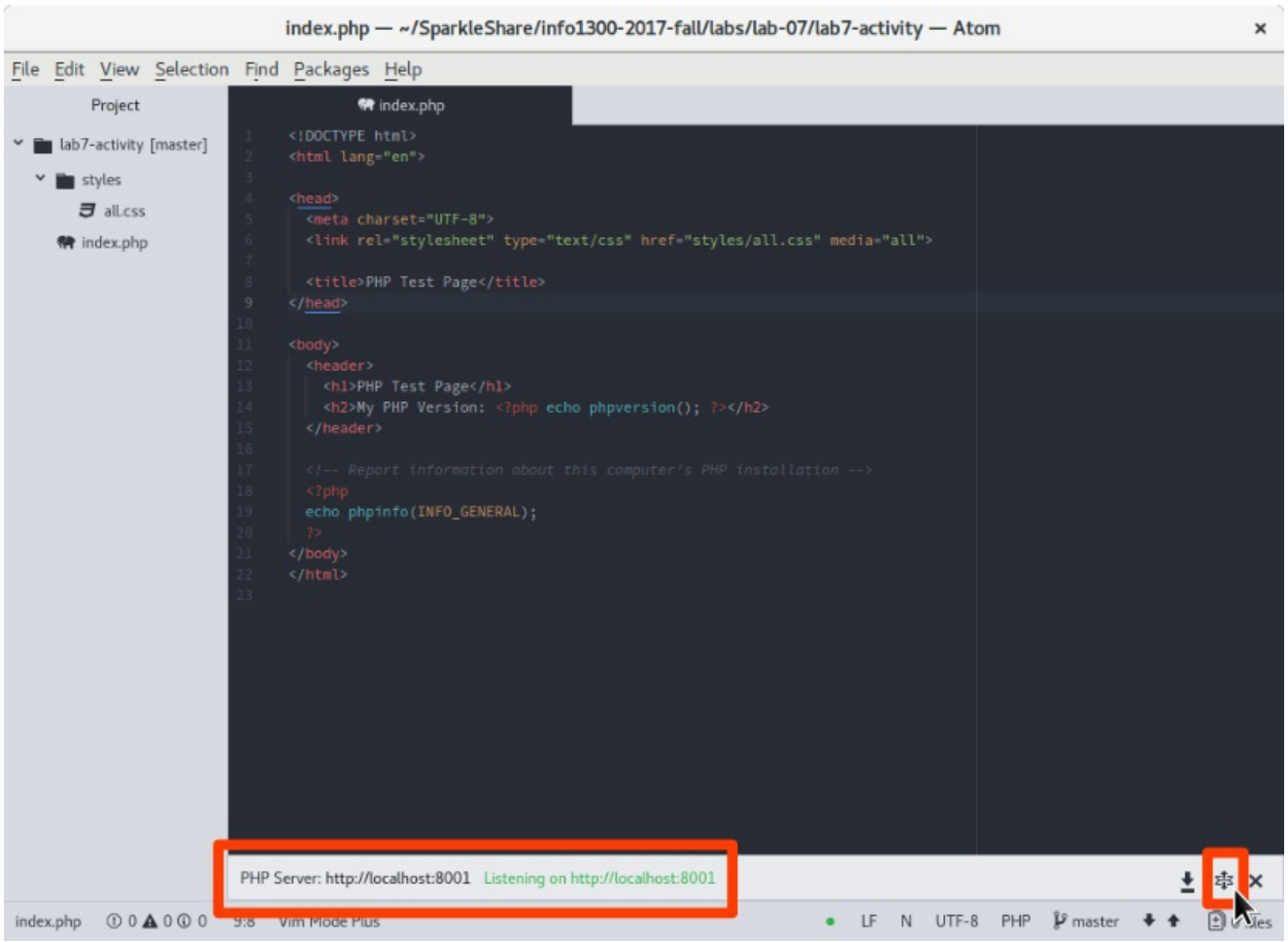
Make sure you *only* have the [TODO: Kyle: Name of folder again] folder/project open in Atom. If you can't start PHP or you don't have a least version 7.1. or need help in general, please ask a TA.

TODO: Sharon: Write this in TA instructions: If you are overwhelmed with questions or you don't know how to fix the problem, tell them to not worry about it now and to go to office hours.

Now open Atom's **Command Palette** (ctrl-shift-p and cmd-shift-p) and search for "**php start**" and press **enter**.



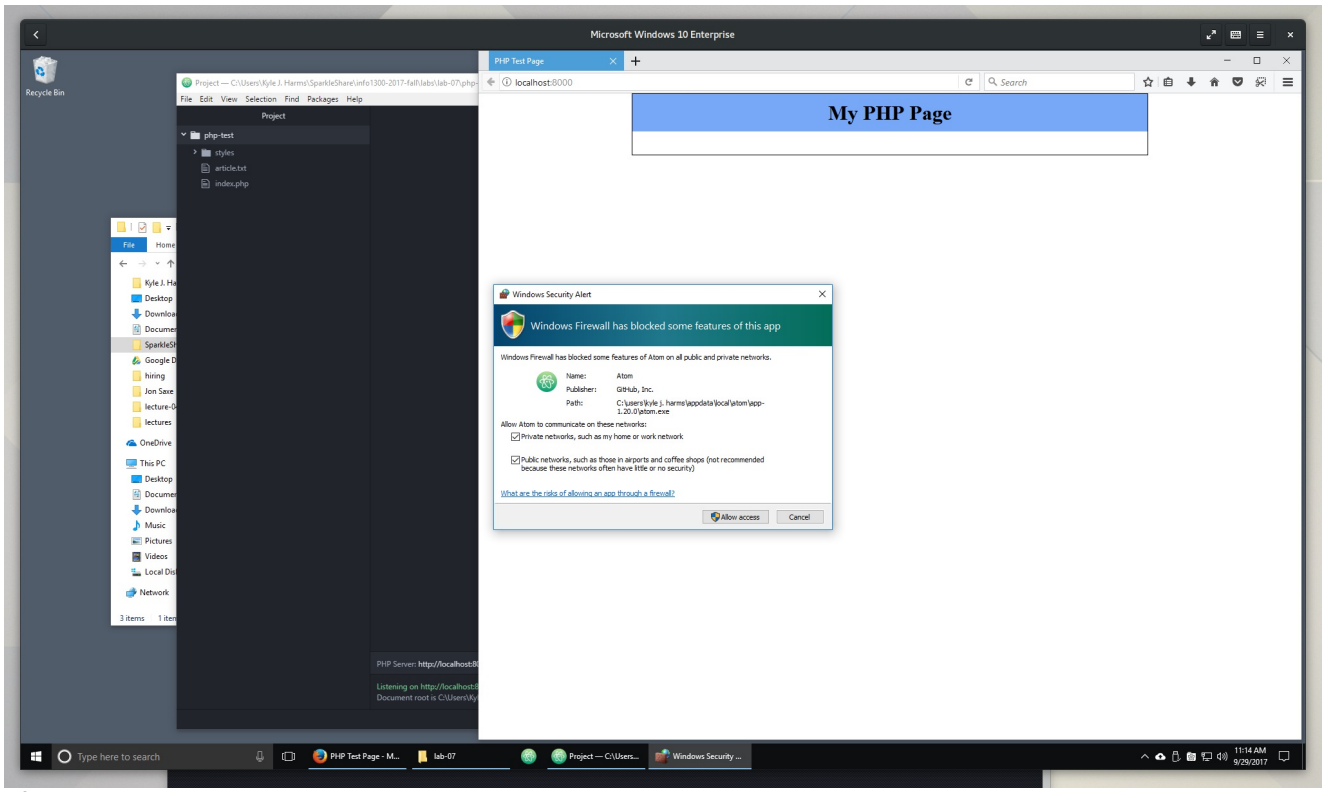
You will now see the PHP server starting in the bottom panel of Atom. See the figure below. On the left side you will see a URL for the local php server. It's usually http://localhost:8000/, however sometimes the server might be different (e.g. http://localhost:8001/). You will need to pay attention to this URL because it's how you will access your PHP server in your web browser. Note: you only need to start the server once when using Atom.



The PHP server panel takes up a lot of space in Atom. It really shouldn't do that, it's bad design. Let's get some screen real estate back by clicking on the collapse button as highlighted in the figure below.

If you are on Windows, the Windows firewall may complain about Atom. Just grant it access to your networks.

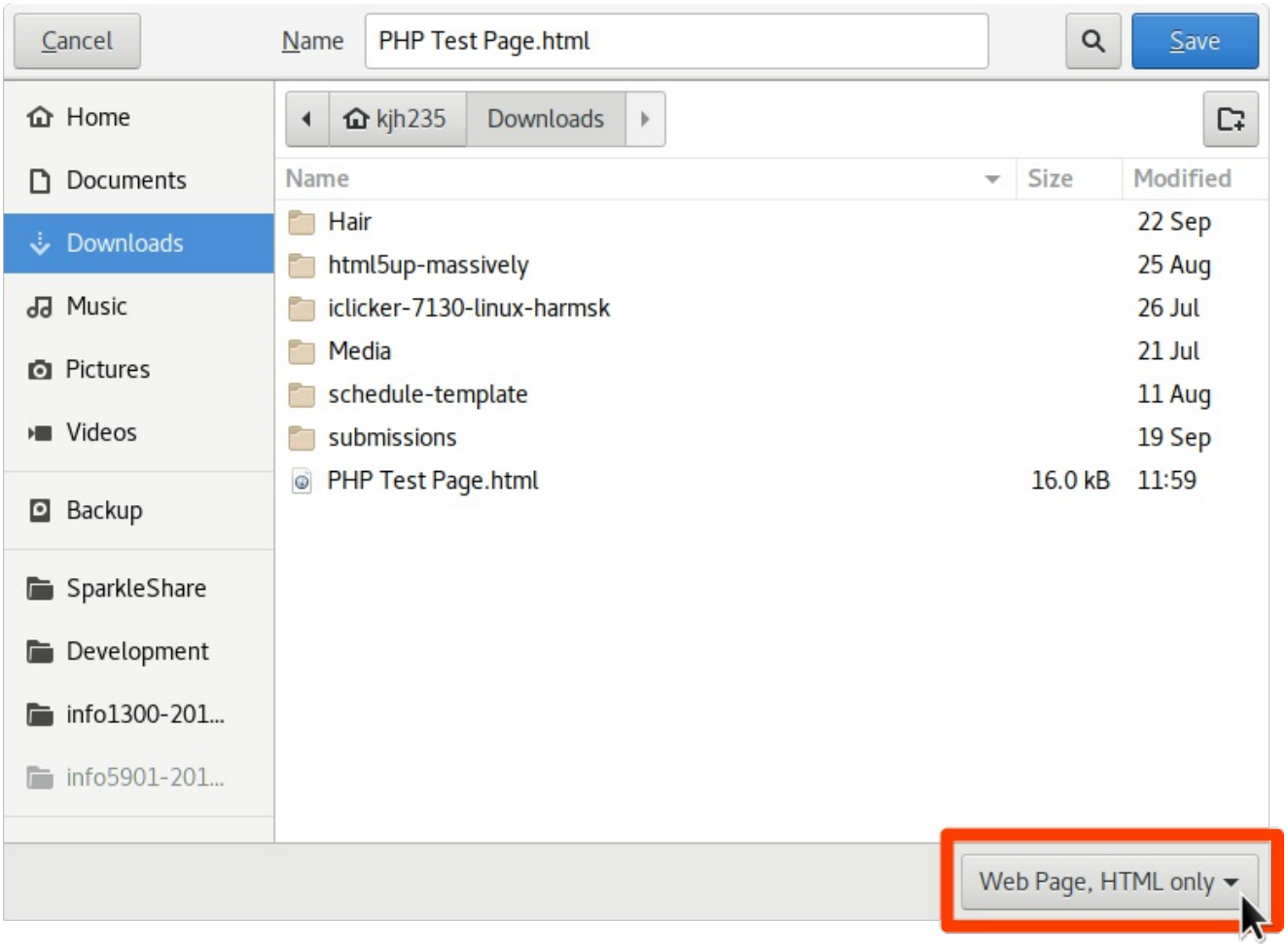
REVIEW: Kyle & Sharon: This section isn't confirmed until the lab material is finished.



After starting the server, your web browser should have opened a page that says “PHP Test Page.” This test web page was generated by the PHP code inside of the **index.php** file in your project!

TODO: Sharon: Need to change based on submission method

If the web page opened in Chrome, close it and **visit the URL of your local PHP server in Firefox**. Now save a copy of this generated web page and submit it to CMS. From the **Firefox menu** select “**Save Page**.” Next, change the format in the Save dialog to “**Web Page, HTML only**.” Upload the saved file to CMS in the Lab 7 assignment.



TODO: Kyle: Instructions for what to edit/change

12. How to submit your assignment

Now that you made edits to the files we cloned from the repository, we now have to push these changes into the repository as well.

1. On the bottom right corner click on icon that says "1 file".

TODO: Sharon: Screenshot of 1 files

2. A new panel should pop up with "Git" on top. Under the section "Unstaged Changes", you should see the name of the file(s) you edited. Click "Stage All" in the top right corner.

TODO: Sharon: Screenshot of Git panel

3. All the files should have moved into the section "Staged Changes".

TODO: Sharon: Screenshot of Git panel staged

4. On the bottom there is a text box saying "Commit message". Write a small message describing the changes made. When you make multiple commits, this allows you to differentiate the different versions and changes you made.

For example: "Added my name". After writing a message, click "Commit".

TODO: Sharon: Screenshot of message and commit

5. Now that you have committed, you have to push the changes into the repository. On the bottom right corner, next to the icon that said file, there should be two arrows pointing down and pointing up. Click it and a small box will pop up.

TODO: Sharon: Screenshot of push/pull

Note: We have to push for the changes to be updated in the repository. Commits are only **LOCAL** to the computer.

6. Click push and if you go back to your GitHub, you should see that your files were updated.

Note: If you were asked for a username and password, this means the SSH key was not completed correctly.

7. Congrats! You have finished the lab and you are set up for the rest of the semester.

TODO: Sharon: Make a commit presentation TODO: Kyle: Make a looping video

IMPORTANT: This is only half of your participation grade; both attendance and participation are required to receive credit. If you fail to complete this during lab section, you may submit this afterwards so long as you also attended lab.