Getting Started with Jupyter Hub

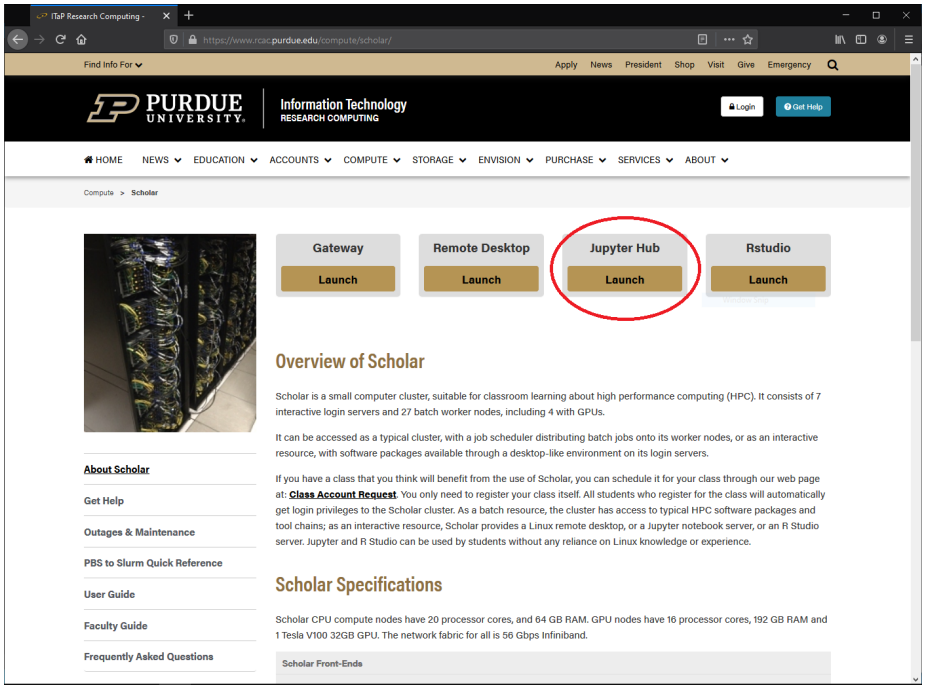
Introduction

For this course, we are going to be using Jupyter Hub. Jupyter Hub is a popular utility in the data science field. One of the more tedious parts of programming is getting your coding environment setup on your personal computer. Jupyter Hub allows us to avoid doing this. It is a personal cloud programming environment that is already setup for you. All you have to do is login, then you can begin programming in a web browser.

Another benefit of Jupyter is that it allows programmers to run Python code in an visual and interactive environment. Jupyter Hub is comprised on Jupyter Notebooks, which are essentially Python files with text between code snippets (Jupyter can also be used with Shell and R scripts). These text segments can be used to explain each the code. After each of the code segments, you are able to see the output. For example, if we make a graph in Python (which we will), that graph will appear after that respective code segment.

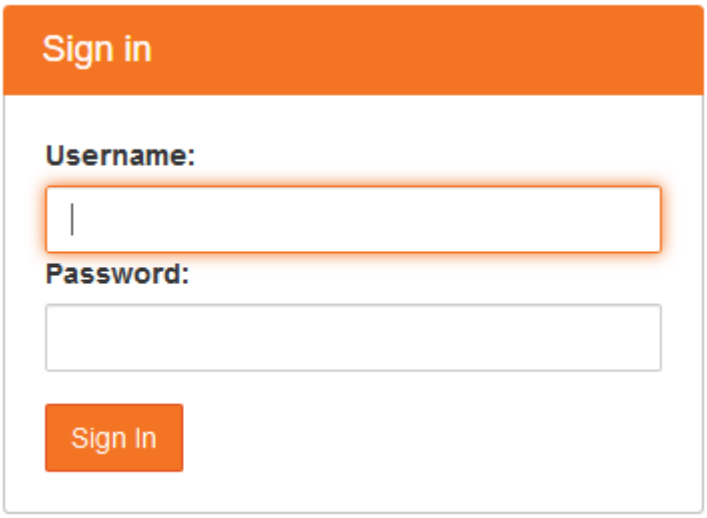
Throughout the Python portion of this course, we will be using Jupyter Notebooks for class activities. We will upload modules to Brightspace. All you have to do is download these modules off of Brightspace, upload it to Jupyter Hub, then follow along with the notebook. When programming, practice and experimenting with code is essential to learning. This is why we have chosen this approach. It will allow you to follow along with the lessons as you practice with the code.

Step 1: Launch Jupyter Hub



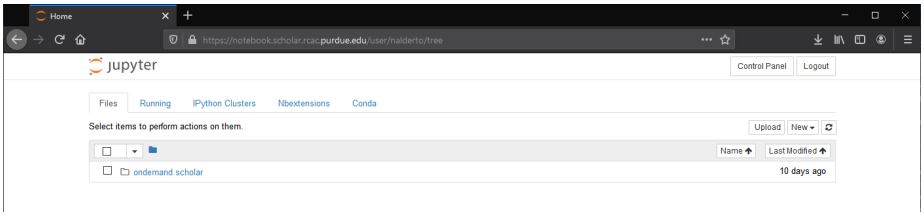
The instance of Jupyter Hub that we are going to use for this class is on a powerful Purdue server known as Scholar. Scholar's resources can be accessed here. To access Jupyter Hub, simply click the "Launch" button under the Jupyter Hub title.

Step 2: Log into Jupyter Hub



Next, you need to type in your credentials. Your username is your Purdue Career Account username, which is the part before the "@purdue.edu" in your email address. For example, if your email is "ewaltenb@purdue.edu", then your username would be "ewaltenb".
Your password is your Purdue Career Account password. This is **NOT** your BoilerKey password with with "push" or "pin." This is the password that you use to log into Office 365 and Outlook.

Step 3: Welcome to Jupyter Hub!

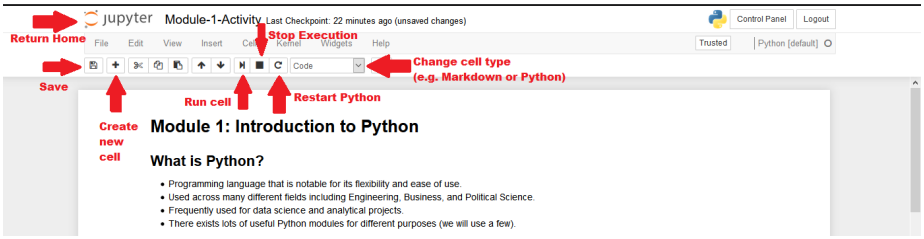


This is the main Jupyter Hub page. It is essentially a file explorer. When you create or upload a notebook, it will appear here. You can even create folders to organize your Notebooks, if you so wish. To create new notebooks or folders, click the "New" button on the top right.

Step 4: Upload a Class Activity

All of our class activities are going to be uploaded to Brightspace as a Jupyter Notebook. Download the notebook from Brightspace, click the "Upload" button on Jupyter Hub, select the appropriate file, then click the blue "Upload" button.

Step 5: Opening a Notebook



Simply click the name of the newly uploaded notebook and your are ready to starting coding! Once the notebook is open, it should look something like this. We are going to quickly go over the most important components of the toolbar.

- The Jupyter icon in the top left will return you back to the Jupyter Hub page.
- While Jupyter has autosave functionality, it is always a good idea to save your code regularly with the save button.
- If you want to create a new "cell," click the plus button. A cell is just Markdown or Python snippet. Markdown is a basic markup language that is used to format text with headers, bullet points, bold, etc.
- The button that looks like the next track button on Spotify is actually how you run your cells. After you write some code, you can click that button and it will execute the code. The output of the code will appear underneath the respective cell.
- Sometimes your code won't work right. For example, maybe it is running for a really long time. You can click the black square stop button, and your code will stop executing.
- The circular arrow icon will restart Python. If you are experiencing any issues with Jupyter (e.g. it's running slow), you can click that button, and everything should be fixed (hopefully)!
- The dropdown will allow you to select your cell type. If you want to write some text, you can select Markdown. However, 99% of the time, you are going to be selecting code (a.k.a. Python).

Step 6: Running Code

Jupyter Notebooks are organized as a series of "cells." As was mentioned above, these cells contain either text or code. The toolbar controls described above control the currently selected cell. This means when you click the "run" button, it will only run the code in the currently selected cell.

It is important to note that cells can interact with one another! For instance, if you define a variable in the first cell, it can be accessed in another cell. The order in which you run the cells also matters. If you define a variable in the first cell, but you don't run it, then you will not be able to access that variable in a later cell.

You can determine which cell is currently selected based upon the highlighting around the respective cell. Here is what a currently selected text and code cell look like.

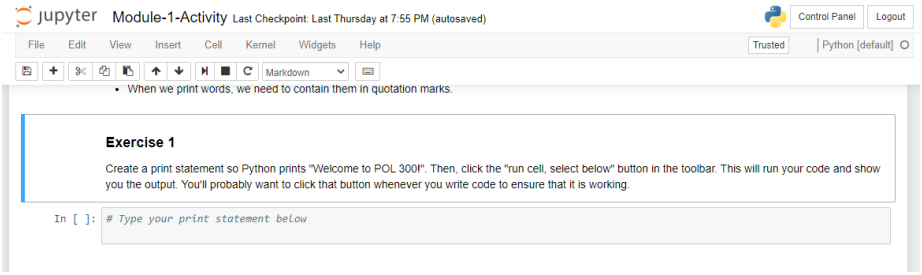


Figure 1: A text cell that is currently selected

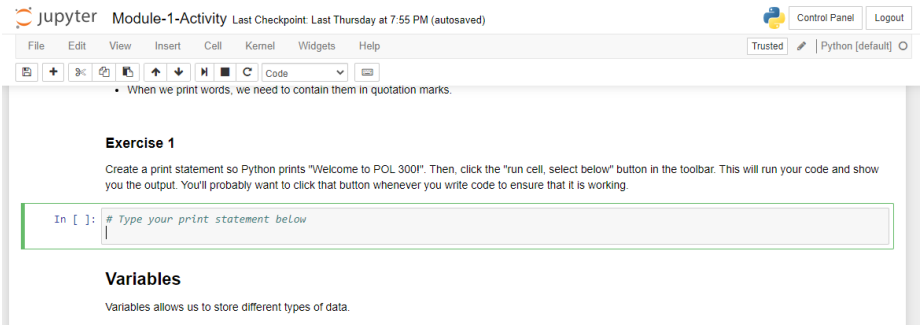


Figure 2: A code cell that is currently selected

Switching which cell is currently selected is as simple as clicking on a different cell.

Step 7: Exporting a Notebook

Once you have finished the exercises in a notebook, you need to submit the notebook for grading via Gradescope. **FOLLOW THESE STEPS CAREFULLY AND COMPLETELY, OTHERWISE YOU MAY NOT RECEIVE FULL CREDIT.**

1. In the toolbar, click "Kernel", then "Restart & Run All".

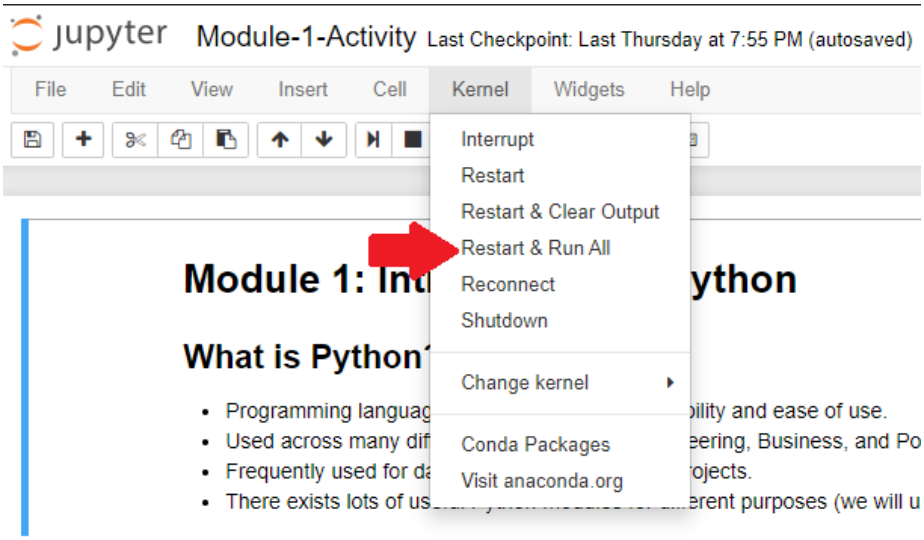


Figure 3: Select "Restart & Run All"

2. **Review your notebook!!!** There is a chance that running all the code again could change your results. The "Run All" command will run all of your cells in order from top to bottom.
3. Back in the toolbar, click "File", "Download as", then "HTML (.html)". This will download your notebook with the `.html` file extension.
4. Open up the HTML file with your browser of choice. In Google Chrome, you should to be able to click on the HTML file in the downloads bar on the bottom of your browser window. In other browsers, go to the downloads location, and click on the file. The HTML file will be opened by your browser.
5. Next, open the print dialog box in the browser tab of your recently downloaded HTML file. The print dialog will usually appear when you use the keyboard shortcut Ctrl-P (Cmd-P on Mac).
6. Select the option to "Save as PDF". The verbiage may vary depending on your browser or operating system. For example, in Firefox on Windows

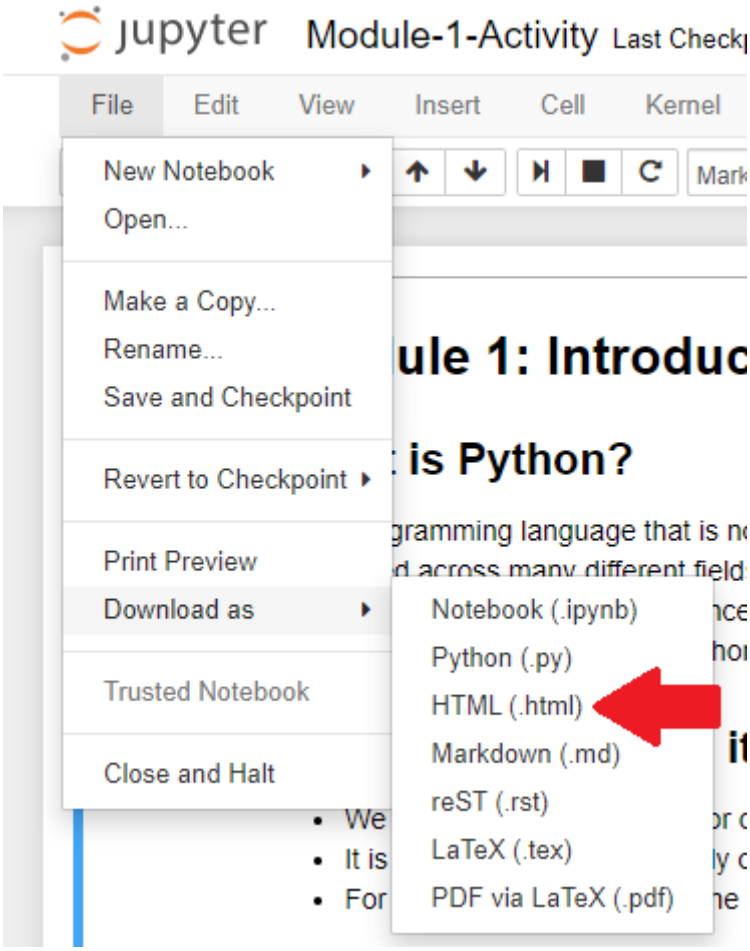


Figure 4: Download as a HTML file

10, there is an option of "Microsoft Print to PDF". As long as the HTML page is now saved as a PDF, it should suffice.

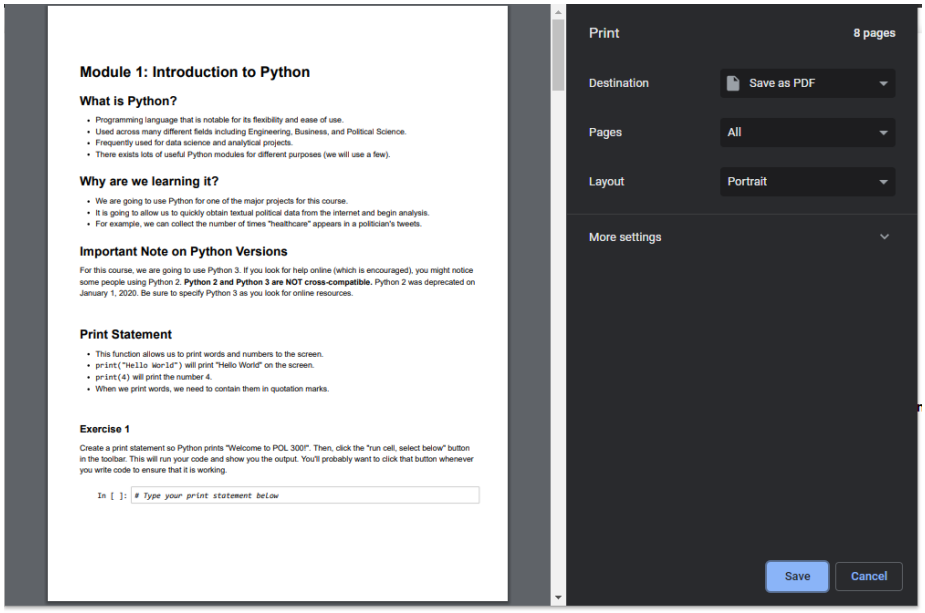


Figure 5: Save as PDF

7. Finally, go to Gradescope, select the assignment, and upload the PDF file we generated in the previous step.

Now you should hopefully be ready to start programming in Python using Jupyter Hub! If you are experiencing any issues or have question, please ask!

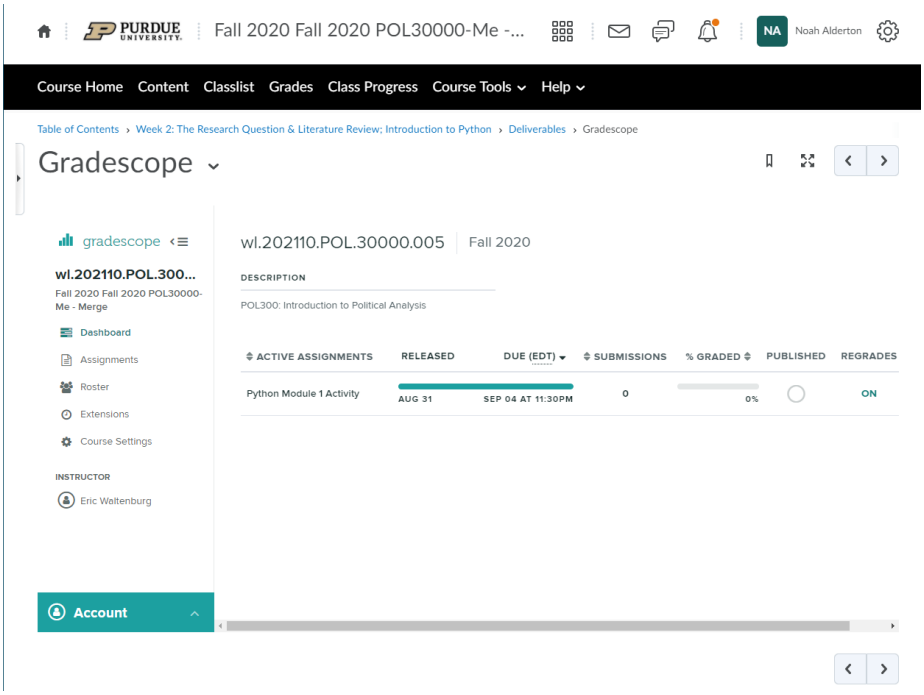


Figure 6: Gradescope Landing Page