

Jupyter Notebook Overview:

A Jupyter notebook is a “REPL” that runs in a web browser. A REPL stands for a “Read-Execute-Print-Loop”, and it is an interface that allows python code to be run within cells in a web browser.

The **cell** is the main unit of interface and is typically either a block of code (an *Input*), or text markdown. Each block of code will have a corresponding *Output* cell (which can be referenced directly if need be as a variable as ‘Out[#]’).

Jupyter notebooks were first introduced for data science, as the more immediate and responsive REPL format allowed for faster insights and easier presentation and portability.

Notebooks have also become a standard way to present information to stakeholders, including non-technical stakeholders - for the latter can port this over to a Colab. Great way to walk through equations. In fact I think a GREAT way to present maths with its integrated latex.

One side note: when it comes to environment setups and some of the nuances of Python and coding in general - basically things I’ve only done once or twice when learning them - I may not know off the top of my head how to do something but can always figure it out (like the issue where the Tensors weren’t running the other day, once I had 5 minutes I figured it out).

Ok, to get started with Jupyter Notebook, you must download it using **pip** (this is Python’s internal protocol for downloading and upgrading packages and libraries). You will use pip for all library and package downloads. There are other programs to download these things but we’ll just stick to using pip for now.

Installing:

1. Open Terminal:

It should look something like this:

```
ryanmiller-alt@Ryans-MacBook-Air-4 ~ %
```

The important thing to note is that in the terminal, everything to the right of the % is input, and the tilde itself (~) represents that top level directory, which is where you start. Before that is some computer id mumbo jumbo. I will remove all the computer identification beforehand for the rest of this document but leave the directory we are in. For the first few steps you can be in any directory so might as well be in the top level.

2. Update Pip:

Use pip to update itself, although your version of Python is new enough you might see something like “Requirement already satisfied” and/or a Deprecation warning.

```
~ % python3 -m pip install --upgrade pip
```

3. Download Jupyter Notebook and other important libraries in the terminal using pip:

```
~ % pip install jupyterlab
~ % pip install notebook
~ % pip install numpy
~ % pip install astropy
```

```
~ % pip install einsteinpy
~ % pip install matplotlib
~ % pip install seaborn
~ % pip install simply
```

Type the “pip...” in one at a time and press enter. The terminal should start scrolling downloading looking messages.

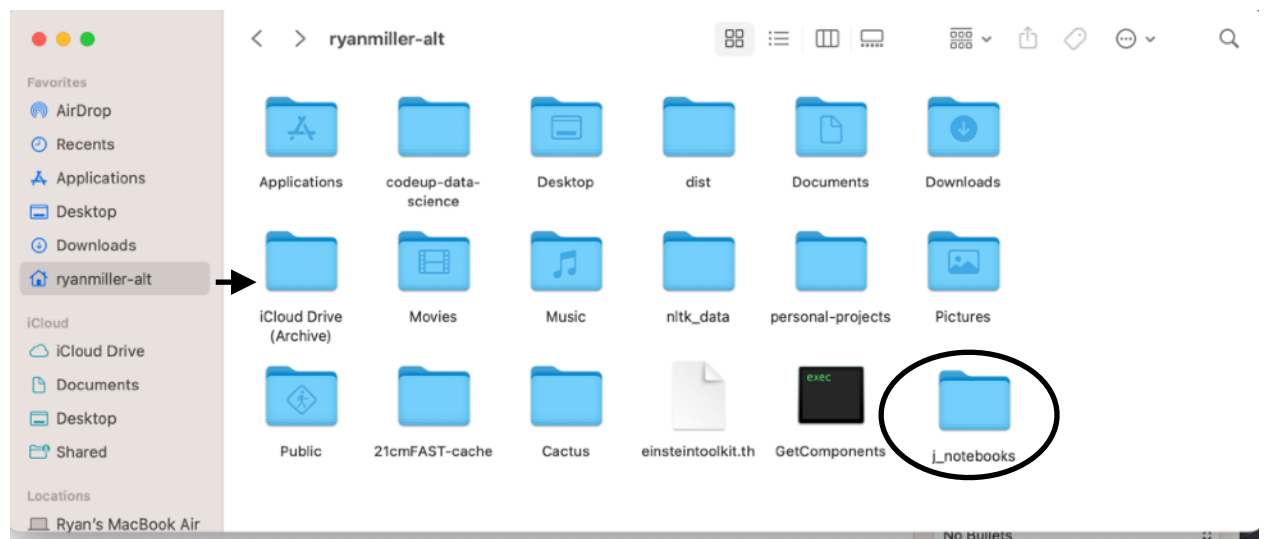
These should be all the libraries you need to get started.

- **jupyterlab** and **notebook** are for Jupyter Notebooks.
- **numpy** is the foundational python program for linear algebra that powers pretty much every advanced computational library.
- **astropy** includes units and other things used in einsteinpy.
- **einsteinpy** is what we are trying to figure out.
- **matplotlib** and **seaborn** are both plotting programs. Seaborn is a library that lives on top of Matplotlib and allows for additional functionality if needed.

Note: In general, any time you encounter a situation where you need to use a specific library, or if you try and use a library or package and you get an error saying it doesn't exist, the solution is typically a pip install of that library.

4. Create a Folder to Hold Notebooks

This can be done using the Terminal, but I prefer to do it in the “non-technical” way. Go into the finder and navigate towards the top level directory, which should be at the bottom on the left side (for example, mine is ‘ryanmiller-alt’ - it will be whatever the terminal computer id says). Then simply right click to create a new folder. Name it something like ‘j_notebooks’ for now:

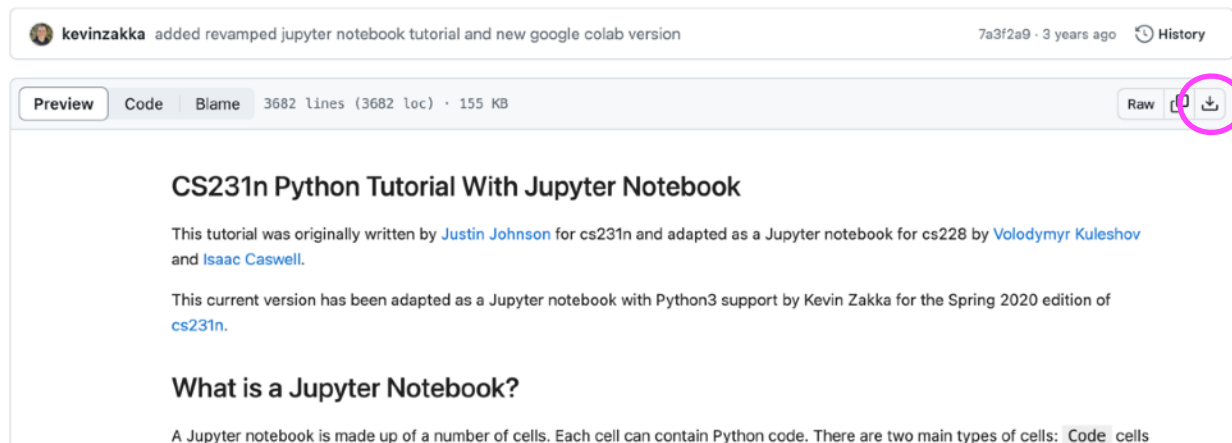


Basic Downloading, then Opening a Jupyter Notebook:

5. Downloading a Notebook

Go ahead and navigate to the following webpage: <https://github.com/cs231n/cs231n.github.io/blob/master/jupyter-notebook-tutorial.ipynb>

Once there you'll see this. Download the file using the **button circled**:



The file extension for these files is **'.ipynb'**

Lastly, move this file from your download folder into the directory (folder) you just created.

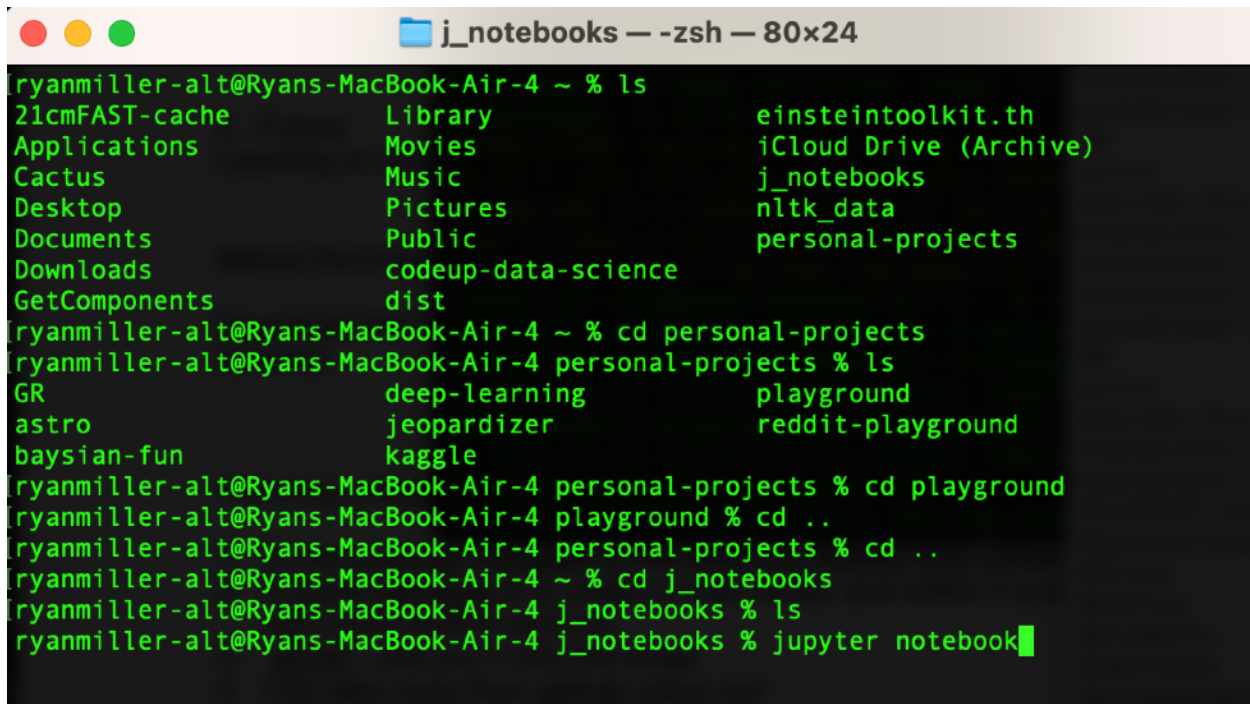
Note: This is the most straightforward, non-technical way for downloading a Jupyter Notebook (there are other ways, particularly utilizing Git and Github, but those can be discussed later and better in person/Zoom anyway).

6. Navigating to the Jupyter Notebook client:

The first thing to do is open a terminal window and navigate to the directory (folder) that you created in step 4. A trick here is to use the **[tab]** key to autocomplete as you move down the file tree to the folder you created. You do this using the 'cd' command in the terminal, then typing in the first letter and pressing tab which will then either autocomplete the folder/file you are looking for, or if there are multiple files/folders with the same initial letter it will show you what the options are while allowing you to add more letters directly to that same terminal line until you get the unique folder or file you are looking for. For example, after typing 'cd P' and hitting the tab key from my top level directory:

```
ryanmiller-alt — -zsh — 80x24
ryanmiller-alt@Ryans-MacBook-Air-4 ~ % ls
21cmFAST-cache      GetComponents        codeup-data-science
Applications         Library              dist
Cactus              Movies               einsteintoolkit.th
Desktop              Music                iCloud Drive (Archive)
Documents            Pictures              nltk_data
Downloads            Public               personal-projects
ryanmiller-alt@Ryans-MacBook-Air-4 ~ % cd P
Pictures/  Public/
```

Navigating to the folder just created looks something like the picture below, although I do navigate around the tree to show the key commands such as 'cd' to move directories; 'ls' is another helpful command that lists the files and folders for quick reference if need be. Also note how you can move back up the tree one level using "cd ..", that is a space and then two periods. In the last line I entered the command that starts the notebook "jupyter notebook" but haven't hit enter yet.

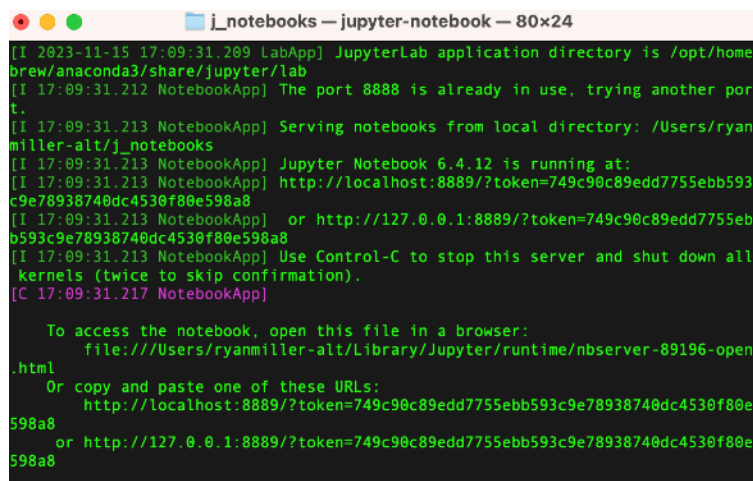


```
j_notebooks — -zsh — 80x24
ryanmiller-alt@Ryans-MacBook-Air-4 ~ % ls
21cmFAST-cache      Library              einsteintoolkit.th
Applications         Movies              iCloud Drive (Archive)
Cactus              Music               j_notebooks
Desktop             Pictures            nltk_data
Documents           Public              personal-projects
Downloads           codeup-data-science
GetComponents       dist
ryanmiller-alt@Ryans-MacBook-Air-4 ~ % cd personal-projects
ryanmiller-alt@Ryans-MacBook-Air-4 personal-projects % ls
GR                  deep-learning        playground
astro              jeopardizer          reddit-playground
baysian-fun        kaggle
ryanmiller-alt@Ryans-MacBook-Air-4 personal-projects % cd playground
ryanmiller-alt@Ryans-MacBook-Air-4 playground % cd ..
ryanmiller-alt@Ryans-MacBook-Air-4 personal-projects % cd ..
ryanmiller-alt@Ryans-MacBook-Air-4 ~ % cd j_notebooks
ryanmiller-alt@Ryans-MacBook-Air-4 j_notebooks % ls
ryanmiller-alt@Ryans-MacBook-Air-4 j_notebooks % jupyter notebook
```

Note that as you move through the file tree the directory you are in is shown to the immediate left of the %.

7. Running the notebook client:

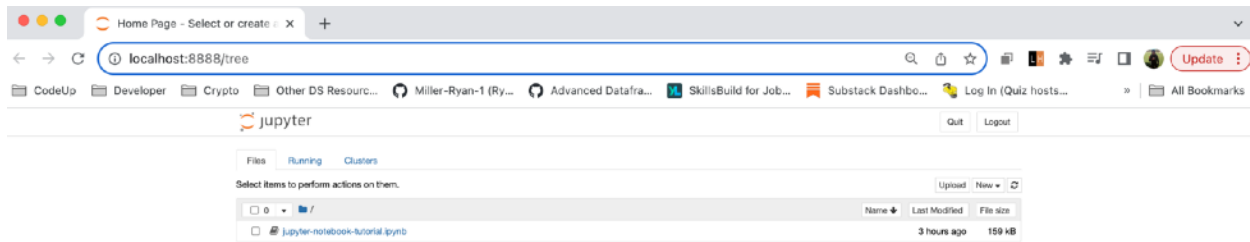
Once you have the terminal looking like that last line above, press enter. A browser window will appear, but first the terminal, which should disappear behind the web browser, is now where the jupyter notebook's kernel runs (its little brain) so that window **needs to be kept open and is unusable for terminal tasks** - if you need another terminal, simply open another one (using Shell>New Terminal). The terminal will now look like this:



```
j_notebooks — jupyter-notebook — 80x24
[I 2023-11-15 17:09:31.209 LabApp] JupyterLab application directory is /opt/homebrew/anaconda3/share/jupyter/lab
[I 17:09:31.212 NotebookApp] The port 8888 is already in use, trying another port.
[I 17:09:31.213 NotebookApp] Serving notebooks from local directory: /Users/ryanmiller-alt/j_notebooks
[I 17:09:31.213 NotebookApp] Jupyter Notebook 6.4.12 is running at:
[I 17:09:31.213 NotebookApp] http://localhost:8889/?token=749c90c89edd7755ebb593c9e78938740dc4530f80e598a8
[I 17:09:31.213 NotebookApp] or http://127.0.0.1:8889/?token=749c90c89edd7755ebb593c9e78938740dc4530f80e598a8
[I 17:09:31.213 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 17:09:31.217 NotebookApp]

To access the notebook, open this file in a browser:
file:///Users/ryanmiller-alt/Library/Jupyter/runtime/nbsrvr-89196-open.html
Or copy and paste one of these URLs:
http://localhost:8889/?token=749c90c89edd7755ebb593c9e78938740dc4530f80e598a8
or http://127.0.0.1:8889/?token=749c90c89edd7755ebb593c9e78938740dc4530f80e598a8
```

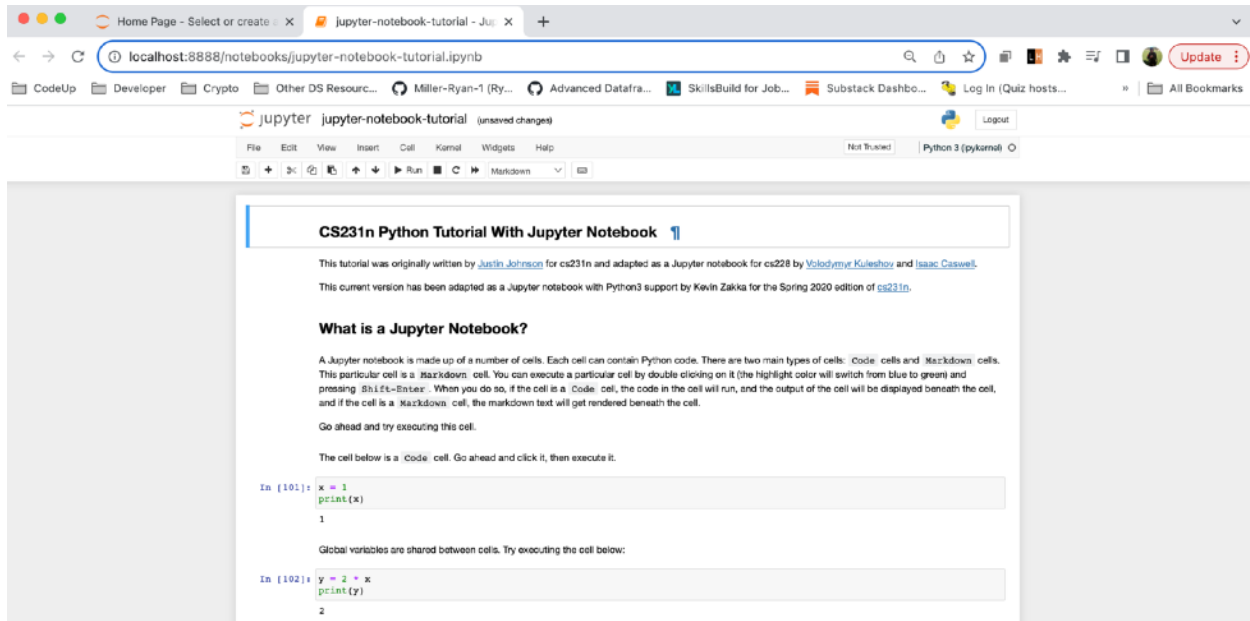
And the newly opened browser should look like this:



It is hard to see, but the file in there is the one we downloaded and moved.

8. Running a Notebook:

To run a file, simply select it in the window and click on it. It will open another browser window that will be the actual notebook.



The file is a tutorial that you can play around with yourself! The easiest way to learn Python scripting is finding tutorial files written in these notebooks and working through them.

9. Creating a new notebook:

In the first browser page created when launching the jupyter notebooks, you can also create a new notebook by clicking on 'New' in the upper right of the file list and select "Python 3 (ipykernel)". This will launch a blank notebook.

Using Jupyter Notebooks - Basics:

10. Start with the Tutorial

The tutorial we downloaded will help with Python and getting used to the Notebook as a tool. I would say the most important things to know from a overall notebook perspective are the following shortcut keys:

- Esc - Transitions from cell entry mode to cell manipulation mode. Basically in cell entry mode anything you type goes into the cell; in cell manipulation mode, you can use the following commands to change, move, delete and add cells:
 - 'b' - adds a new empty cell below the one currently selected
 - 'dd' - deletes the cell currently selected
 - 'y' - converts a cell to a code cell
 - 'm' - converts cell to markdown
 - '1','2','3','4','5' - Creates a markdown cel with heading style text (1 = biggest, 5 = smallest).

The notebook automatically saves after pretty much each calculation (you should still periodically save and make sure the terminal running the kernel is always open and not throwing errors).

Another protip - sometimes Python returns a value in a notebook, sometimes it does not. There are rules to this which I can't recall but you get a feel for it after awhile. What you can always do though it add in a "print(var)" statement with whatever you want visibility on as the variable. Helps debugging and troubleshooting immensely.

In order to reset the cell, you can either re-run it, or if you have multiple cells depending on one another you can go ahead and use Kernel>Restart & Clear Output if you have to do a larger reset. Just know that all variables will be removed and all cells will have to recalculate from scratch if time is a consideration.