

- [Installing our software stack](#)

- [Julia](#)
 - [Juno](#)
 - [Jupyter](#)
 - [Sample Jupyter Notebook](#)

Installing our software stack

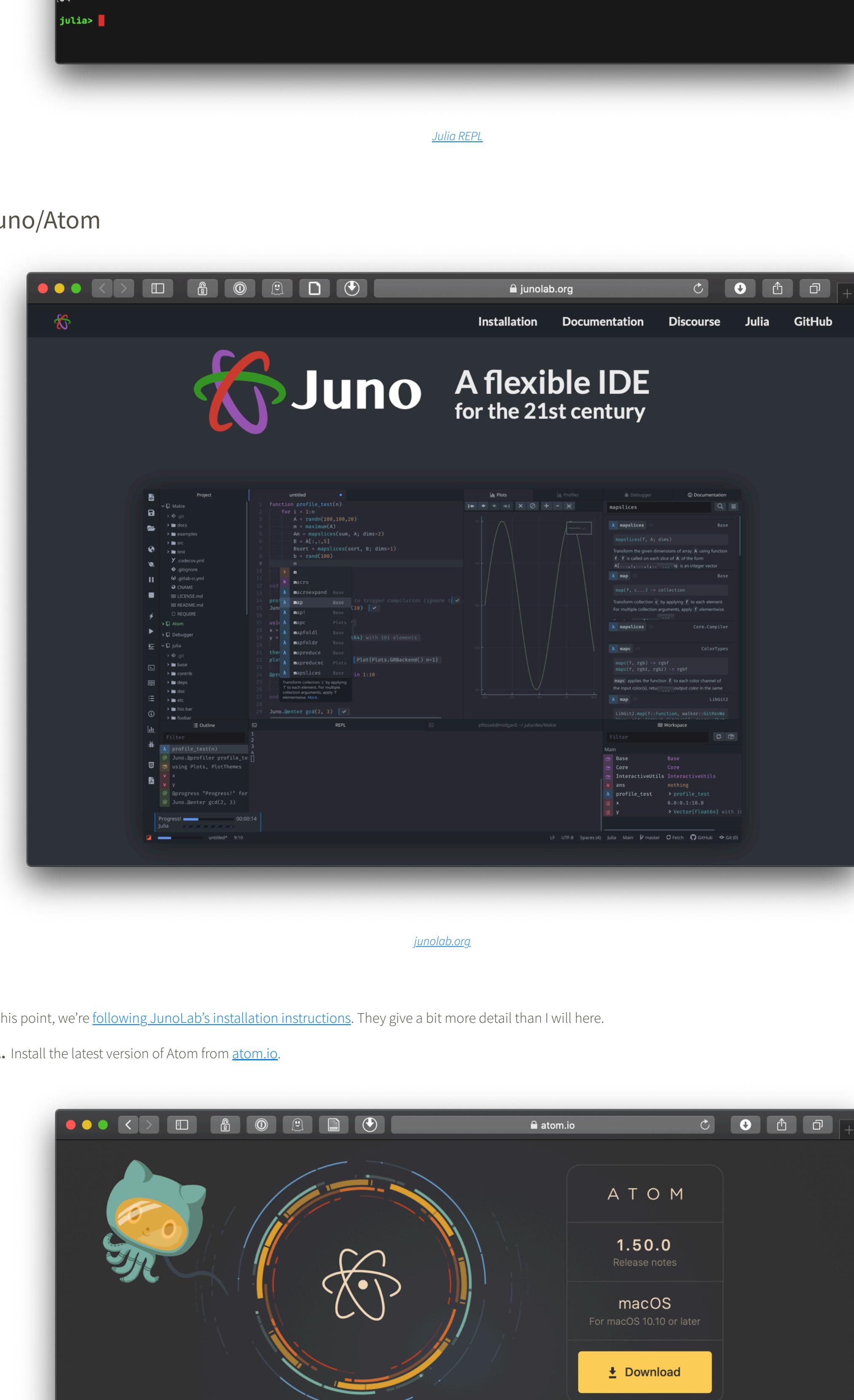
If you have issues, or see where I can improve this guide, do let me know.

Our software stack consists of four main components:

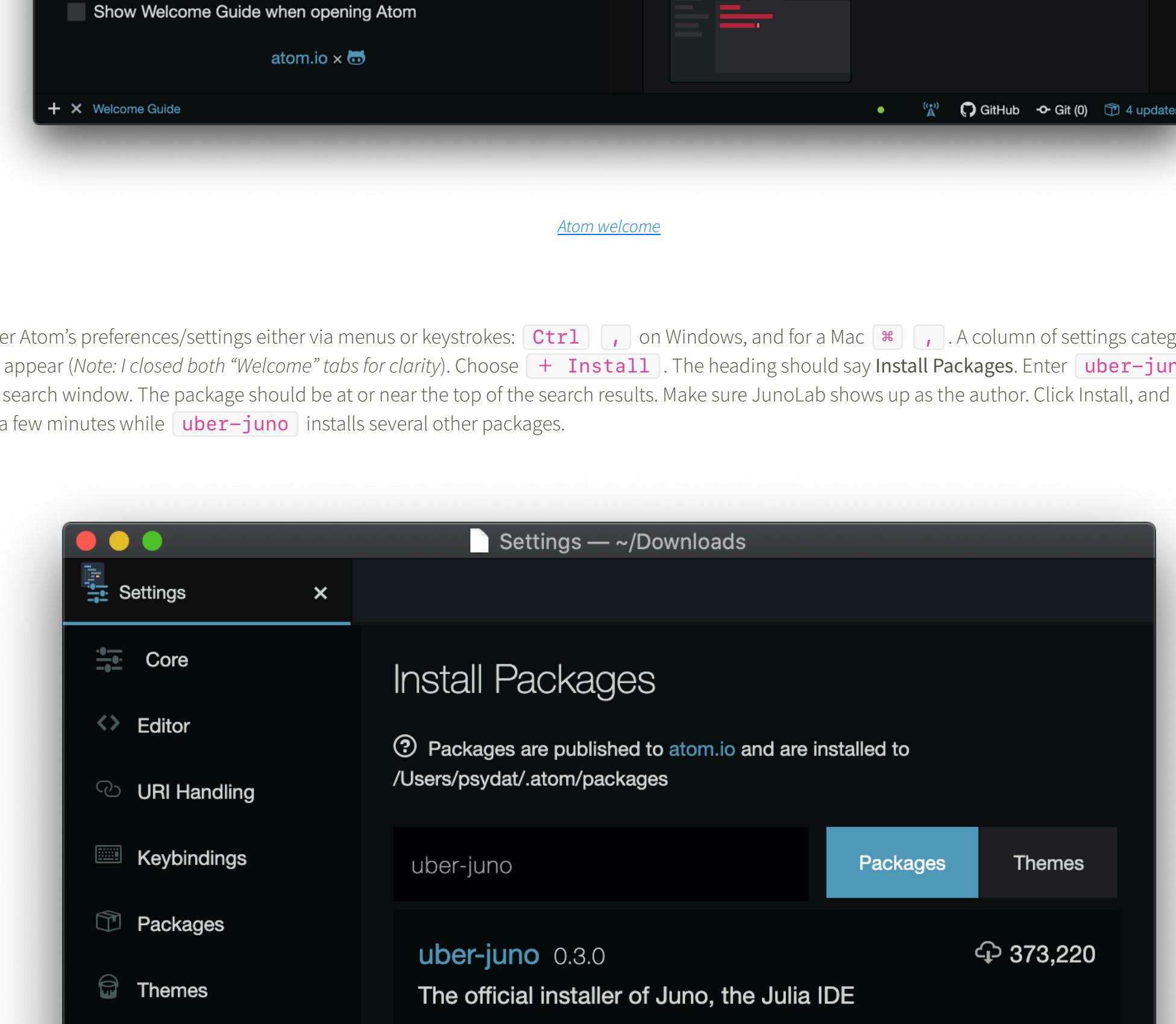
1. [Julia](#), our language of choice. Julia is new open-source language, designed specifically for technical and scientific computing.
2. [Atom](#), a highly extensible open-source text editor. Atom's capabilities are added via a large ecosystem of packages.
3. [Juno](#), a set of Atom packages that full Julia IDE (integrated development environment) for the desktop.
4. [Jupyter](#), a web-based IDE for literate programming. Jupyter combines code, rich text, interactive widgets, and more, all in the same document. There is an example of a Jupyter notebook at the end of this guide.

After installation, you will have a stable platform for developing code in almost any programming language. I chose these components so that the software can grow with you. Starting out, you will have Julia and Python, and then add R (statistical computing) later in the semester. A nice side-effect is that you'll also be equipped for any kind of writing (code, prose, research paper, etc.)

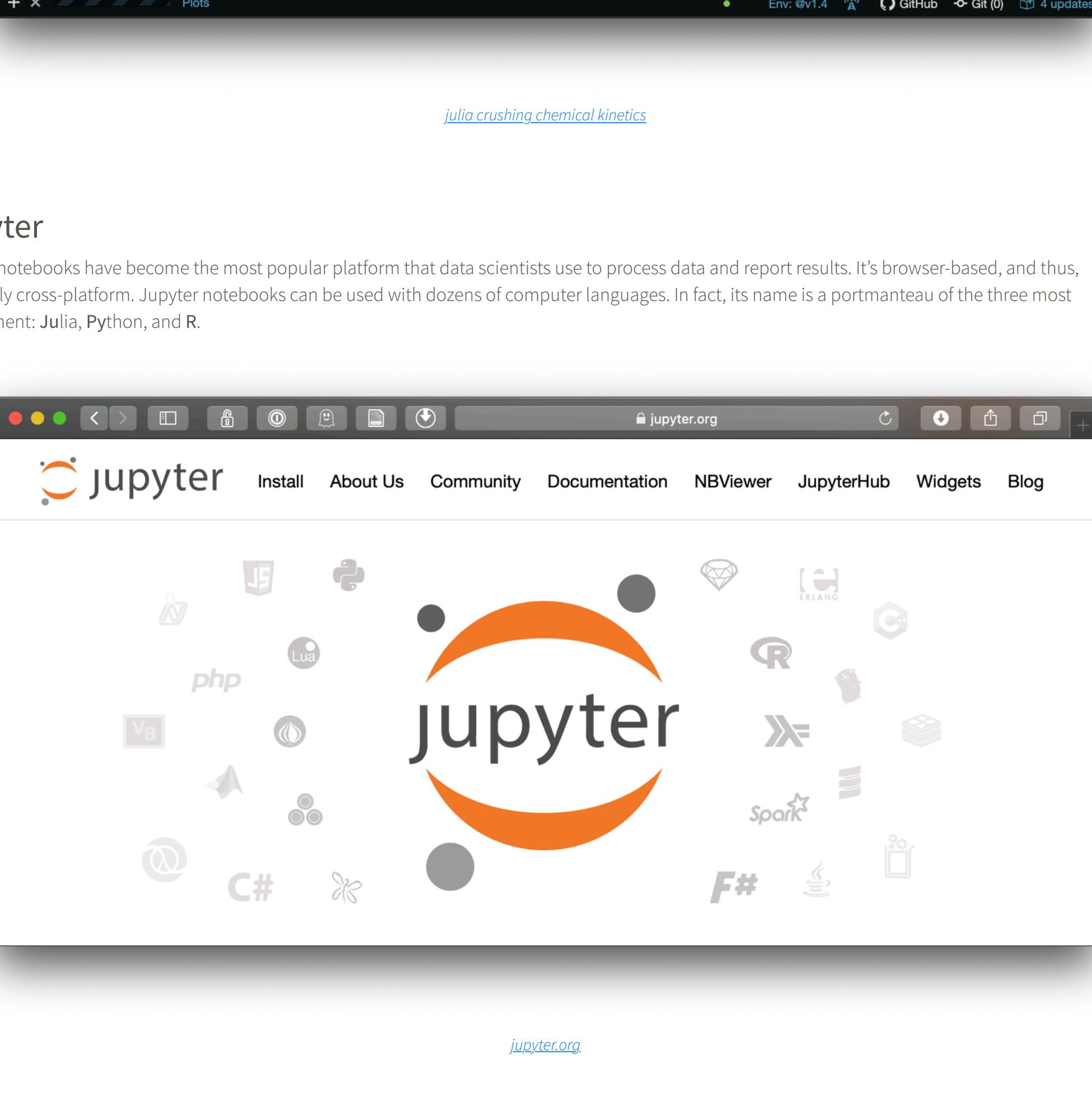
Julia



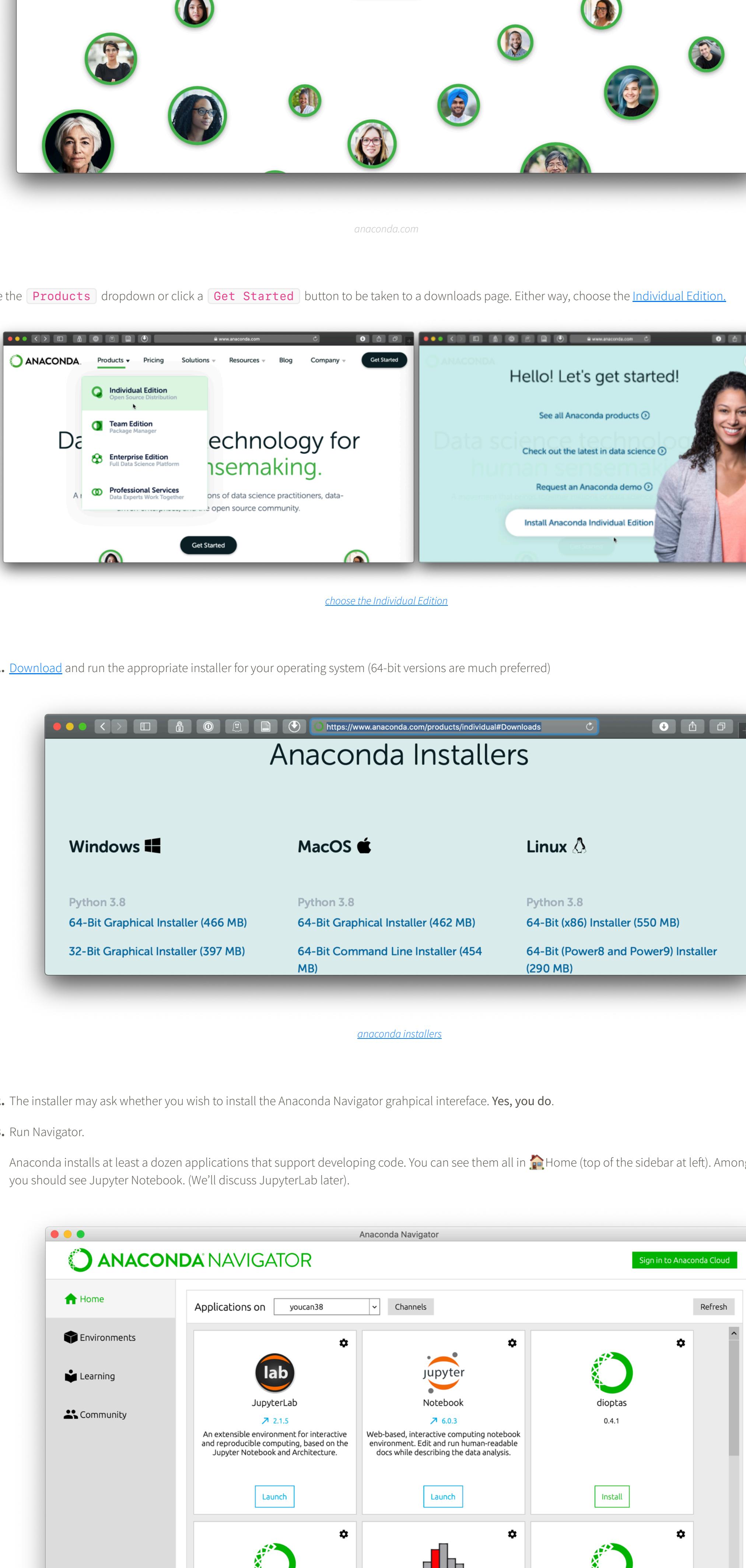
1. [Download](#) and run the appropriate installer for your operating system (64-bit versions are much preferred)



2. If you're feeling impatient, you can run the application from your start menu or command line: [julia](#). You'll be presented with the Julia REPL (read-evaluate-print loop). The REPL is interactive, and we'll do maintenance and housekeeping in it, but most work will take place in an IDE (integrated development environment). That's where Atom, Juno, and Jupyter come in.

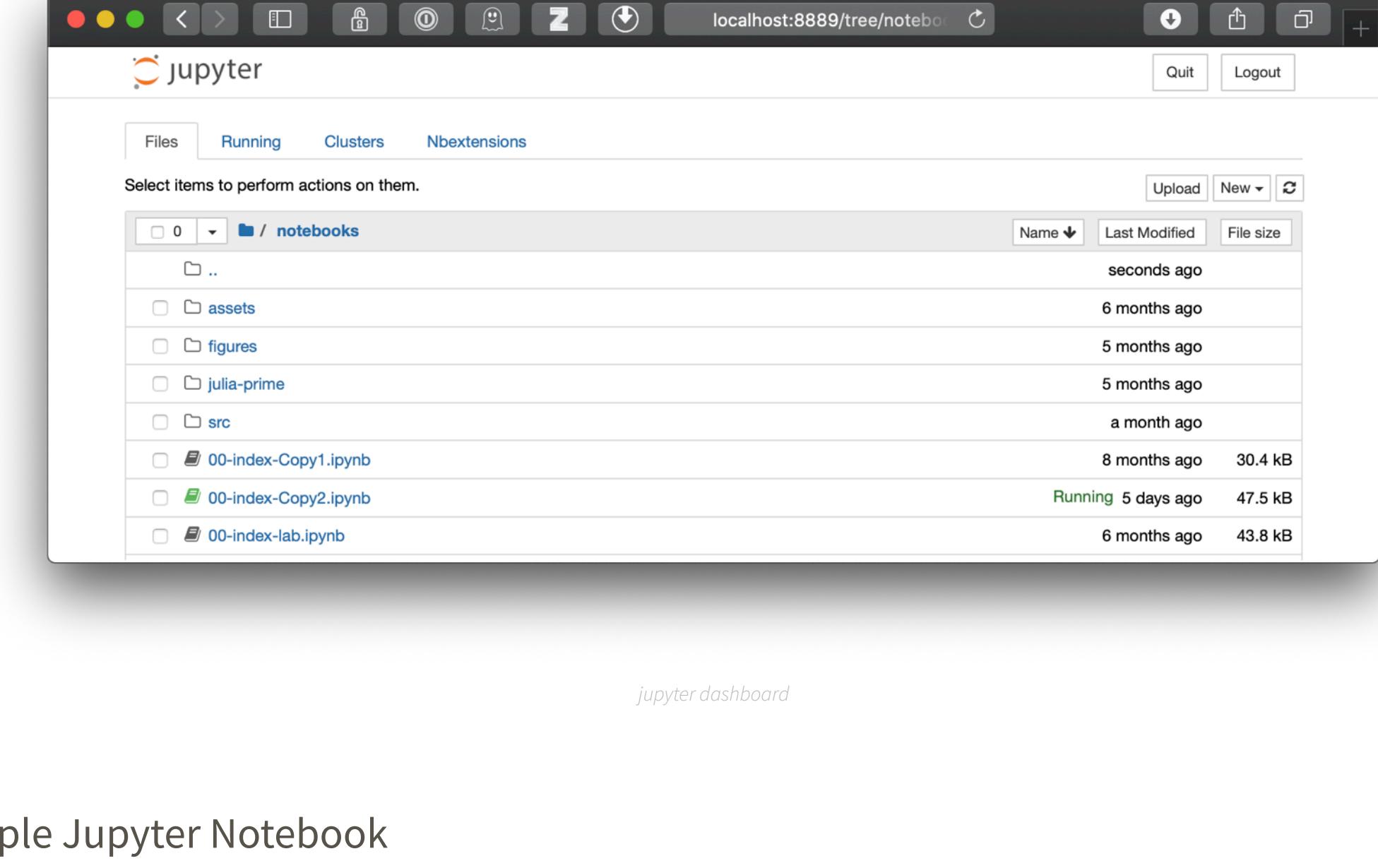


Juno/Atom

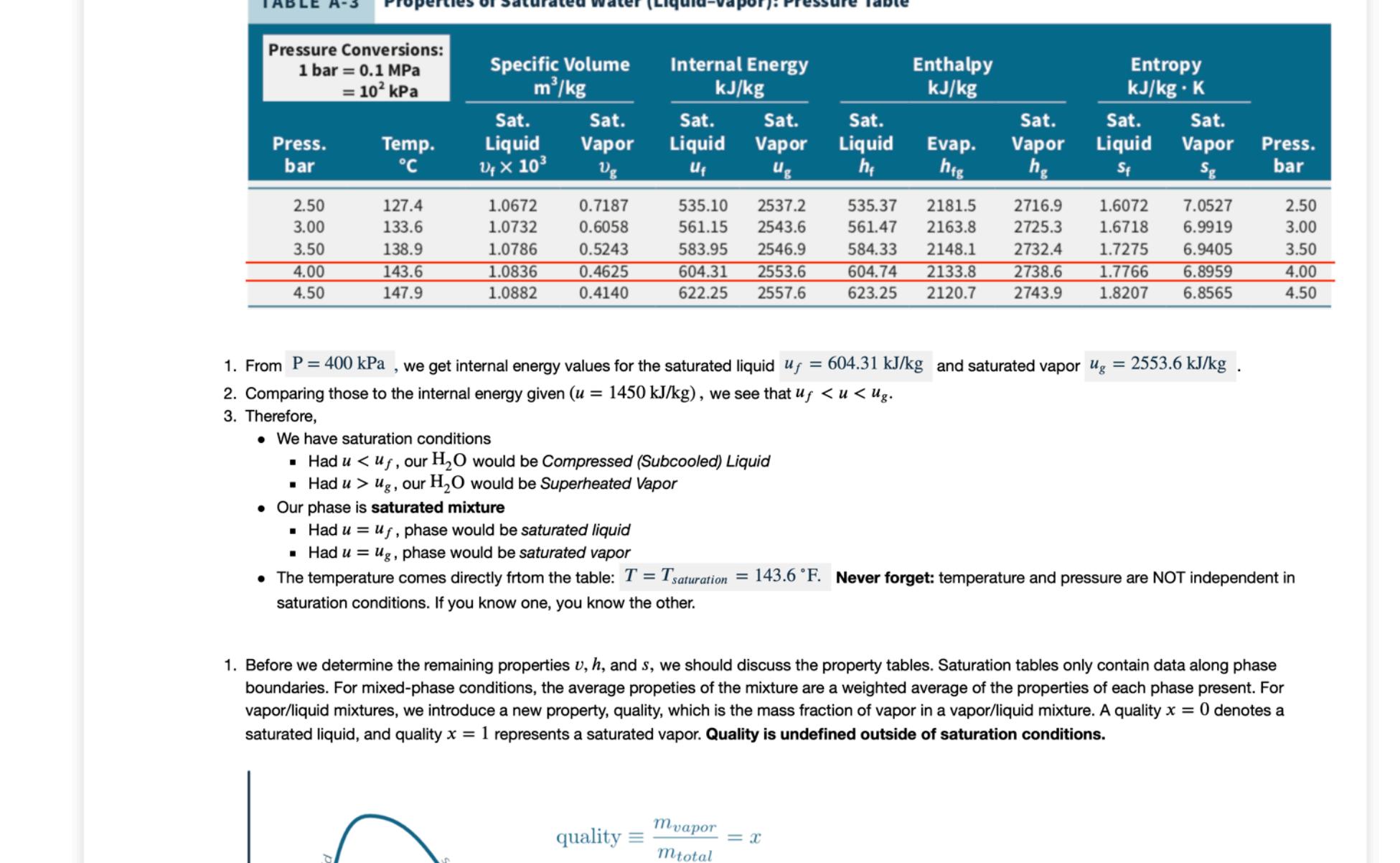


- At this point, we're [following JunoLab's installation instructions](#). They give a bit more detail than I will here.

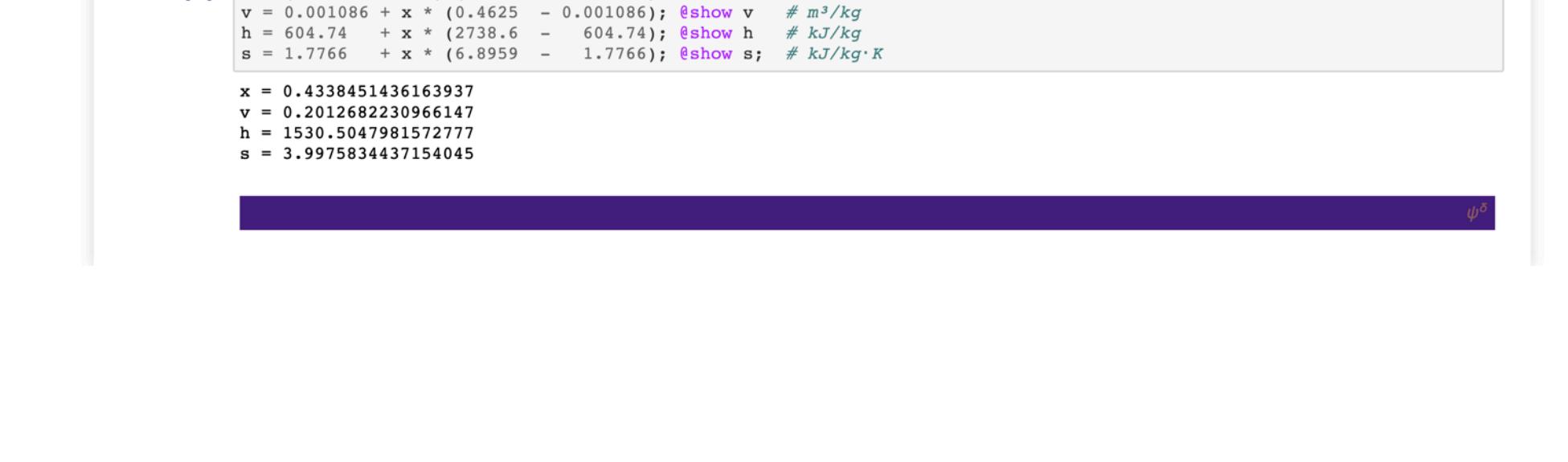
1. Install the latest version of Atom from [atom.io](#).



2. Run Atom. You'll greeted with a welcome screen.



3. Enter Atom's preferences/settings either via menus or keystrokes: [Ctrl + ,](#) on Windows, and for a Mac [⌘ + ,](#). A column of settings categories will appear (Note: I closed both "Welcome" tabs for clarity). Choose [+ Install](#). The heading should say [Install Packages](#). Enter [uber-juno](#). The search results make sure JunoLab shows up as the author. Click [Install](#), and relax for a few minutes while [uber-juno](#) installs several other packages.



4. When the installation finishes, pat yourself on the back: you're all set to work with Julia in a desktop platform. We have just one more task to complete.

2. The installer may ask whether you wish to install the Anaconda Navigator graphical interface. Yes, you do.

3. Run Navigator. Anaconda installs at least a dozen applications that support developing code. You can see them all in [Home](#) (top of the sidebar at left). Among them, you should see Jupyter Notebook. (We'll discuss JupyterLab later).

4. Launching Jupyter Notebook opens Jupyter Dashboard in a browser window. You'll see a directory listing of their current folder, usually your home folder. From there, navigate to the folder where your notebooks are stored and get to work.

Sample Jupyter Notebook

- 1.

2. After we determine the relevant properties, a , b , and c , we should discuss the property tables. Saturation tables only contain data along phase boundaries. For mixed-phase conditions, the average properties of the mixture are a weighted average of the properties of each phase present. For vapor/liquid mixtures, we introduce a new property, quality, which is the mass fraction of vapor in a vapor/liquid mixture. A quality $x = 0$ denotes a saturated liquid, and quality $x = 1$ denotes a saturated vapor. Quality is determined outside of saturation conditions.

3. To determine the relevant properties, a , b , and c , we first calculate x , then use it to calculate the other properties. Remember, get values from the saturation tables at $P = 400 \text{ kPa}$.

4. Launching Jupyter Notebook opens Jupyter Dashboard in a browser window. You'll see a directory listing of their current folder, usually your home folder. From there, navigate to the folder where your notebooks are stored and get to work.

Sample Jupyter Notebook

- 1.

2. After we determine the relevant properties, a , b , and c , we should discuss the property tables. Saturation tables only contain data along phase boundaries. For mixed-phase conditions, the average properties of the mixture are a weighted average of the properties of each phase present. For vapor/liquid mixtures, we introduce a new property, quality, which is the mass fraction of vapor in a vapor/liquid mixture. A quality $x = 0$ denotes a saturated liquid, and quality $x = 1$ denotes a saturated vapor. Quality is determined outside of saturation conditions.

3. To determine the relevant properties, a , b , and c , we first calculate x , then use it to calculate the other properties. Remember, get values from the saturation tables at $P = 400 \text{ kPa}$.

4. Launching Jupyter Notebook opens Jupyter Dashboard in a browser window. You'll see a directory listing of their current folder, usually your home folder. From there, navigate to the folder where your notebooks are stored and get to work.

