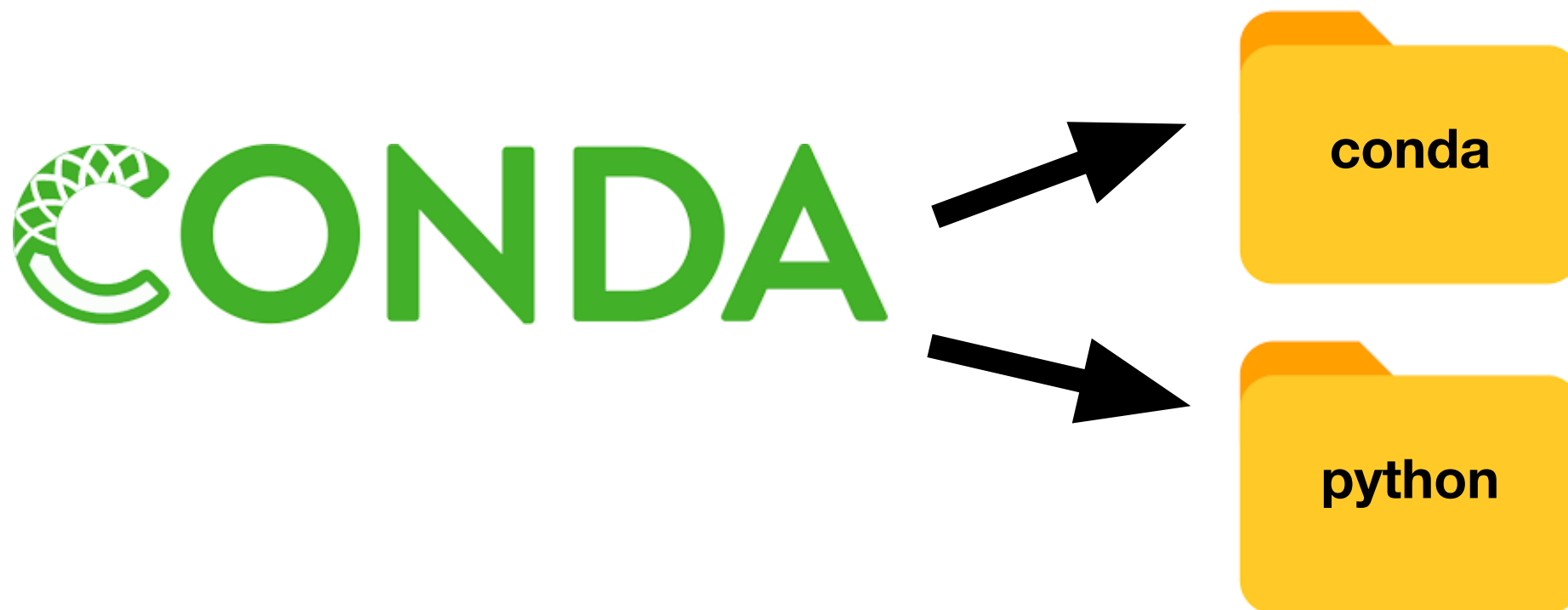


Introduction to Python for Scientific Programming

Step 1: Installing Python tools on your machine

We are going to use a package manager called miniconda
<https://conda.io/miniconda.html>

Go ahead and download the appropriate miniconda installer on to your machine



Step 2: Install Jupyter Notebooks

For this we are going to use the conda installation that we just performed

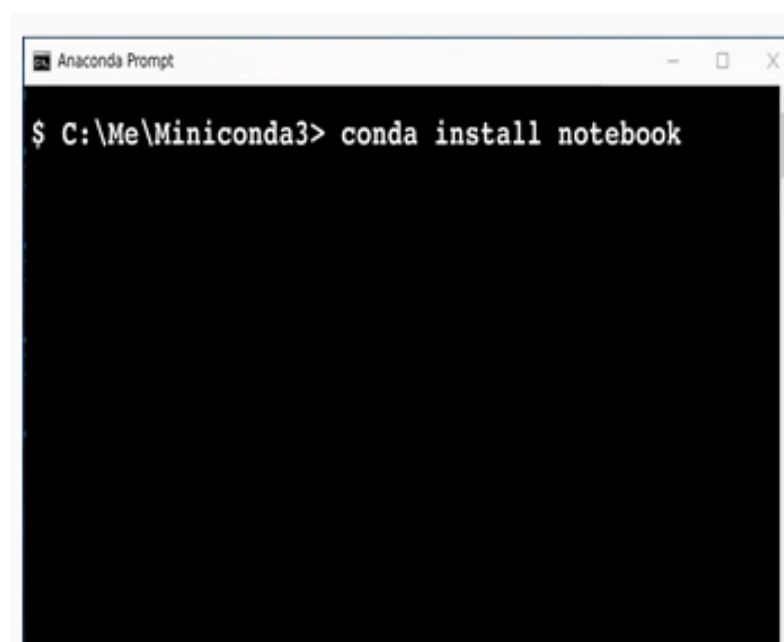
We will need to open up a command line interface (CLI) for this

If you have a mac, open the Terminal app; If you have a windows box you will use the Anaconda prompt that was just installed by miniconda

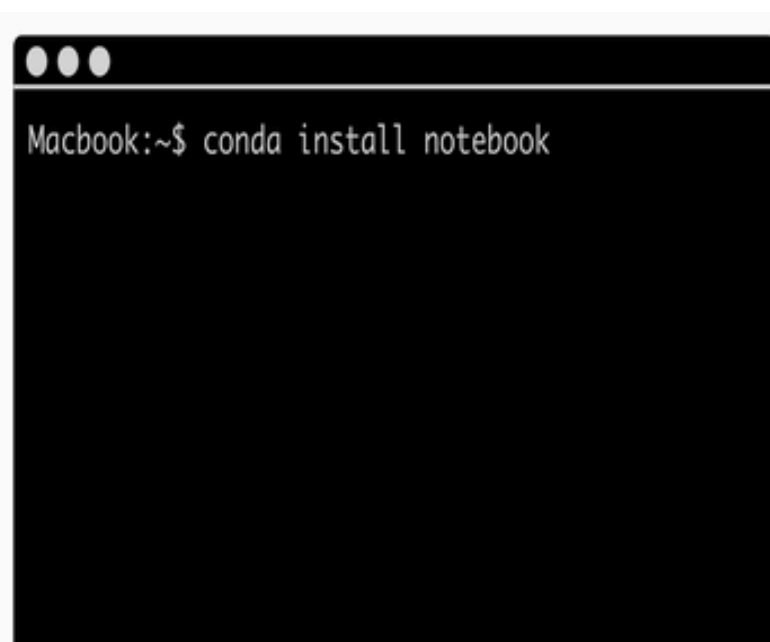
For a windows app use the command line prompt (search for cmd or use desktop shortcut to conda)

```
`conda install -c conda-forge jupyterlab`
```

Windows



Mac



Step 3: Install more libraries

Again we are going to use conda for this. We let the package manager do all the hard work for us and it will just give us the libraries that we need

The basic call is ``conda install some_package``

Here is the list of packages that we want for now:

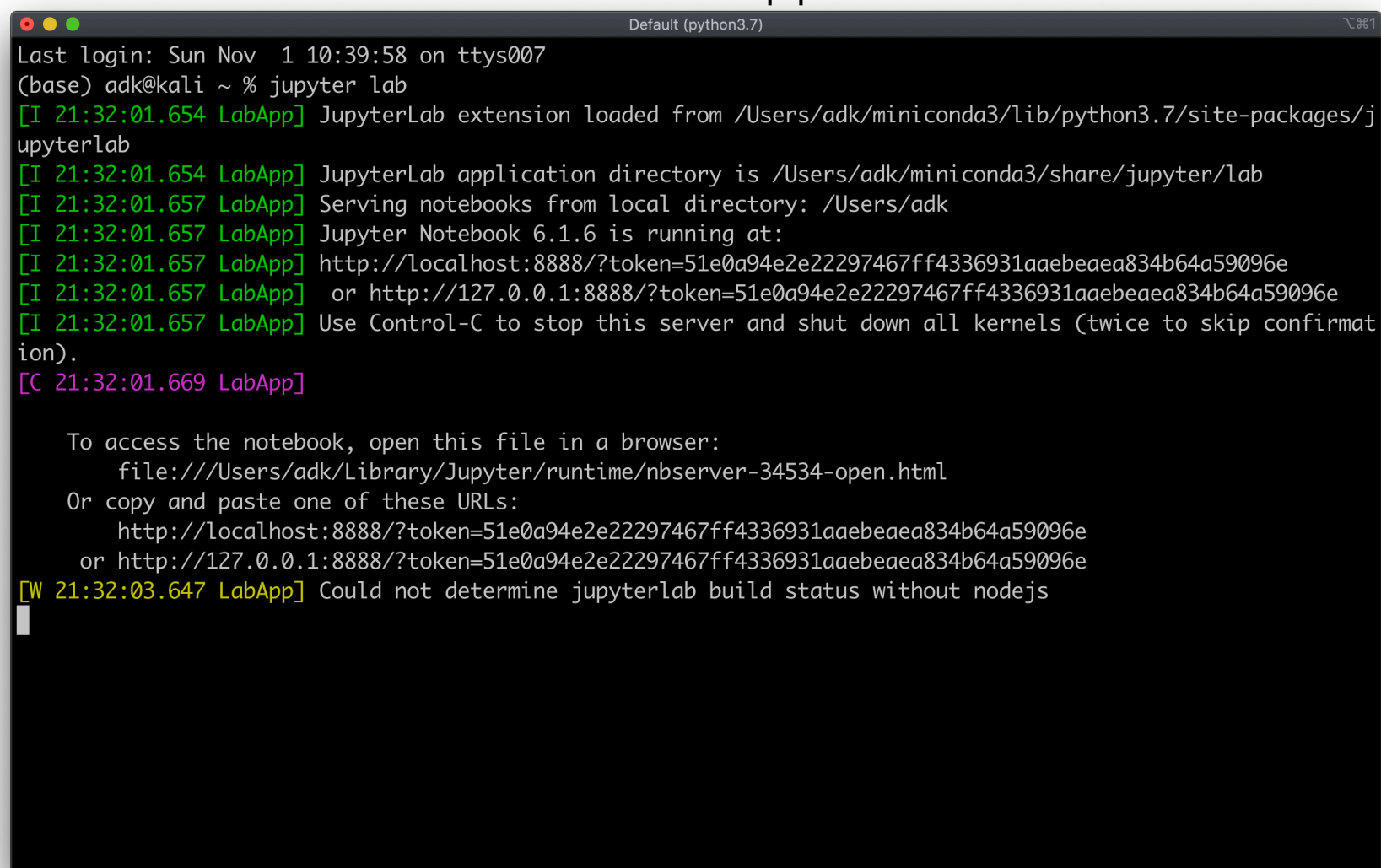
- numpy
- scipy
- matplotlib

Go ahead and install all three of those using ``conda install`` now

Step 4: Start a Jupyter lab server

Working from the CLI still, type `jupyter lab`

That will bring up a bit of text in your command like so, and a browser window should appear

A terminal window titled 'Default (python3.7)' showing the output of the 'jupyter lab' command. The window has standard macOS window controls (red, yellow, green buttons) in the top-left corner. The text in the terminal is as follows:

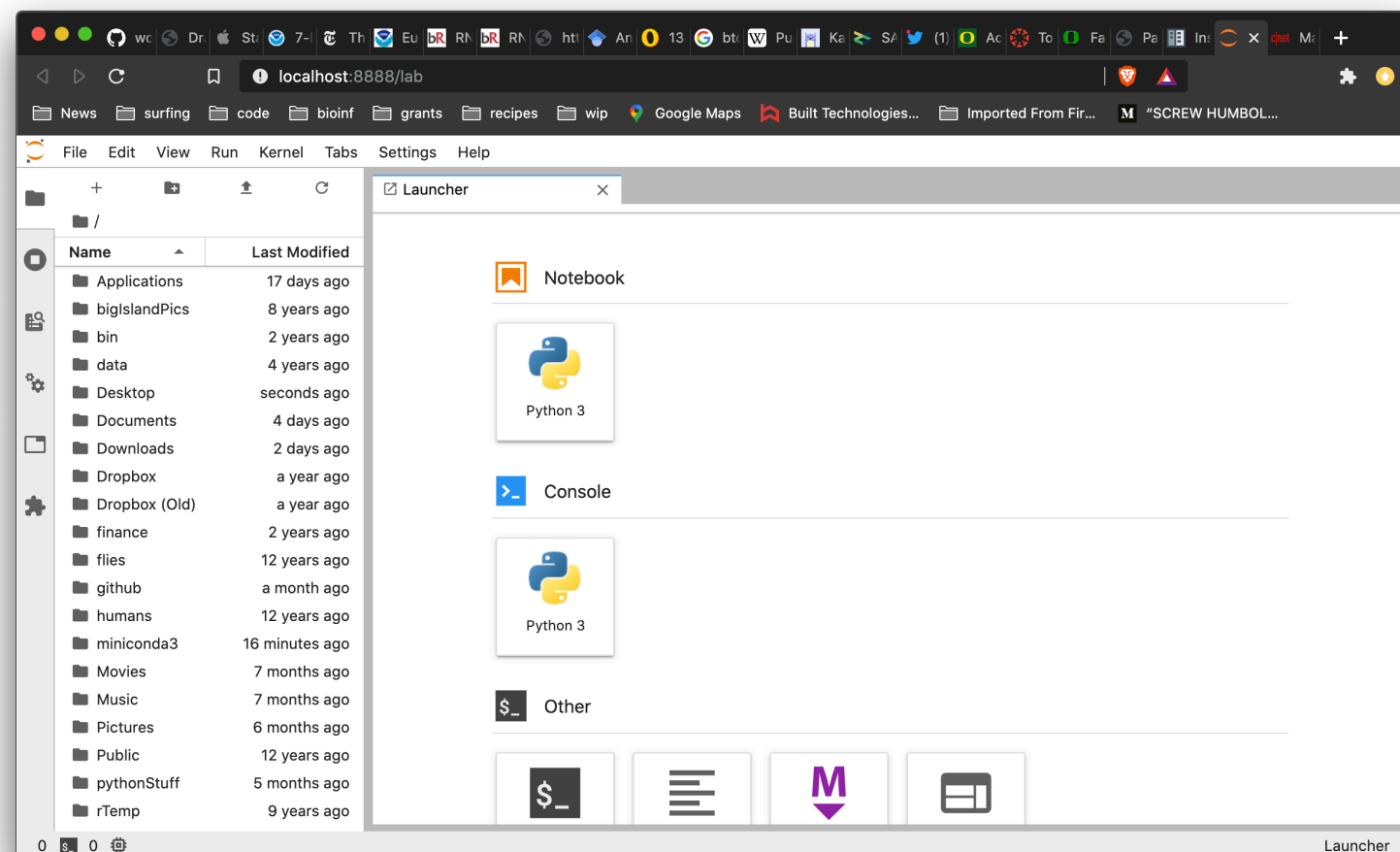
```
Last login: Sun Nov  1 10:39:58 on ttys007
(base) adk@kali ~ % jupyter lab
[I 21:32:01.654 LabApp] JupyterLab extension loaded from /Users/adk/miniconda3/lib/python3.7/site-packages/jupyterlab
[I 21:32:01.654 LabApp] JupyterLab application directory is /Users/adk/miniconda3/share/jupyter/lab
[I 21:32:01.657 LabApp] Serving notebooks from local directory: /Users/adk
[I 21:32:01.657 LabApp] Jupyter Notebook 6.1.6 is running at:
[I 21:32:01.657 LabApp] http://localhost:8888/?token=51e0a94e2e22297467ff4336931aaebeaea834b64a59096e
[I 21:32:01.657 LabApp] or http://127.0.0.1:8888/?token=51e0a94e2e22297467ff4336931aaebeaea834b64a59096e
[I 21:32:01.657 LabApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 21:32:01.669 LabApp]

To access the notebook, open this file in a browser:
    file:///Users/adk/Library/Jupyter/runtime/nbserver-34534-open.html
Or copy and paste one of these URLs:
    http://localhost:8888/?token=51e0a94e2e22297467ff4336931aaebeaea834b64a59096e
    or http://127.0.0.1:8888/?token=51e0a94e2e22297467ff4336931aaebeaea834b64a59096e
[W 21:32:03.647 LabApp] Could not determine jupyterlab build status without nodejs
```

Step 4: Start a Jupyter lab server

Working from the CLI still, type `jupyter lab`

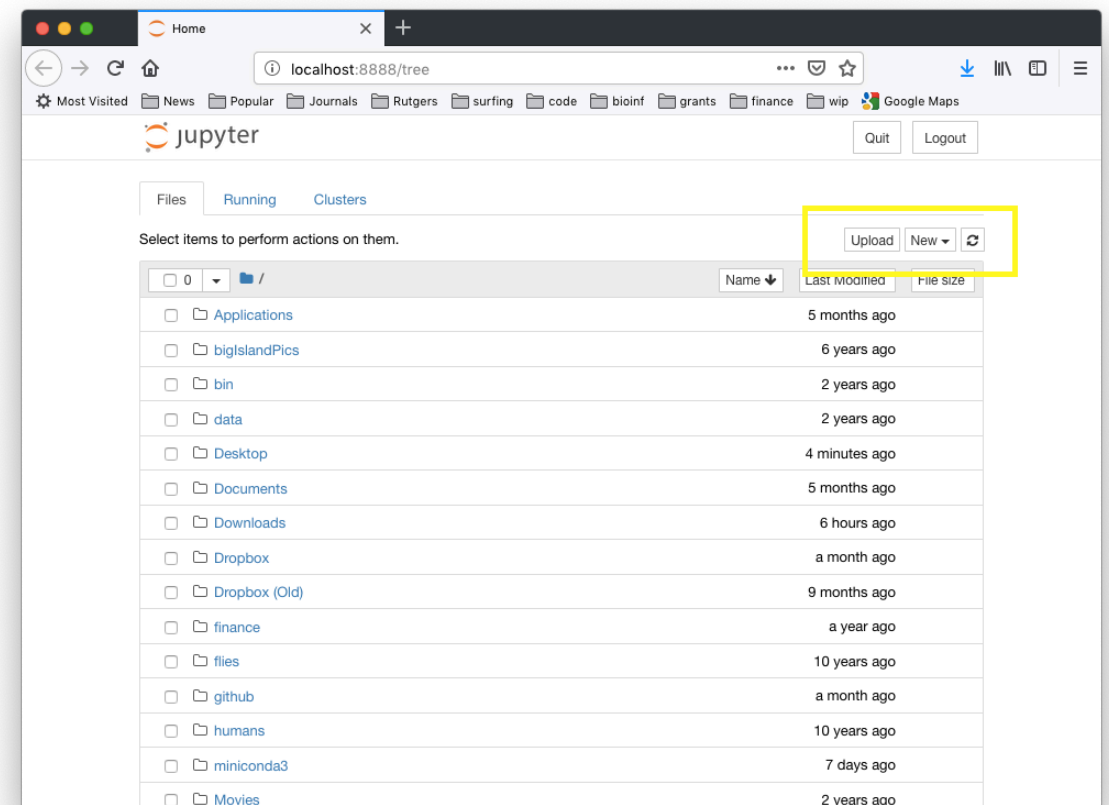
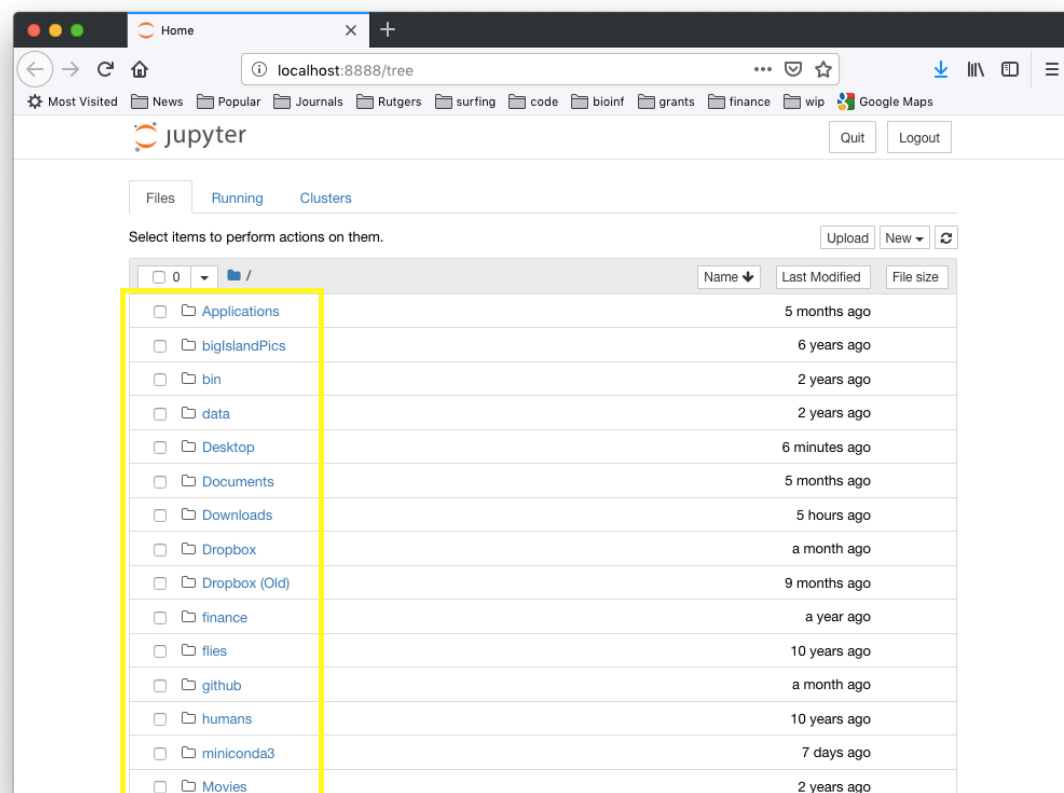
That will bring up a bit of text in your command like so, and a browser window should appear



Step 5: Pat yourself on the back

Python and the associated tools we need are installed. Nice.

Now let's get familiar with the jupyter lab and Jupyter notebooks a bit

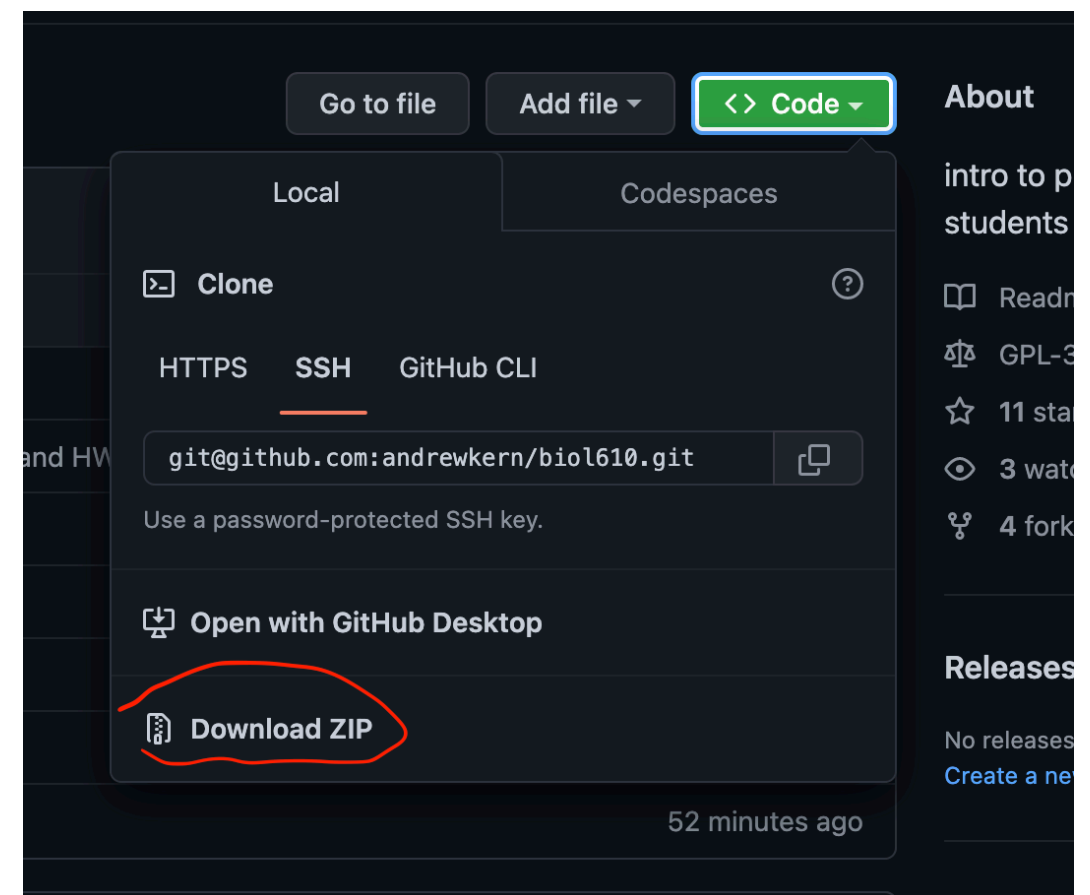


Step 6: Download class materials

This class will be delivered all through Jupyter notebooks that I've put on GitHub for them to be freely available.

Navigate a browser to
<https://github.com/andrewkern/biol610>

Easiest way to get the class materials— download zip

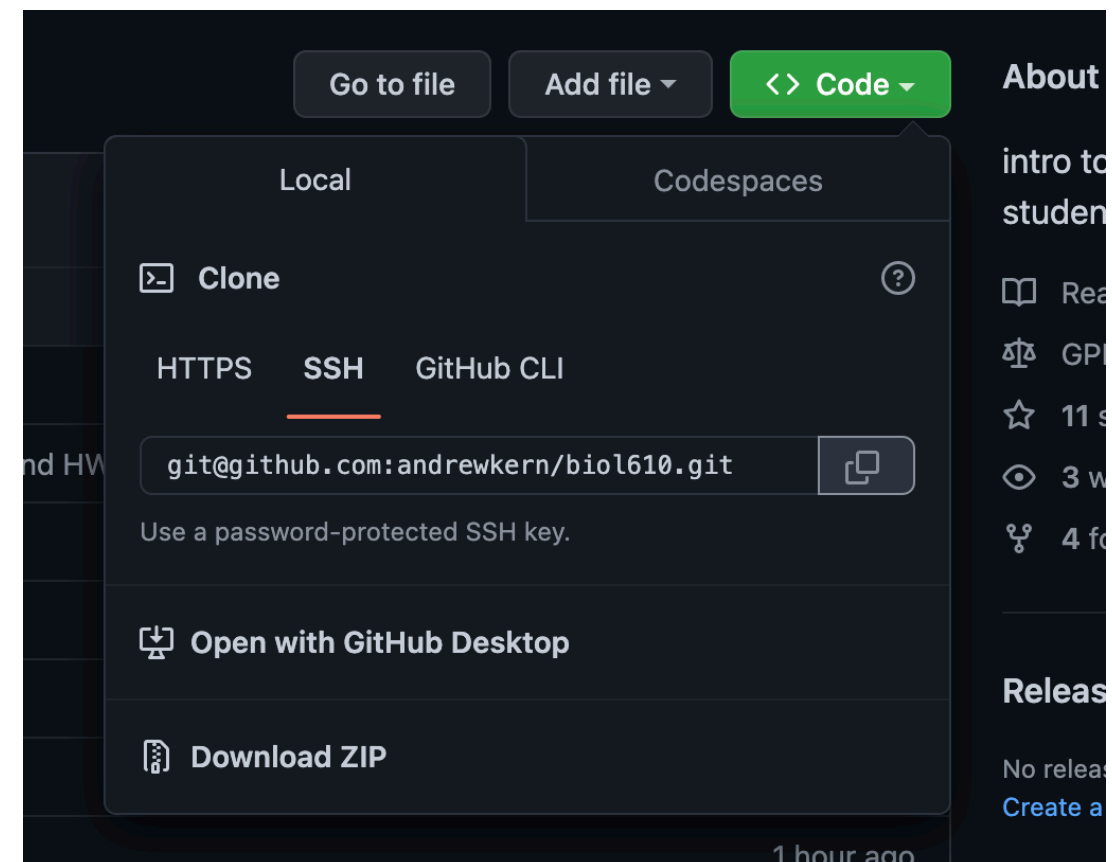


Step 6: Download class materials

This class will be delivered all through Jupyter notebooks that I've put on GitHub for them to be freely available.

Navigate a browser to
<https://github.com/andrewkern/biol610>

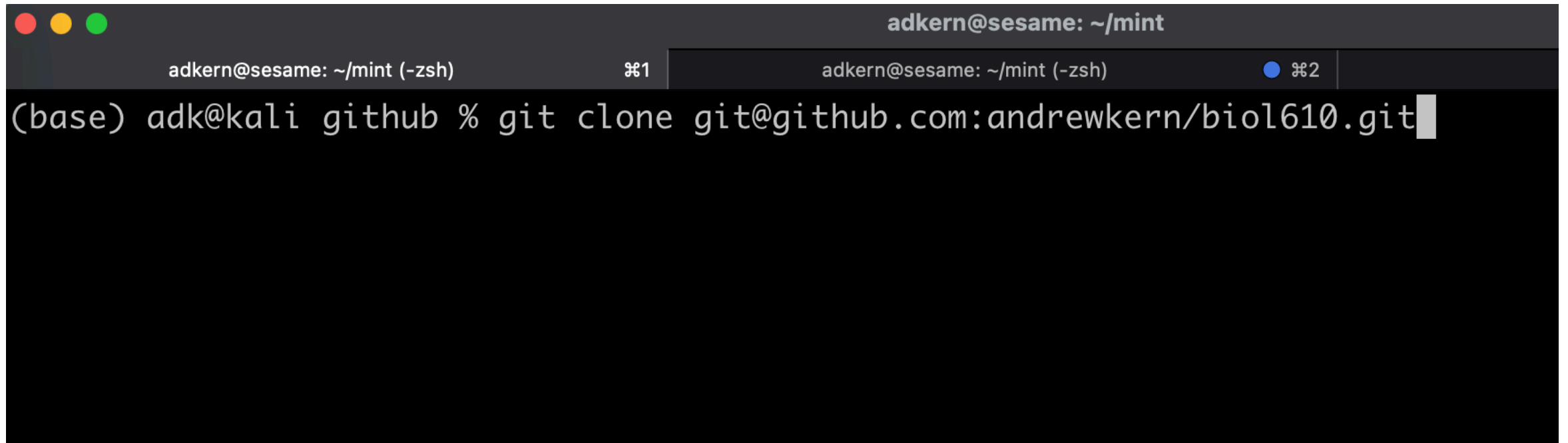
Or.... use git to “clone the repo”



Step 6: Download class materials

This class will be delivered all through Jupyter notebooks that I've put on GitHub for them to be freely available.

Navigate a browser to
<https://github.com/andrewkern/biol610>

A terminal window with a dark background and light gray text. The window has a title bar with three colored circles (red, yellow, green) on the left and the text 'adkern@sesame: ~/mint' on the right. Below the title bar, there are two tabs. The first tab is labeled 'adkern@sesame: ~/mint (-zsh)' and has a small icon to its right. The second tab is labeled 'adkern@sesame: ~/mint (-zsh)' and has a small blue circle icon to its right. The main area of the terminal shows the command '(base) adk@kali github % git clone git@github.com:andrewkern/biol610.git' with a cursor at the end of the command.

```
adkern@sesame: ~/mint
adkern@sesame: ~/mint (-zsh)
adkern@sesame: ~/mint (-zsh)
(base) adk@kali github % git clone git@github.com:andrewkern/biol610.git
```

This will make a copy of the materials that is linked to mine on the internet (GitHub)

Step 7: Use Jupyter and find lecture 1 notebook

The screenshot displays the JupyterLab interface. On the left, the file browser shows the directory structure: `github / notebooks /`. A list of files is shown with columns for Name and Last Modified. The file `lecture1.ipynb` is selected. On the right, the notebook editor shows the content of `lecture1.ipynb`. The notebook has a title `Python baby steps` and a description: "In this notebook we are going to cover some of the very first steps to using python in the jupyter notebook framework. Let's start with a traditional programming first step-- the hallowed 'Hello, world!'". The first code cell contains the code `print("Hello, world!")`, which has been executed, resulting in the output `Hello, world!`. The notebook also includes a section on `Variables`, explaining that a variable is a name used to denote something or a value, and that in python, variables can be declared and values can be assigned to it. An example is given: "For instance let's declare a variable named `x` and set it equal to 2". The second code cell contains the code `x = 2`, which has been executed, resulting in no output.

File browser details:

Name	Last Modified
hello.png	3 years ago
Homework...	3 years ago
lecture1.ip...	a year ago
lecture10.i...	a year ago
lecture2.ip...	2 years ago
lecture3.ip...	a year ago
lecture4.ip...	a year ago
lecture5.ip...	2 years ago
lecture6.ip...	2 years ago
lecture7.ip...	a year ago
lecture8.ip...	a year ago
lecture9.ip...	a year ago
Inf	2 years ago
Is_orchid.f...	3 years ago
lysozymeS...	3 years ago
lysozymeS...	3 years ago
lysozymeS...	3 years ago
mlc	2 years ago
my_examp...	4 years ago
my_examp...	4 years ago
my_examp...	4 years ago

Notebook content:

Python baby steps

In this notebook we are going to cover some of the very first steps to using python in the jupyter notebook framework. Let's start with a traditional programming first step-- the hallowed "Hello, world!"

```
[1]: print("Hello, world!")
```

Hello, world!

here we are using the built-in function `print` that forms one of the most basic tools in python. We will use `print` for making our programs write to the screen. `print` takes as input what we call a string (more later) and then prints that string as output.

Congrats-- this is your first bit of python coding.

Variables

A name that is used to denote something or a value is called a variable. In python, variables can be declared and values can be assigned to it as follows,

For instance let's declare a variable named `x` and set it equal to 2

```
[2]: x = 2
```

notice there was no output associated with setting the value of `x`. To check if everything went okay we can `print` it