Center for Advanced Computing

www.cac.cornell.edu

Starting soon: Introduction to JupyterLab for Python

CAC Services for Cornell researchers:

- Grant proposal preparation.
- Technical consulting and training, including software development, database development, data management, cloud computing, and containerization.
- Computational support, including cloud computing, cluster management, and archival storage services.

Learn more at www.cac.cornell.edu

Center for Advanced Computing

Introduction to JupyterLab for Python

Christopher Cameron
Computational Scientist
Cornell University Center for Advanced Computing
https://cac.cornell.edu/Cameron/
cjc73@cornell.edu

Workshop Questions

1. What is JupyterLab?
2. What is it good for?
    1. Features that make Jupyterlab useful for researchers
    2. Where to get more information

Jupyter/Python take time to learn, and this is the first step. The initial learning curve looks very steep when you don’t have the requisite background knowledge, but it is surmountable.

- See our upcoming workshops on Python and the command line (Linux)

What is JupyterLab?

- Browser-based interface for interactive data science, scientific computing, scripting, and programming.

- Jupyter *notebooks* are the main type of document
  - Similar to an Rmarkdown notebook in some respects

- Jupyter notebooks can be shared directly or exported to other formats
  - As software libraries (nbdev)
  - As books (jupyter-book)
  - As slides
  - As web apps (voilà or mybinder.org)
  - As static web pages (nbviewer)
  - As PDF documents

History of JupyterLab

- Inspired by the notion of a scientist’s or mathematician’s notebook.
- “Computational Notebook”
- Mathematica introduced interactive math notebooks in the late 1980’s
- **IPython Notebook** 2011
- **Jupyter Notebook** 2014 (supporting **Ju**lia, **Pyt**hon and **R** languages)
- **JupyterLab** first available in 2018
  - 2019 extensions supported
  - 2021-2002 added IDE features (debugger)
  - Future: collaborative editing?

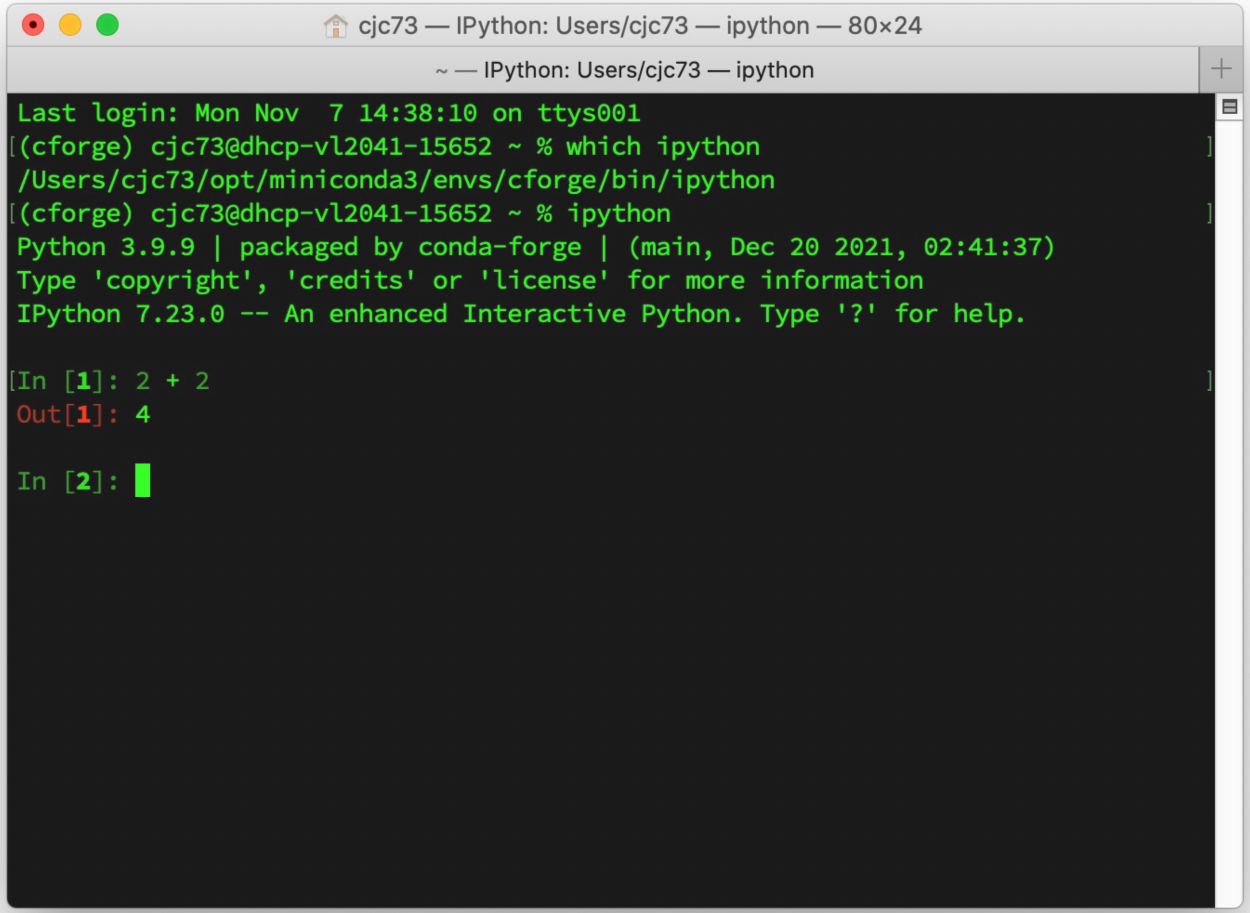
JupyterLab Development

- JupyterLab is one of major projects developed and maintained by Project Jupyter.
  - Open-source, free and runs on Linux, Windows and MacOS.
  - https://jupyter.org/
  - Supported by NumFOCUS, which also supports many other Python data science and scientific computing efforts.
  - Other projects include JupyterHub, a multi-user version of the JupyterLab server and various language kernels for JupyterLab.

Python, R and Julia

- Trio of modern open-source computer languages favored by data scientists.
  - JupyterLab stands for the **Ju**lia, **Pyt**hon, and **R** languages
- Python and R have significant overlap and similarity, but
  - Python is more general
  - Python tends to be favored for deep learning
  - R and Python are both popular in machine learning
  - R tends to be favored for statistical analysis
  - Both have huge communities and many add-on packages
- Julia is general purpose language designed at MIT with numerical computing in mind.
  - Only recently reached version 1.0
  - Designed to be more performant but it is still developing
  - Small ecosystem compared to R and Python (but can use R and Python)
  - Keep an eye on it!

Python Interpreter


cjc73 — IPython: Users/cjc73 — ipython — 80×24
~ — IPython: Users/cjc73 — ipython

```
Last login: Mon Nov  7 14:38:10 on ttys001
(cforge) cjc73@dhcp-vl2041-15652 ~ % which ipython
/Users/cjc73/opt/miniconda3/envs/cforge/bin/ipython
(cforge) cjc73@dhcp-vl2041-15652 ~ % ipython
Python 3.9.9 | packaged by conda-forge | (main, Dec 20 2021, 02:41:37)
Type 'copyright', 'credits' or 'license' for more information
IPython 7.23.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: 2 + 2
Out[1]: 4

In [2]:
```

Productive for interactive use, hard to document and share

JupyterLab with Python is…

Good for Scientists and Communicators:

- unstructured data
- file system scripting
- data scraping, cleaning and formatting
- data visualization
- Interface with custom code in C/C++, Fortran
- tabular data with Pandas
- statistical analysis with stats-models and scikit-learn (but can have weird defaults)
- Interface with ML and deep learning libraries

Consider other tools for:

- Writing large software libraries (but see nbdev)
- Backend web development

Parallelization is not straightforward unless the library you are using does it automatically

Why JupyterLab instead of:

- IDLE - usually distributed with Python, an interpreter and editor with basic functionality. Not a notebook.

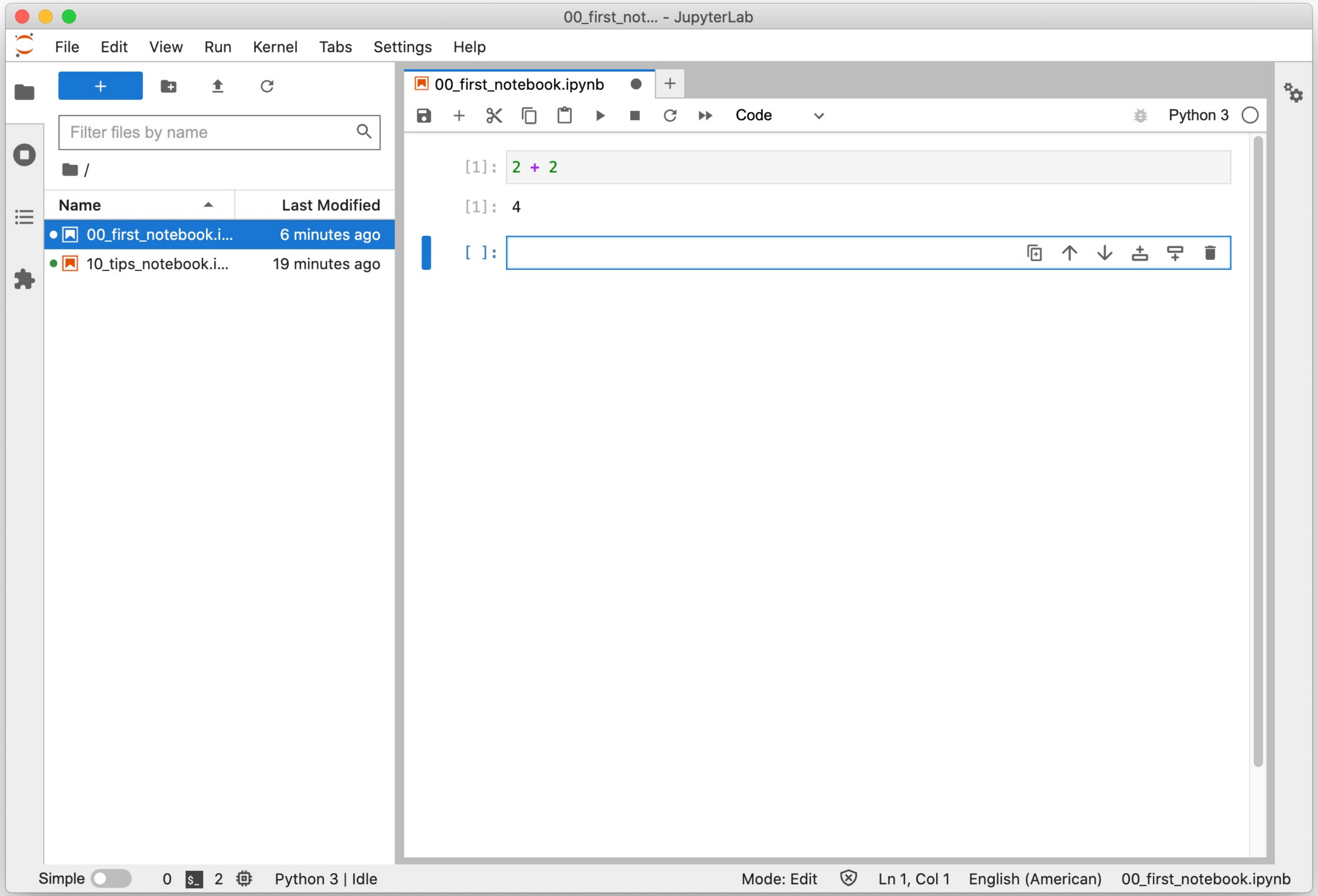
- Jupyter Notebook - Jupyter Notebook evolved into Jupyter Lab. The transition is still underway but Notebook will go away. Time to move!

- Microsoft VS Code - Integrated Development Environment with many add-ons to support Python. Even has an extension to support writing Jupyter notebook format. Imposes a software development model on projects.

- PyCharm - Powerful IDE with many features. Focused on software development, not data analysis or communication.

- Google Colab – modified Jupyter notebook on the cloud.

JupyterLab Interface



```
[1]: 2 + 2
[1]: 4
[ ]:
```

Overview: Installing JupyterLab (once)

- Download and install **conda**, an environment and package manager.
  - Package manager locates and installs requested software
    - Manages compatibility and dependencies among installed packages
  - Environments are independent collections of software.
    - You might create different environments for different projects or
    - Re-create a given environment on multiple computers.

- Use conda to create an environment for JupyterLab with Python and other packages.

Installing Conda

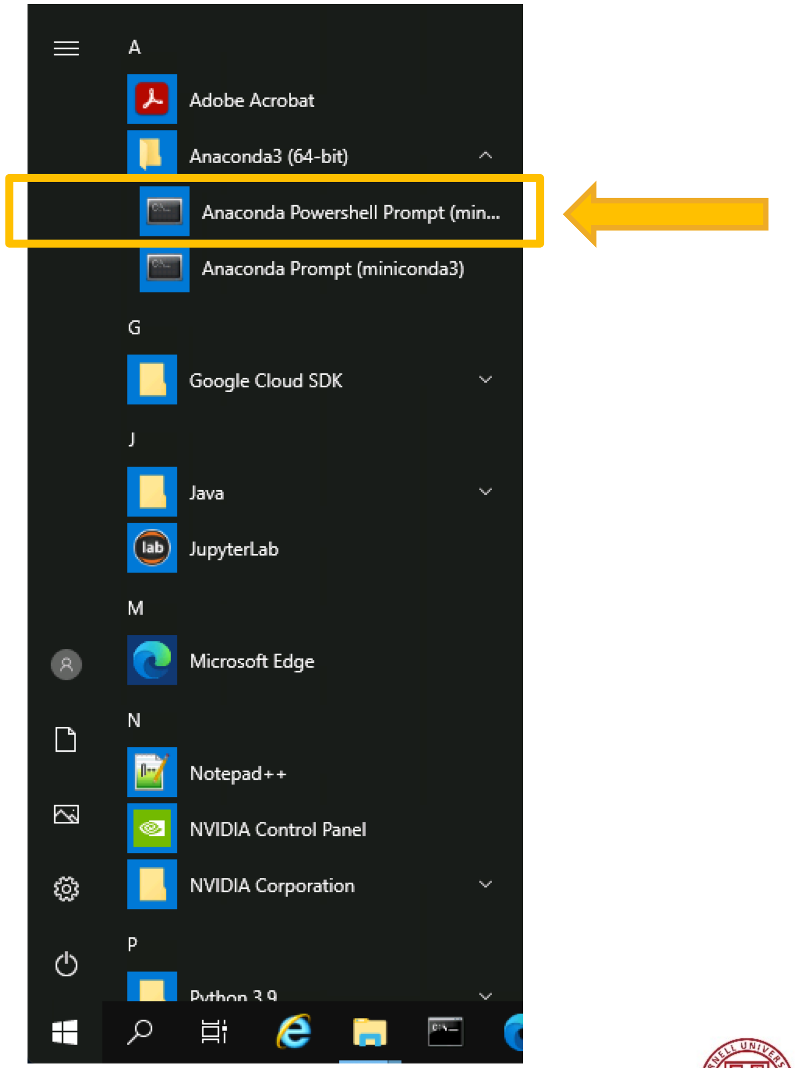
- Download miniconda installer from: https://docs.conda.io/en/latest/miniconda.html
  - Use personal install (default option)

- Alternatively, download Anaconda and get a large collection of packages and other programs to explore:
  - https://docs.conda.io/projects/conda/en/stable/user-guide/install/index.html

- Need help deciding?
  - https://docs.conda.io/projects/conda/en/stable/user-guide/install/download.html#anaconda-or-miniconda

Opening Anaconda Prompt - Windows

- After miniconda is installed, look for “Anaconda Powershell Prompt” in the start menu.
- This is the Windows equivalent of the MacOS/Linux Terminal for our purposes.



Installing JupyterLab

- miniconda creates a base environment for you and can even activate it automatically, **but**
  - Don’t use the base environment! Keep the base env lean.
- Make a new environment for each kind of work (I named it “scu”)
  - In Anaconda Powershell or MacOS Terminal (or iTerm)

```
conda create -n scu -c conda-forge jupyterlab=3 "ipykernel>=6" pandas
numpy scipy tqdm matplotlib plotnine nodejs ipywidgets jupytext
jupyterlab-spellchecker
```

  - Answer [Y]es when prompted and wait for install
  - After install completes, activate new environment (scu):

```
conda activate scu
```

Making R and Julia accessible via JupyterLab (optional)

- If R is installed (http://lib.stat.cmu.edu/R/CRAN/), you can install the IRkernel following the directions at https://irkernel.github.io/installation/

- If you install Julia (https://julialang.org/) you can then install the IJulia kernel following the directions at https://julialang.github.io/IJulia.jl/stable/manual/installation/

Note: Launching JupyterLab

- The method I am showing is one of many ways to launch JupyterLab
  - I like to compartmentalize my projects.
  - I might need to open more than one project at a time.
  - It works on my own machine and on remote machines.
  - I need to share some (but not all) projects with others (via git).
  - I prefer simple workflows

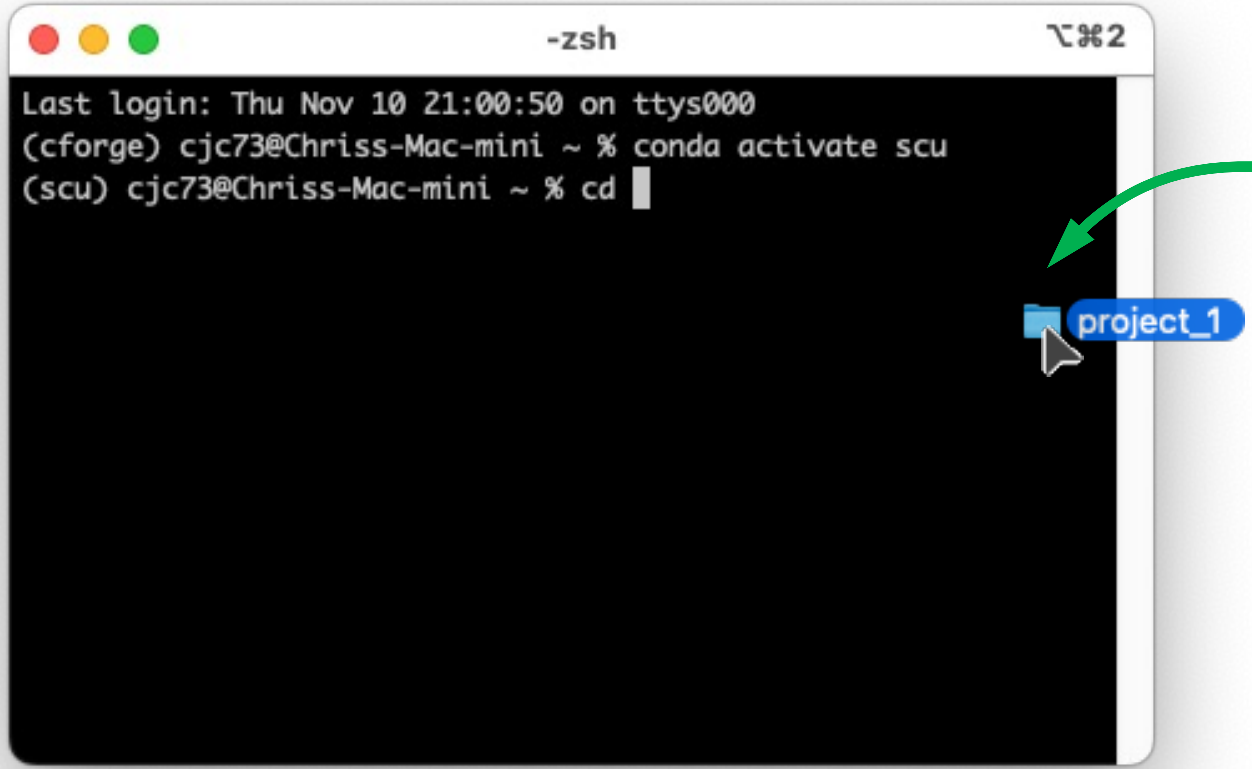
- Setup: create a directory for your project
  - Make a subfolder for *notebooks* and *data*
  - Avoid spaces in the folder names.

Overview: Launching JupyterLab (each time)

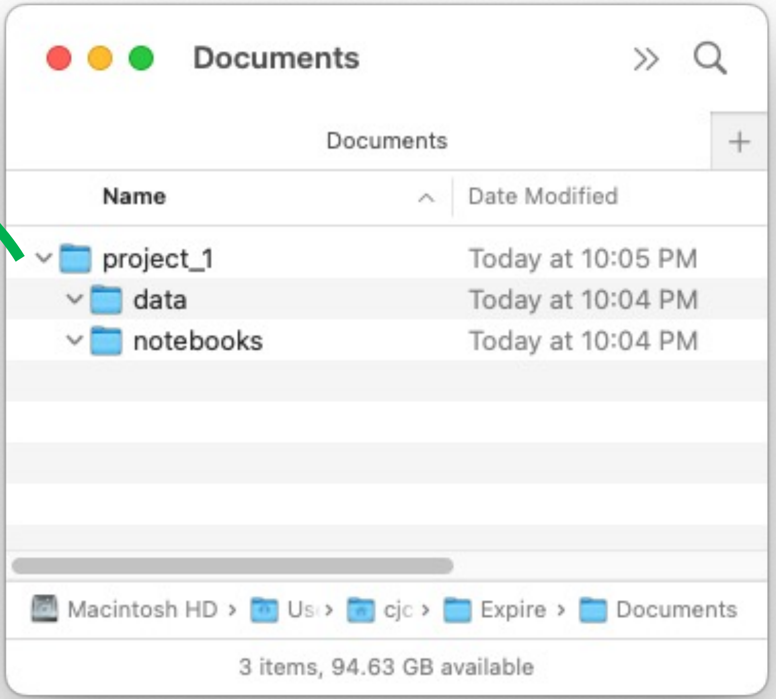
1. Open Terminal (MacOS/Linux) or Anaconda Powershell (Windows)
2. Activate the conda environment
   ```
   conda activate scu
   ```
3. Navigate to your project directory on command line
4. Launch JupyterLab
   ```
   jupyter-lab
   ```
5. Wait for your web browser to launch

(Launching on remote server involves SSH login + SSH tunnel in addition to above. Come to our linux workshop for details!)

Launching JupyterLab

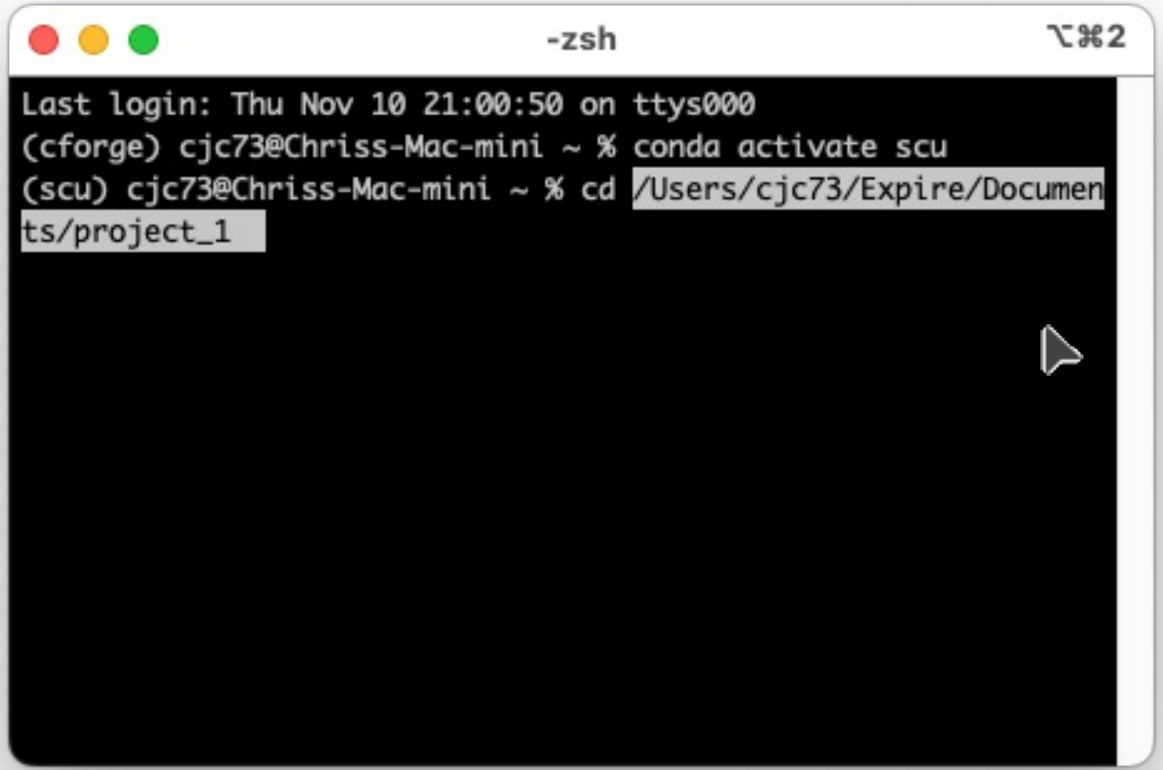


```
Last login: Thu Nov 10 21:00:50 on ttys000
(cforge) cjc73@Chriss-Mac-mini ~ % conda activate scu
(scu) cjc73@Chriss-Mac-mini ~ % cd 
```

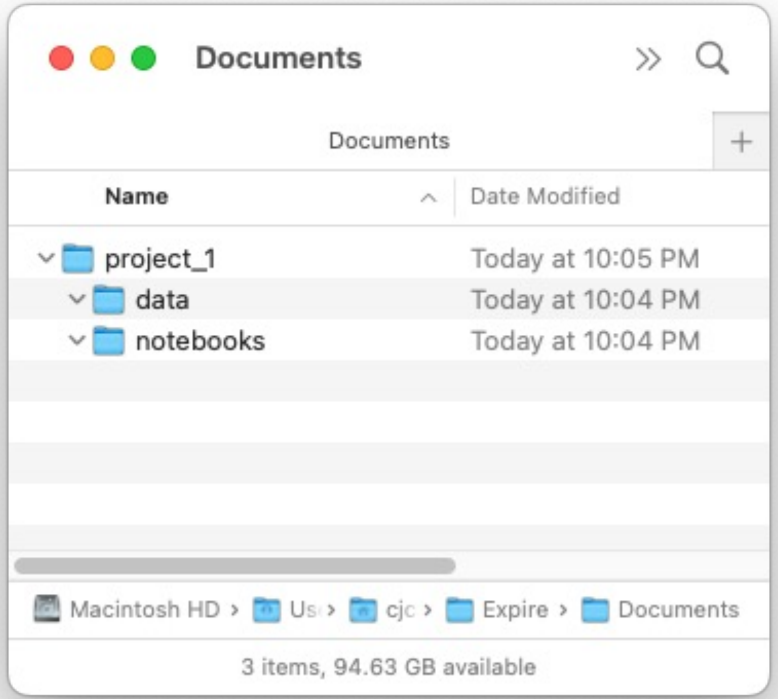

Documents
Documents
Macintosh HD › Us › cjc › Expire › Documents
3 items, 94.63 GB available

Name	Date Modified
project_1	Today at 10:05 PM
data	Today at 10:04 PM
notebooks	Today at 10:04 PM

Launching JupyterLab



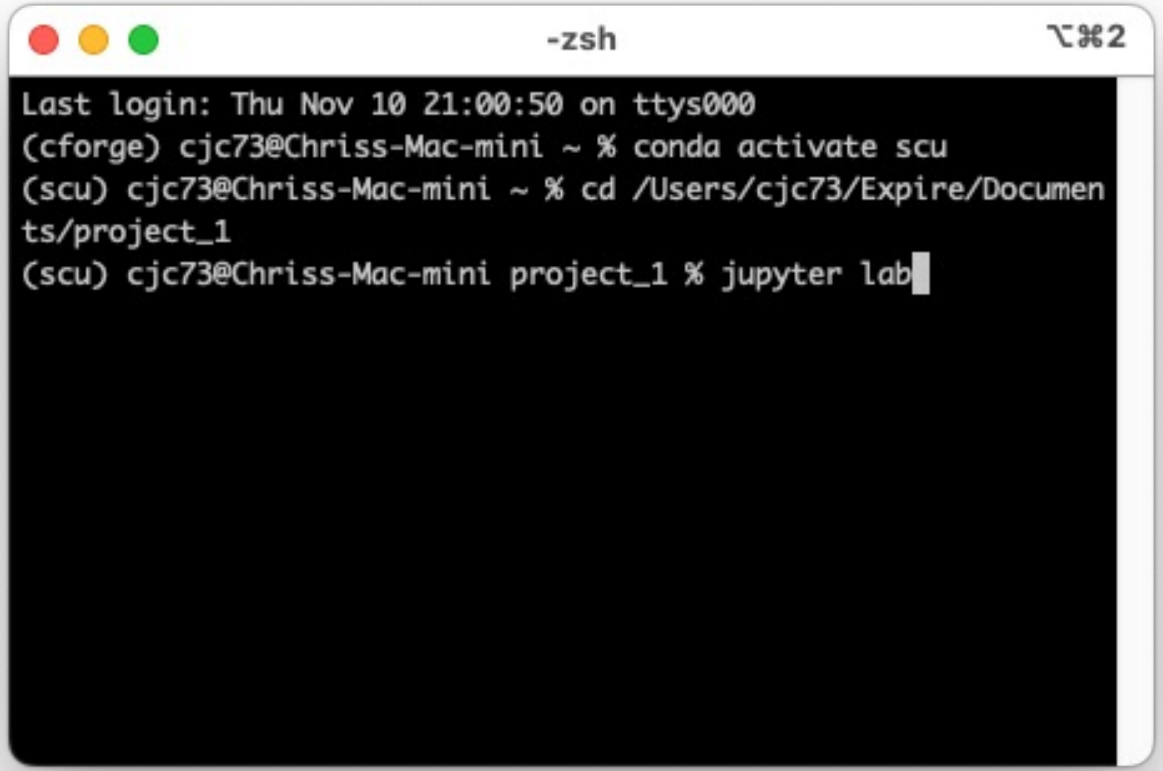
```
Last login: Thu Nov 10 21:00:50 on ttys000
(cforge) cjc73@Chriss-Mac-mini ~ % conda activate scu
(scu) cjc73@Chriss-Mac-mini ~ % cd /Users/cjc73/Expire/Documen
ts/project_1
```


Documents
Documents
Name
Date Modified

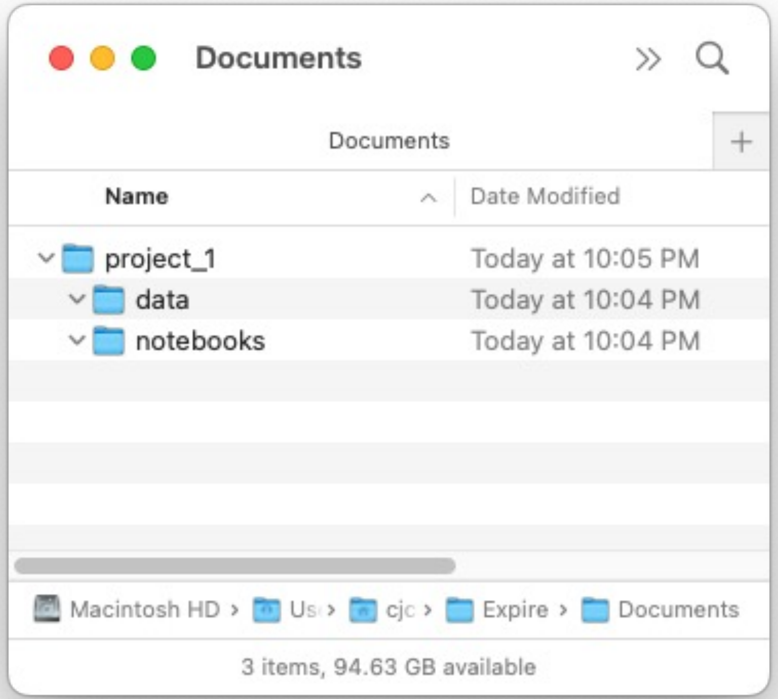
Name	Date Modified
project_1	Today at 10:05 PM
data	Today at 10:04 PM
notebooks	Today at 10:04 PM

Macintosh HD › Us › cjc › Expire › Documents
3 items, 94.63 GB available

Launching JupyterLab



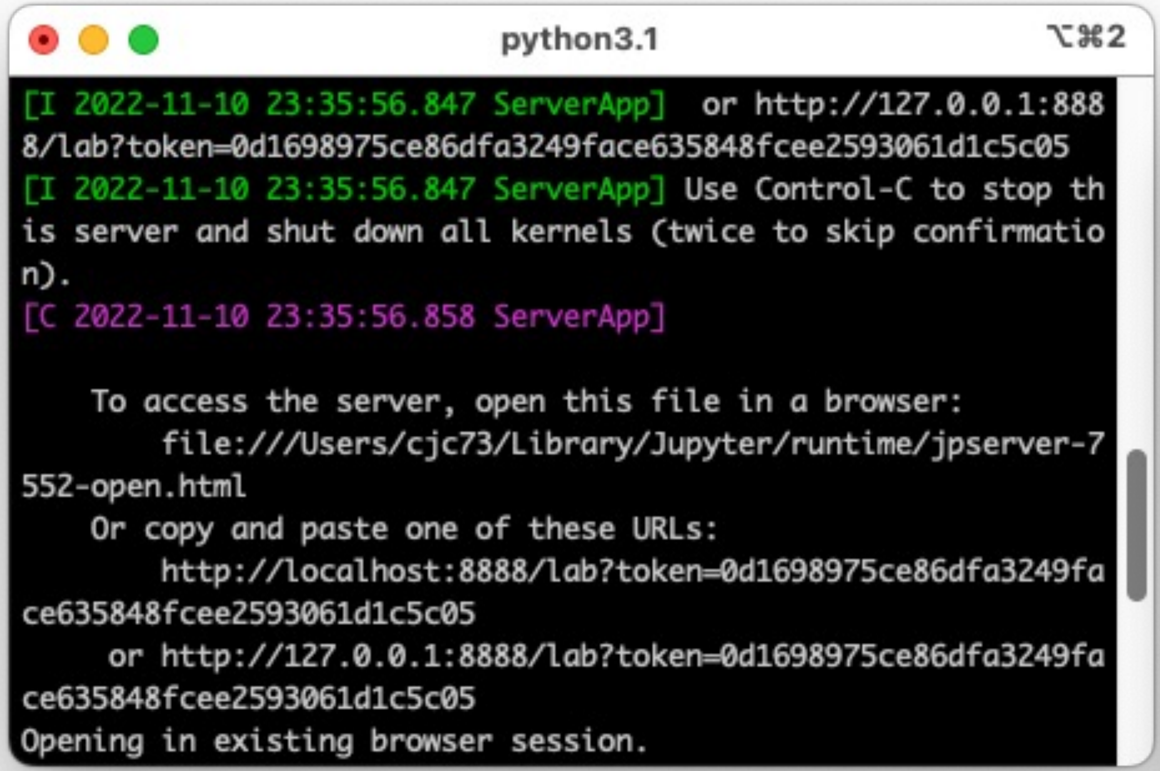
```
Last login: Thu Nov 10 21:00:50 on ttys000
(cforge) cjc73@Chriss-Mac-mini ~ % conda activate scu
(scu) cjc73@Chriss-Mac-mini ~ % cd /Users/cjc73/Expire/Documen
ts/project_1
(scu) cjc73@Chriss-Mac-mini project_1 % jupyter lab
```



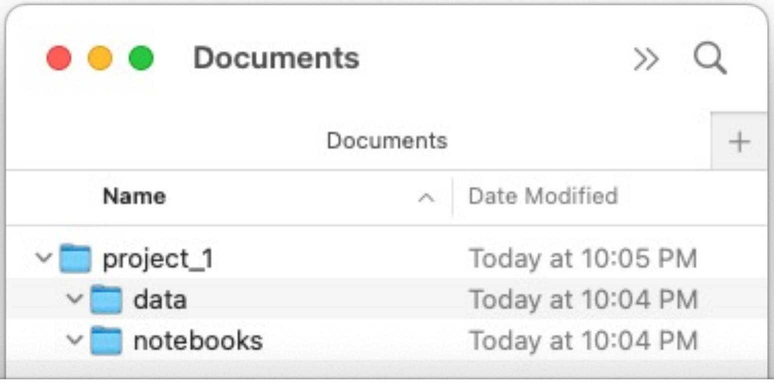
Name	Date Modified
project_1	Today at 10:05 PM
data	Today at 10:04 PM
notebooks	Today at 10:04 PM

Macintosh HD › Us › cjc › Expire › Documents
3 items, 94.63 GB available

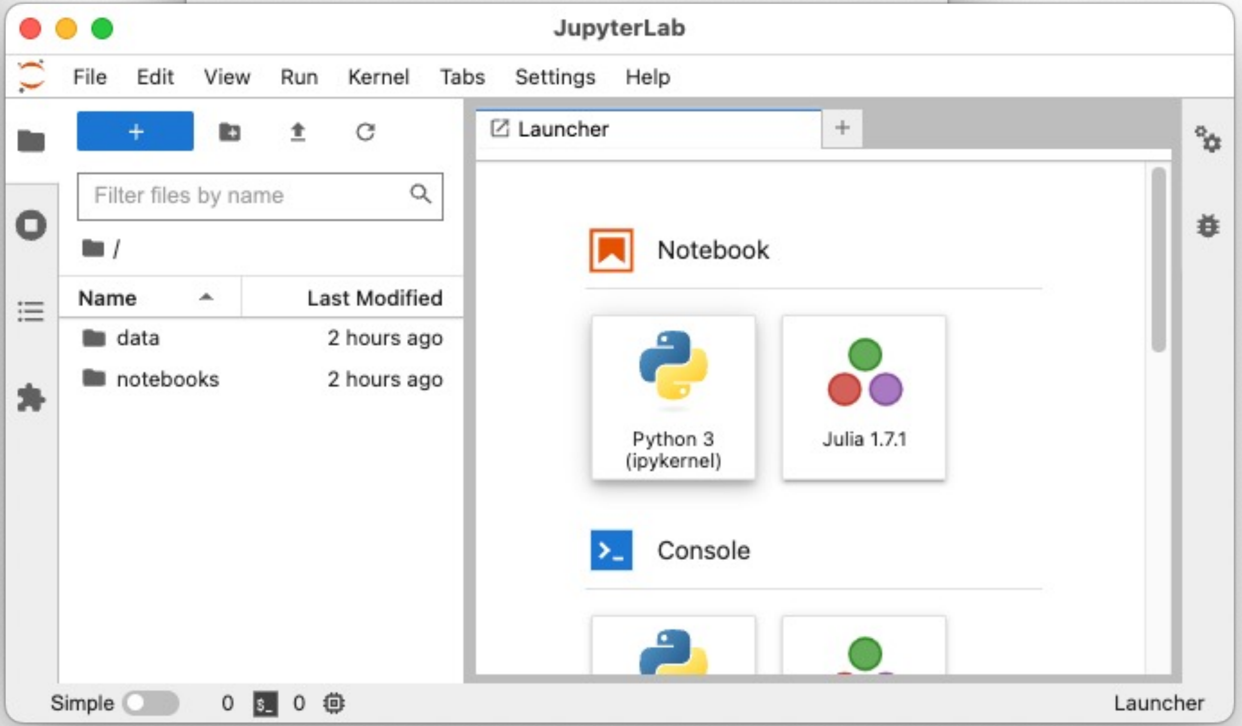
Launching JupyterLab



```
[I 2022-11-10 23:35:56.847 ServerApp]  or http://127.0.0.1:888
8/lab?token=0d1698975ce86dfa3249face635848fcee2593061d1c5c05
[I 2022-11-10 23:35:56.847 ServerApp] Use Control-C to stop th
is server and shut down all kernels (twice to skip confirmatio
n).
[C 2022-11-10 23:35:56.858 ServerApp]

    To access the server, open this file in a browser:
        file:///Users/cjc73/Library/Jupyter/runtime/jpserver-7
552-open.html
    Or copy and paste one of these URLs:
        http://localhost:8888/lab?token=0d1698975ce86dfa3249fa
ce635848fcee2593061d1c5c05
     or http://127.0.0.1:8888/lab?token=0d1698975ce86dfa3249fa
ce635848fcee2593061d1c5c05
Opening in existing browser session.
```



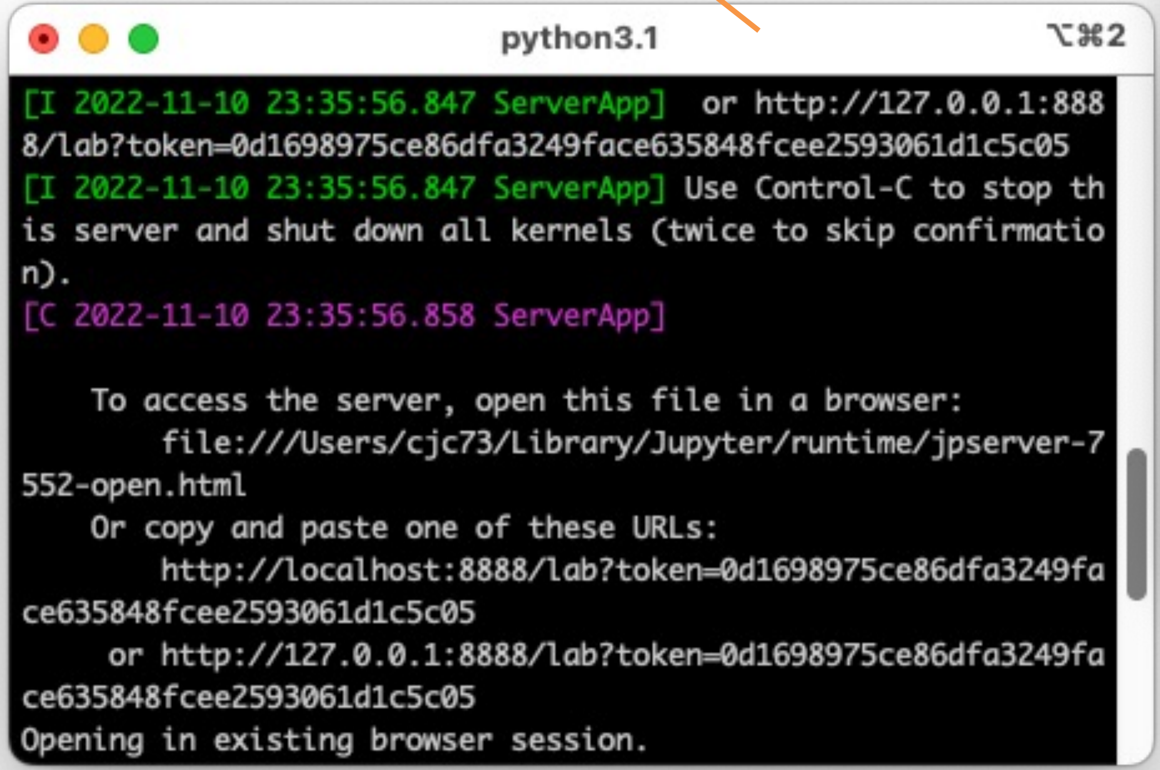
Name	Date Modified
project_1	Today at 10:05 PM
data	Today at 10:04 PM
notebooks	Today at 10:04 PM



Name	Last Modified
data	2 hours ago
notebooks	2 hours ago

Closing JupyterLab

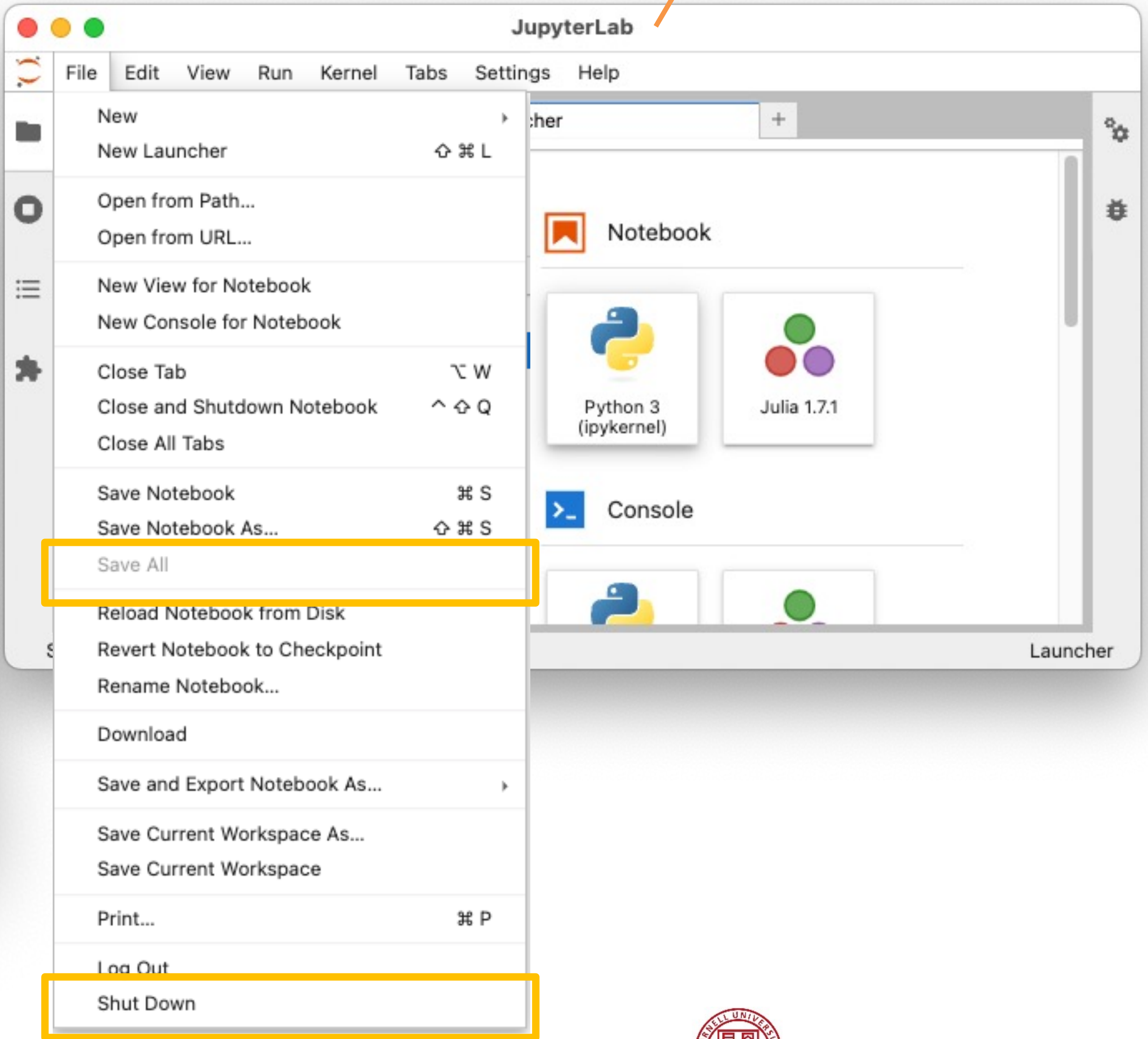
Web Server



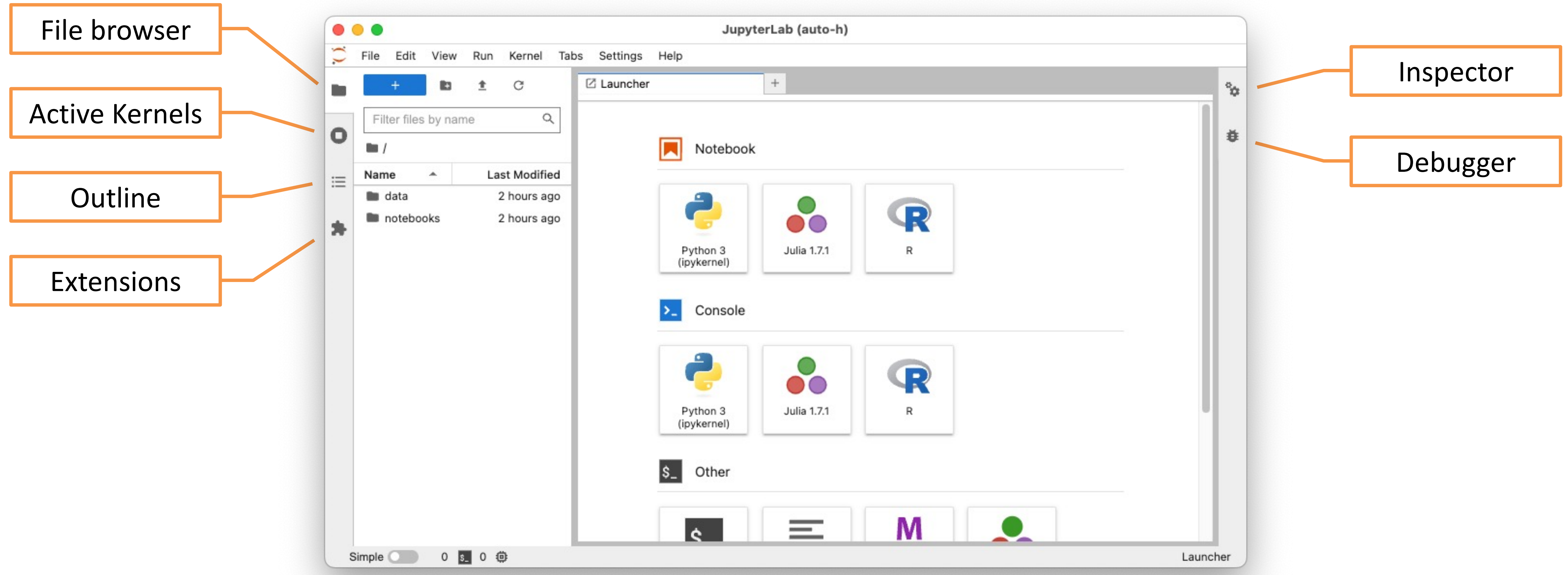
```
python3.1                                        ⌥⌘2
[I 2022-11-10 23:35:56.847 ServerApp]  or http://127.0.0.1:888
8/lab?token=0d1698975ce86dfa3249face635848fcee2593061d1c5c05
[I 2022-11-10 23:35:56.847 ServerApp] Use Control-C to stop th
is server and shut down all kernels (twice to skip confirmatio
n).
[C 2022-11-10 23:35:56.858 ServerApp]

    To access the server, open this file in a browser:
        file:///Users/cjc73/Library/Jupyter/runtime/jpserver-7
552-open.html
    Or copy and paste one of these URLs:
        http://localhost:8888/lab?token=0d1698975ce86dfa3249fa
ce635848fcee2593061d1c5c05
     or http://127.0.0.1:8888/lab?token=0d1698975ce86dfa3249fa
ce635848fcee2593061d1c5c05
Opening in existing browser session.
```

Leave this window open until you are done.
Closing it stops the web server.

Web Browser

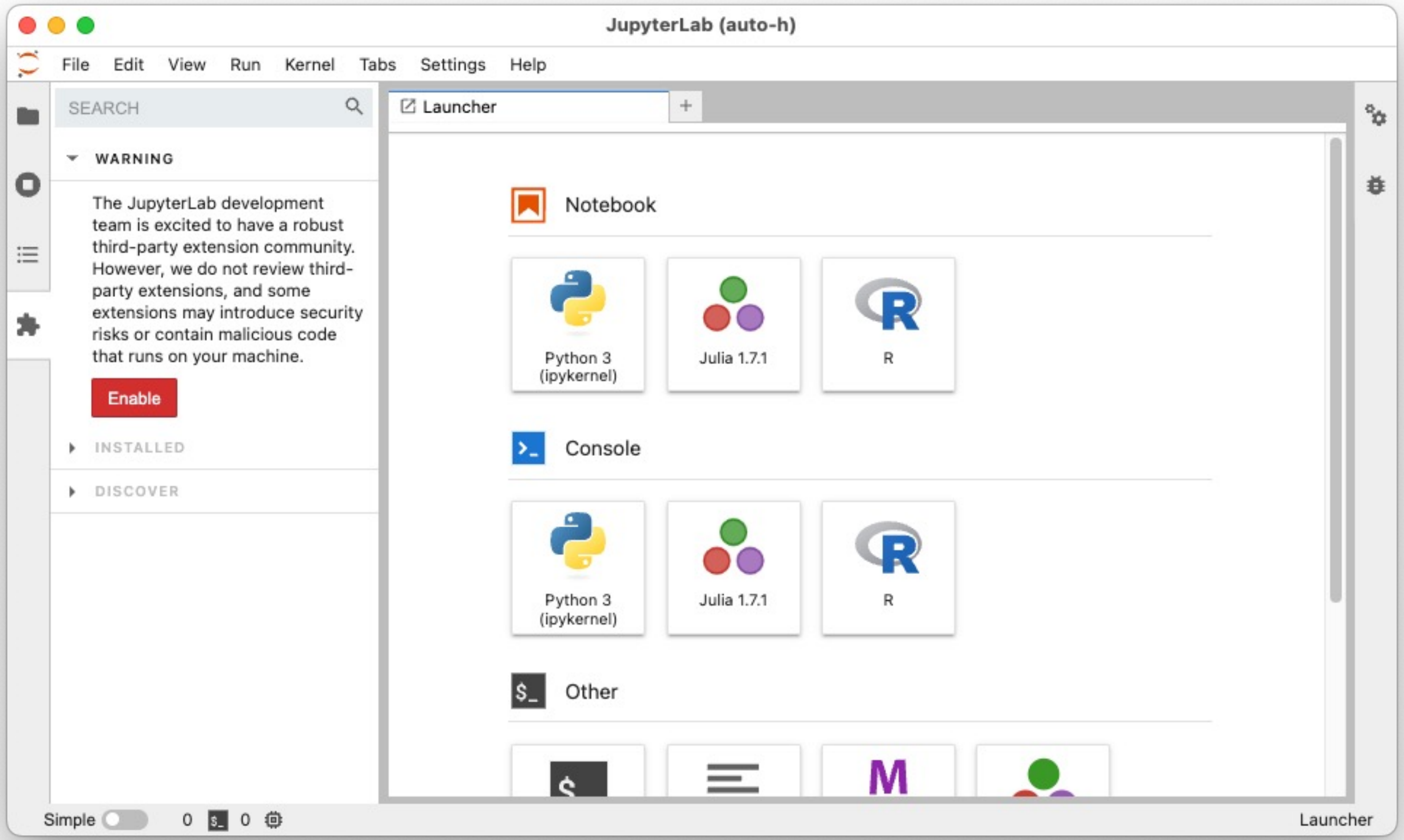


JupyterLab Interface



Activate JupyterLab Extensions (first time)

Extensions

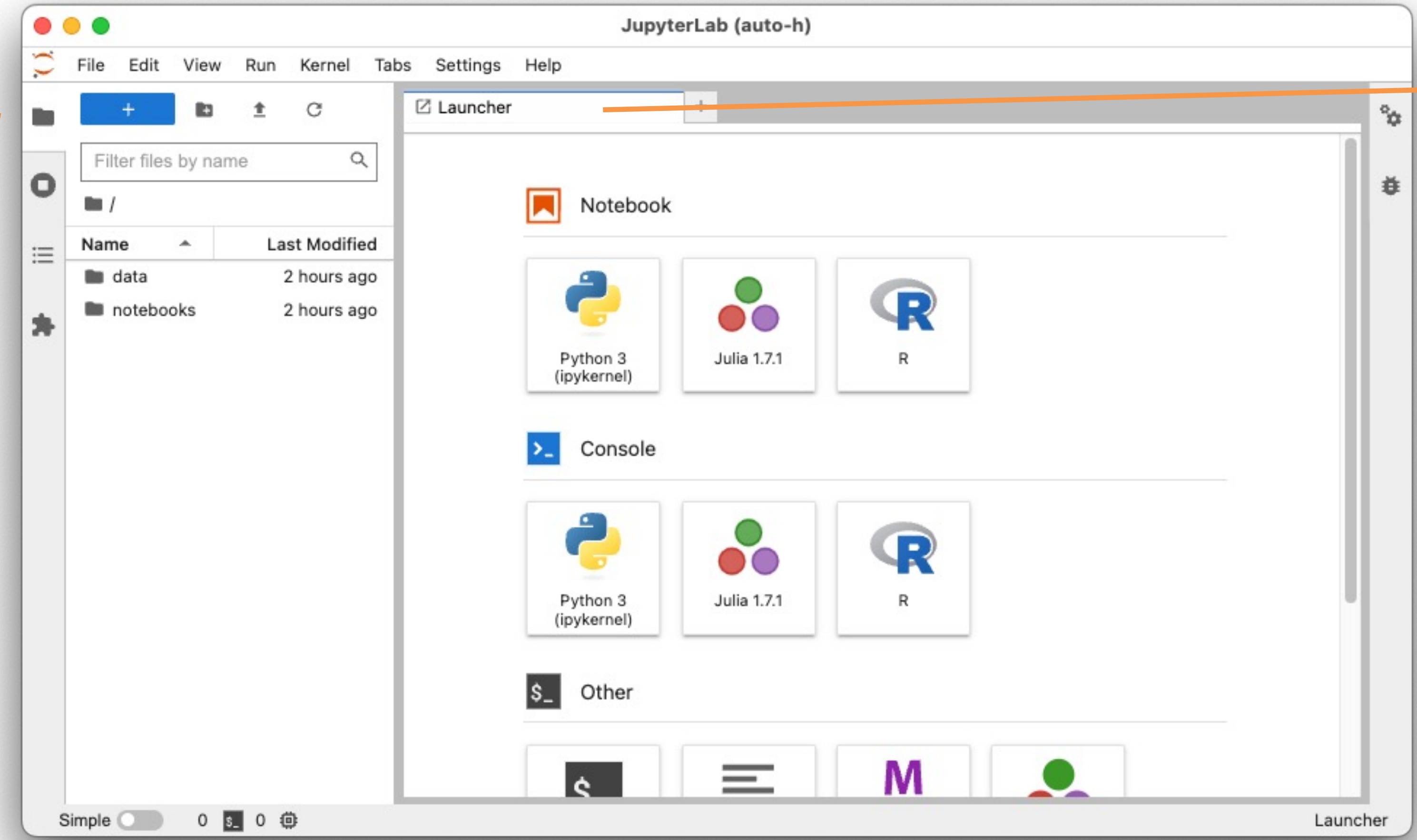


The JupyterLab development team is excited to have a robust third-party extension community. However, we do not review third-party extensions, and some extensions may introduce security risks or contain malicious code that runs on your machine.

Create a Notebook

File Browser

Navigate to "notebooks"



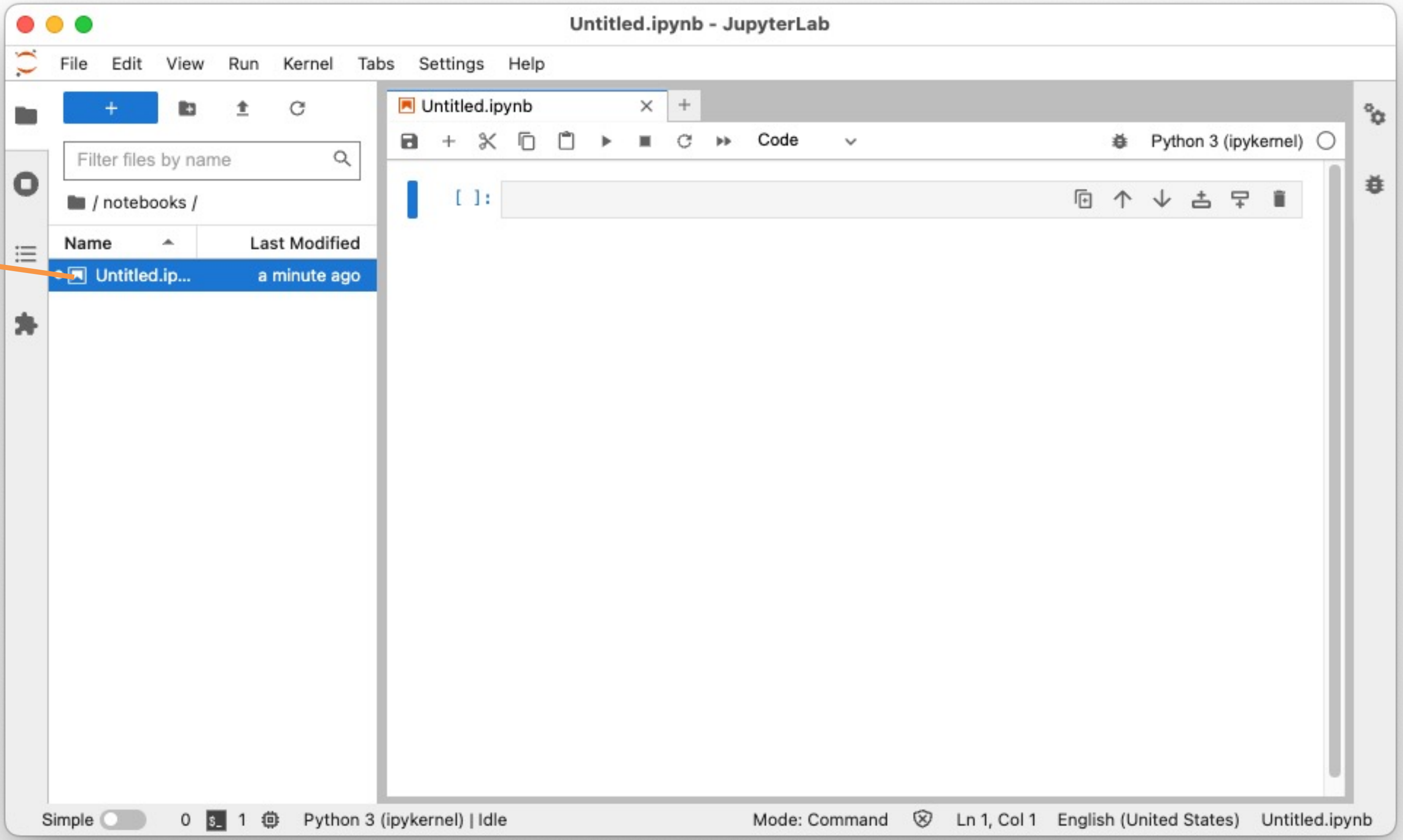
Launcher

Click "Python 3" under "Notebook"

Name your notebook

Untitled Notebook

Right-click to rename

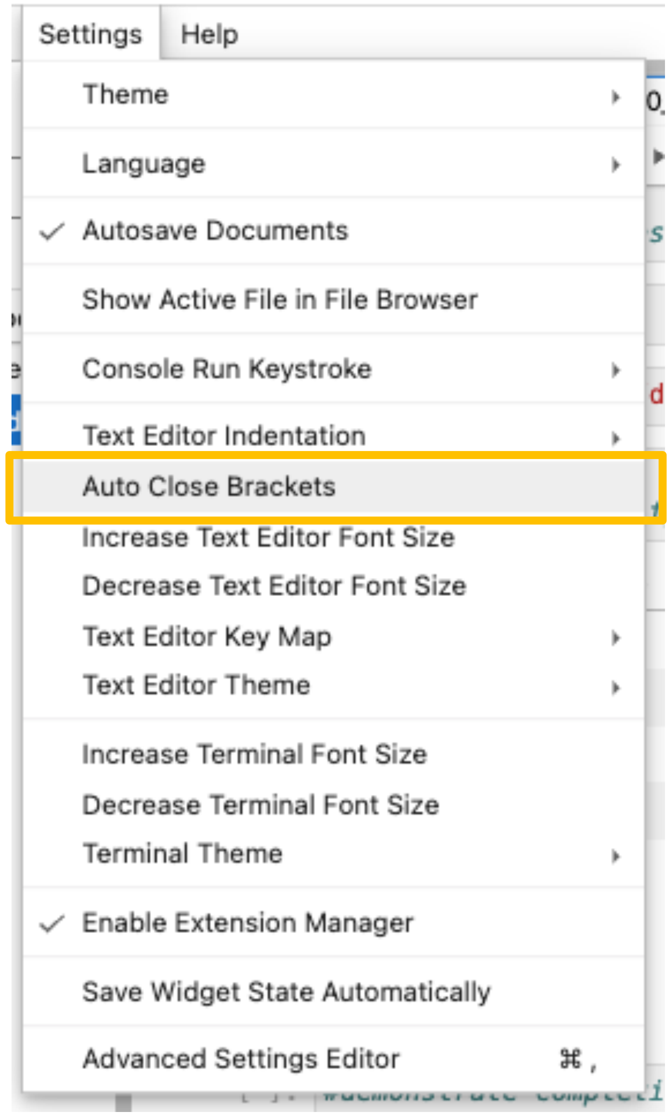


Tip for naming notebooks:

- Notebooks will be listed in alphabetical order.
  - I use 2 digits to prefix the names
    - e.g.
      - “00_project_readme.ipynb”
      - “01_download_data.ipynb”
      - “10_clean_data.ipynb”
      - “20_descriptive_viz.ipynb”
      - “99_scratch.ipynb”
  - Leave gaps in the numbering so you can insert other docs and keep them sorted.

Editor config:

- Auto Close Brackets can be a great timesaver


Settings
Help
Theme
Language
✓ Autosave Documents
Show Active File in File Browser
Console Run Keystroke
Text Editor Indentation
Auto Close Brackets
Increase Text Editor Font Size
Decrease Text Editor Font Size
Text Editor Key Map
Text Editor Theme
Increase Terminal Font Size
Decrease Terminal Font Size
Terminal Theme
✓ Enable Extension Manager
Save Widget State Automatically
Advanced Settings Editor ⌘ ,

Demo time

- Notebooks structure and usage
- Features Demo