

ASTR 119: Session 5

Jupyter Notebooks and Matplotlib



Outline

- 1) New homework due 10/14 at 9:50am
- 2) Visualization of the Day
- 3) Numpy
- 4) Arrays and Memory
- 5) Jupyter
- 6) First Notebook Program
- 7) Clearing Outputs
- 8) Matplotlib basics
- 9) Our first figure, saving figures to a file
- 10) Save your work to GitHub



Homework, due Oct 14, 9:50am

- 1) Make a Jupyter notebook, import numpy and matplotlib, and in a single line use numpy to create an array x running $x = [0, 2 * \pi]$ inclusive with 1000 values. Then use matplotlib to plot the following functions on a single plot, using the x range $x = [0, 2\pi]$ and the vertical range $y = [-1, 10]$:
 - a). $y(x) = 5.5 \cos(2 * x) + 5.5$
 - b). $y(x) = 0.02 * \exp(x)$
 - c). $y(x) = 0.25 * x^2 + 0.1 \sin(10 * x)$
- 2) Make the plot's y label "Measures of Awesomeness" and the x label "Time in ASTR 119".
- 3) In your ASTR-119 repository, create a folder "astr-119-hw-3" and save there the notebook "hw-3.ipynb".
- 4) Make an issue for your repository, and tag your TA. PLEASE CLEAR ALL THE CELLS BEFORE YOU COMMIT THE NOTEBOOK. Your TA will clone your code and email you commented version of the code and a grade.
- 5) To get the full grade possible, all the notebooks will need to run to completion without errors and produce the requested plots.





UC SANTA CRUZ



[Scipy.org](#)

NumPy

NumPy is the fundamental package for scientific computing with Python. It contains among other things:

- a powerful N-dimensional array object
- sophisticated (broadcasting) functions
- tools for integrating C/C++ and Fortran code
- useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

NumPy is licensed under the [BSD license](#), enabling reuse with few restrictions.

Getting Started

- [Getting NumPy](#)
- [Installing the SciPy Stack](#)
- [NumPy and SciPy documentation page](#)
- [NumPy Tutorial](#)
- [NumPy for MATLAB® Users](#)
- [NumPy functions by category](#)
- [NumPy Mailing List](#)

For more information on the SciPy Stack (for which NumPy provides the fundamental array data structure), see [scipy.org](#).

[About NumPy](#)

[License](#)

[Code of Conduct](#)

[Old array packages](#)

Using pip to install numpy

Mac OSX / Linux:

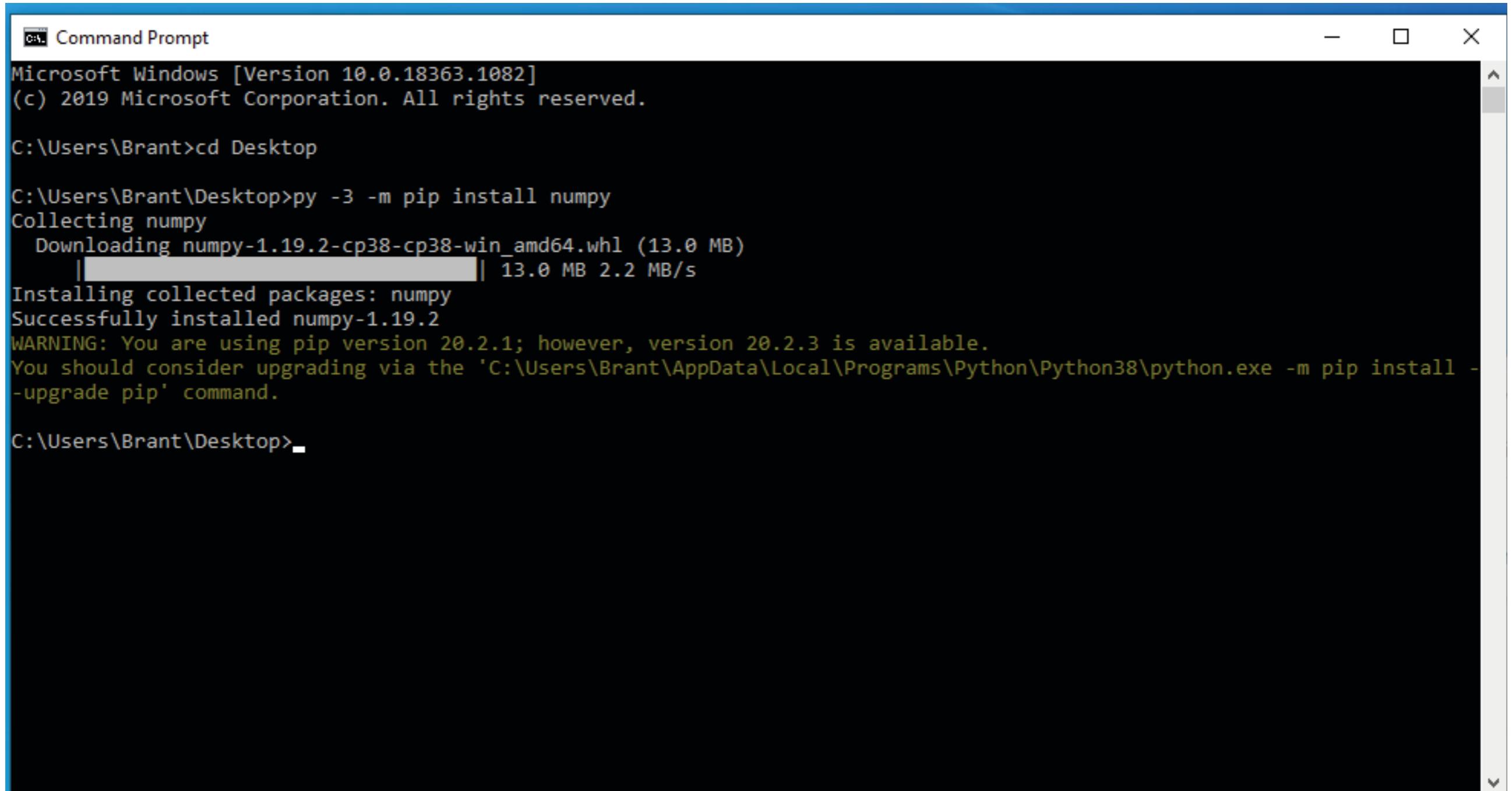
```
$ pip3 install numpy
```

Windows:

```
$ py -3 -m pip install numpy
```



Installation on Windows



A screenshot of a Microsoft Windows Command Prompt window titled "Command Prompt". The window shows the following text output:

```
Microsoft Windows [Version 10.0.18363.1082]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Brant>cd Desktop

C:\Users\Brant\Desktop>py -3 -m pip install numpy
Collecting numpy
  Downloading numpy-1.19.2-cp38-cp38-win_amd64.whl (13.0 MB)
    |████████| 13.0 MB 2.2 MB/s

Installing collected packages: numpy
Successfully installed numpy-1.19.2
WARNING: You are using pip version 20.2.1; however, version 20.2.3 is available.
You should consider upgrading via the 'C:\Users\Brant\AppData\Local\Programs\Python\Python38\python.exe -m pip install --upgrade pip' command.

C:\Users\Brant\Desktop>
```



Numpy

```
1 import numpy as np
2
3 x = 1.0      #define a float
4 y = 2.0      #define another float
5
6 #trigonometry
7 print(np.sin(x))          #sin(x)
8 print(np.cos(x))          #cos(x)
9 print(np.tan(x))          #tan(x)
10 print(np.arcsin(x))       #arcsin(x)
11 print(np.arccos(x))       #arccos(x)
12 print(np.arctan(x))       #arctan(x)
13 print(np.arctan2(x,y))    #arctan(x/y)
14 print(np.rad2deg(x))      #convert rad to deg
15
16 #hyperbolic functions
17 print(np.sinh(x))         #sinh(x)
18 print(np.cosh(x))         #cosh(x)
19 print(np.tanh(x))         #tanh(x)
20 print(np.arcsinh(x))      #arcsinh(x)
21 print(np.arccosh(x))      #arccosh(x)
22 print(np.arctanh(x))      #arctanh(x)
```



Numpy

```
1. brant@eduroam-169-233-222-100.ucsc.edu: /Users/brant/Desktop (bash)
[16:47:13][brant@eduroam-169-233-222-100:~/Desktop]$ python demo_numpy.py
0.841470984808
0.540302305868
1.55740772465
1.57079632679
0.0
0.785398163397
0.463647609001
57.2957795131
1.17520119364
1.54308063482
0.761594155956
0.88137358702
0.0
demo_numpy.py:22: RuntimeWarning: divide by zero encountered in arctanh
  print(np.arctanh(x)) #arctanh(x)
inf
```



NumPy Continued

```
1 import numpy as np
2
3 x = 1.0      #define a float
4 y = 2.0      #define another float
5
6 #exponents and logarithms
7 print(np.exp(x))          #e^x
8 print(np.log(x))          #ln x
9 print(np.log10(x))         #log_10 x
10 print(np.log2(x))         #log_2 x
11
12 #min/max/misc
13 print(np.fabs(x))         #absolute val as a float
14 print(np.fmin(x,y))        #min of x and y
15 print(np.fmax(x,y))        #max of x and y
16
17 #populate arrays
18 n = 100                  #define an int
19 z = np.arange(n,dtype=float) #get an array [0.0,n-1.]
20 z *= 2.0*np.pi /float(n-1) #z = [0,2*pi]
21 sin_z = np.sin(z)          #get an array sin(z)
22
23 #interpolation
24 print(np.interp(0.75,z,sin_z)) #interpolate sin(0.75)
25 print(np.sin(0.75))
```



Numpy, Continued

```
● ● ● 1. brant@eduroam-169-233-222-100.ucsc.edu: /Users;brant/Desktop (bash)
[16:48:23][brant@eduroam-169-233-222-100:~/Desktop]$ python demo_numpy_continued
.py
2.71828182846
0.0
0.0
0.0
1.0
1.0
2.0
0.681436721777
0.681638760023
[16:48:26][brant@eduroam-169-233-222-100:~/Desktop]$
```



Arrays and Memory

How are arrays stored in memory?

Elements of arrays declared using the [] notation will be stored in contiguous regions of memory when possible:

a = [1., 2., 3.]

will be usually allocated as

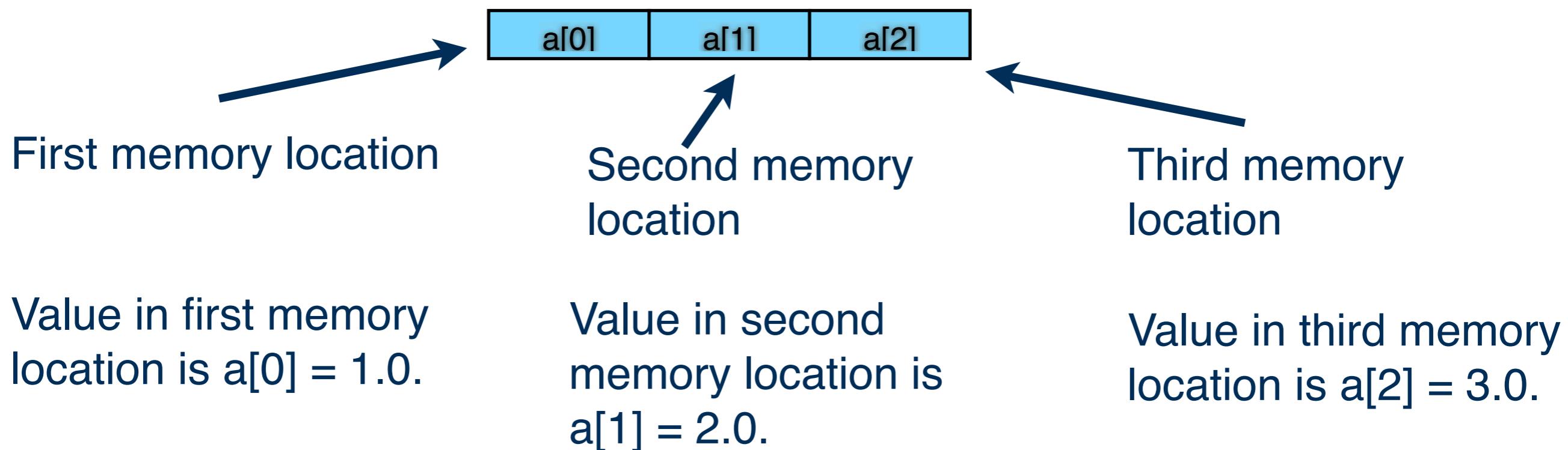


where each  is an 8 byte region in memory.



Arrays and Memory

Since arrays are concatenations of regions in memory, we can treat arrays as a beginning region (the memory location of $a[0]$) followed by subsequent memory locations associated with other elements in the array.



Each element of an array can be thought of as a **memory address** or the **value** contained at that memory address.



Jupyter Notebooks

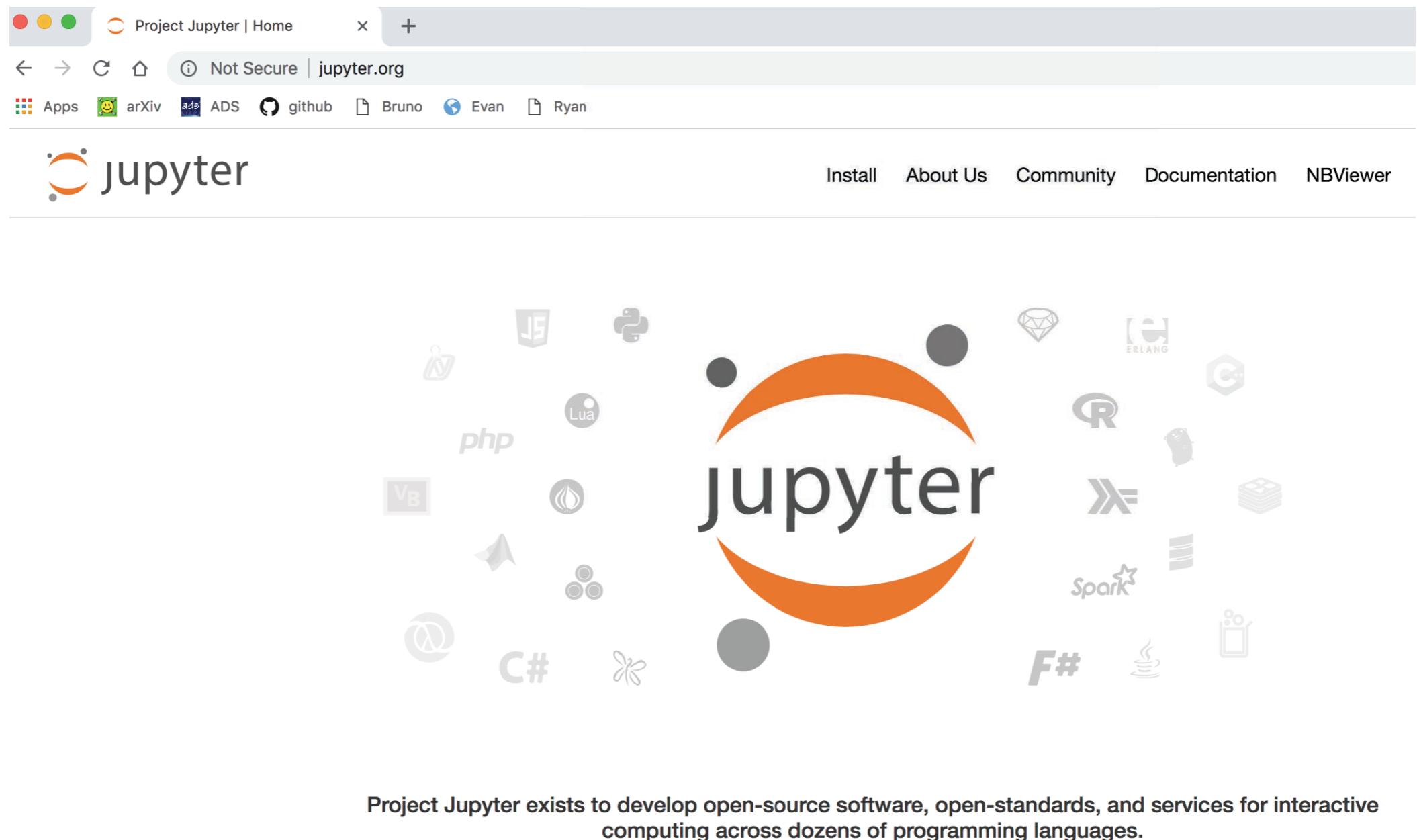
In science, we often want to:

- 1) Transparently share our code for reproducibility
- 2) Comment our code for clarity
- 3) Plot the results of our code interactively to understand our data / calcs.

Jupyter notebooks provide lots of functionality and flexibility, they solve the above problems for us, and they can do much, much more.



Jupyter Basics



A screenshot of a web browser displaying the Project Jupyter homepage. The address bar shows 'Project Jupyter | Home' and 'Not Secure | jupyter.org'. Below the address bar is a navigation bar with links to 'Apps', 'arXiv', 'ADS', 'github', 'Bruno', 'Evan', and 'Ryan'. The main content area features the Jupyter logo, which consists of a large orange circle containing the word 'jupyter' in lowercase, with smaller gray circles around it. In the background of the logo, there are various programming language logos like Python, JavaScript, PHP, and R. Below the logo is a text block: 'Project Jupyter exists to develop open-source software, open-standards, and services for interactive computing across dozens of programming languages.'

jupyter.org

Using pip to install Jupyter

Mac OSX / Linux:

```
$ pip3 install jupyter
```

Windows:

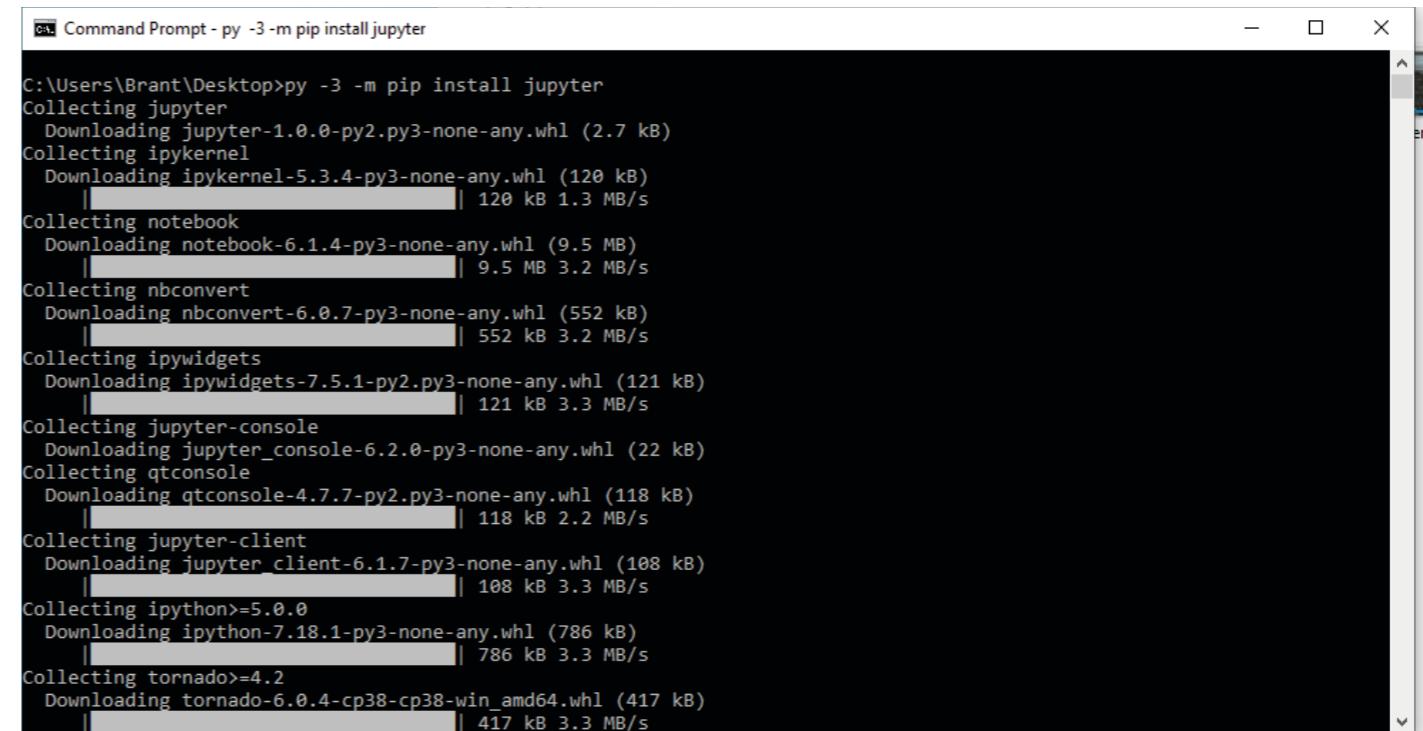
```
$ py -3 -m pip install jupyter
```



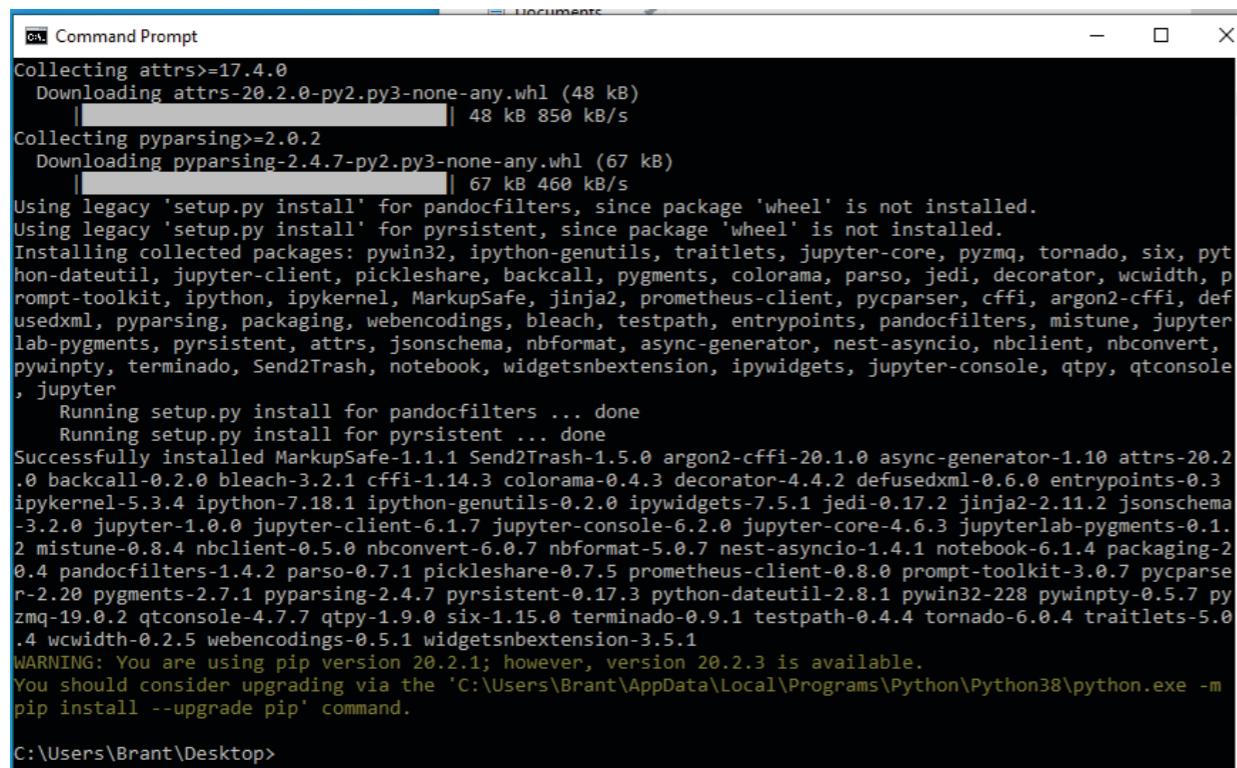
Installing Jupyter on Windows

In a command prompt:

```
$ py -3 -m pip install jupyter
```



```
C:\Users\Brant\Desktop>py -3 -m pip install jupyter
Collecting jupyter
  Downloading jupyter-1.0.0-py2.py3-none-any.whl (2.7 kB)
Collecting ipykernel
  Downloading ipykernel-5.3.4-py3-none-any.whl (120 kB)
|██████████| 120 kB 1.3 MB/s
Collecting notebook
  Downloading notebook-6.1.4-py3-none-any.whl (9.5 kB)
|██████████| 9.5 kB 3.2 MB/s
Collecting nbconvert
  Downloading nbconvert-6.0.7-py3-none-any.whl (552 kB)
|██████████| 552 kB 3.2 MB/s
Collecting ipywidgets
  Downloading ipywidgets-7.5.1-py2.py3-none-any.whl (121 kB)
|██████████| 121 kB 3.3 MB/s
Collecting jupyter-console
  Downloading jupyter_console-6.2.0-py3-none-any.whl (22 kB)
Collecting qtconsole
  Downloading qtconsole-4.7.7-py2.py3-none-any.whl (118 kB)
|██████████| 118 kB 2.2 MB/s
Collecting jupyter-client
  Downloading jupyter_client-6.1.7-py3-none-any.whl (108 kB)
|██████████| 108 kB 3.3 MB/s
Collecting ipython>=5.0.0
  Downloading ipython-7.18.1-py3-none-any.whl (786 kB)
|██████████| 786 kB 3.3 MB/s
Collecting tornado>=4.2
  Downloading tornado-6.0.4-cp38-cp38-win_amd64.whl (417 kB)
|██████████| 417 kB 3.3 MB/s
```



```
Collecting attrs>=17.4.0
  Downloading attrs-20.2.0-py2.py3-none-any.whl (48 kB)
|██████████| 48 kB 850 kB/s
Collecting pyparsing>=2.0.2
  Downloading pyparsing-2.4.7-py2.py3-none-any.whl (67 kB)
|██████████| 67 kB 460 kB/s
Using legacy 'setup.py install' for pandocfilters, since package 'wheel' is not installed.
Using legacy 'setup.py install' for pyrsistent, since package 'wheel' is not installed.
Installing collected packages: pywin32, ipython-genutils, traitlets, jupyter-core, pyzmq, tornado, six, pyhon-dateutil, jupyter-client, pickleshare, backcall, pygments, colorama, parso, jedi, decorator, wcwidth, prompt-toolkit, ipython, ipykernel, MarkupSafe, jinja2, prometheus-client, pycparser, cffi, argon2-cffi, defusedxml, pyparsing, packaging, webencodings, bleach, testpath, entrypoints, pandocfilters, mistune, jupyterlab-pygments, pyrsistent, attrs, jsonschema, nbformat, async-generator, nest-asyncio, nbclient, nbconvert, pywinpty, terminado, Send2Trash, notebook, widgetsnbextension, ipywidgets, jupyter-console, qtpy, qtconsole, jupyter
  Running setup.py install for pandocfilters ... done
  Running setup.py install for pyrsistent ... done
Successfully installed MarkupSafe-1.1.1 Send2Trash-1.5.0 argon2-cffi-20.1.0 async-generator-1.10 attrs-20.2.0 backcall-0.2.0 bleach-3.2.1 cffi-1.14.3 colorama-0.4.3 decorator-4.4.2 defusedxml-0.6.0 entrypoints-0.3 ipykernel-5.3.4 ipython-7.18.1 ipython-genutils-0.2.0 ipywidgets-7.5.1 jedi-0.17.2 jinja2-2.11.2 jsonschema-3.2.0 jupyter-1.0.0 jupyter-client-6.1.7 jupyter-console-6.2.0 jupyter-core-4.6.3 jupyterlab-pygments-0.1.2 mistune-0.8.4 nbclient-0.5.0 nbconvert-6.0.7 nbformat-5.0.7 nest-asyncio-1.4.1 notebook-6.1.4 packaging-20.4 pandocfilters-1.4.2 parso-0.7.1 pickleshare-0.7.5 prometheus-client-0.8.0 prompt-toolkit-3.0.7 pycparse-2.20 pygments-2.7.1 pyparsing-2.4.7 pyrsistent-0.17.3 python-dateutil-2.8.1 pywin32-228 pywintypes-0.5.7 pyzmq-19.0.2 qtconsole-4.7.7 qtpy-1.9.0 six-1.15.0 terminado-0.9.1 testpath-0.4.4 tornado-6.0.4 traitlets-5.0.4 wcwidth-0.2.5 webencodings-0.5.1 widgetsnbextension-3.5.1
WARNING: You are using pip version 20.2.1; however, version 20.2.3 is available.
You should consider upgrading via the 'C:\Users\Brant\AppData\Local\Programs\Python\Python38\python.exe -m pip install --upgrade pip' command.

C:\Users\Brant\Desktop>
```

A bunch of packages will likely get installed, but at the end you shouldn't have any errors.



Using Jupyter Notebooks

On Mac OSX, in a terminal:

\$ jupyter-notebook &

```
bash
[10:28:49] [brant@teal:~/Desktop]$ jupyter-notebook &
```

```
bash
[10:28:49] [brant@teal:~/Desktop]$ jupyter-notebook &
[2] 85513
[10:29:16] [brant@teal:~/Desktop]$ [I 10:29:16.884 NotebookApp] The port 8888 is
already in use, trying another port.
[I 10:29:16.887 NotebookApp] Serving notebooks from local directory: /Users/bran
t/Desktop
[I 10:29:16.887 NotebookApp] The Jupyter Notebook is running at:
[I 10:29:16.887 NotebookApp] http://localhost:8889/?token=fe70527703a9fe15f97afa
84e81d942cb849e78af88ece3d
[I 10:29:16.887 NotebookApp] or http://127.0.0.1:8889/?token=fe70527703a9fe15f9
7afa84e81d942cb849e78af88ece3d
[I 10:29:16.887 NotebookApp] Use Control-C to stop this server and shut down all
kernels (twice to skip confirmation).
[C 10:29:16.897 NotebookApp]

To access the notebook, open this file in a browser:
file:///Users/brant/Library/Jupyter/runtime/nbsrvr-85513-open.html
Or copy and paste one of these URLs:
http://localhost:8889/?token=fe70527703a9fe15f97afa84e81d942cb849e78af88
ece3d
or http://127.0.0.1:8889/?token=fe70527703a9fe15f97afa84e81d942cb849e78af88
ece3d
```

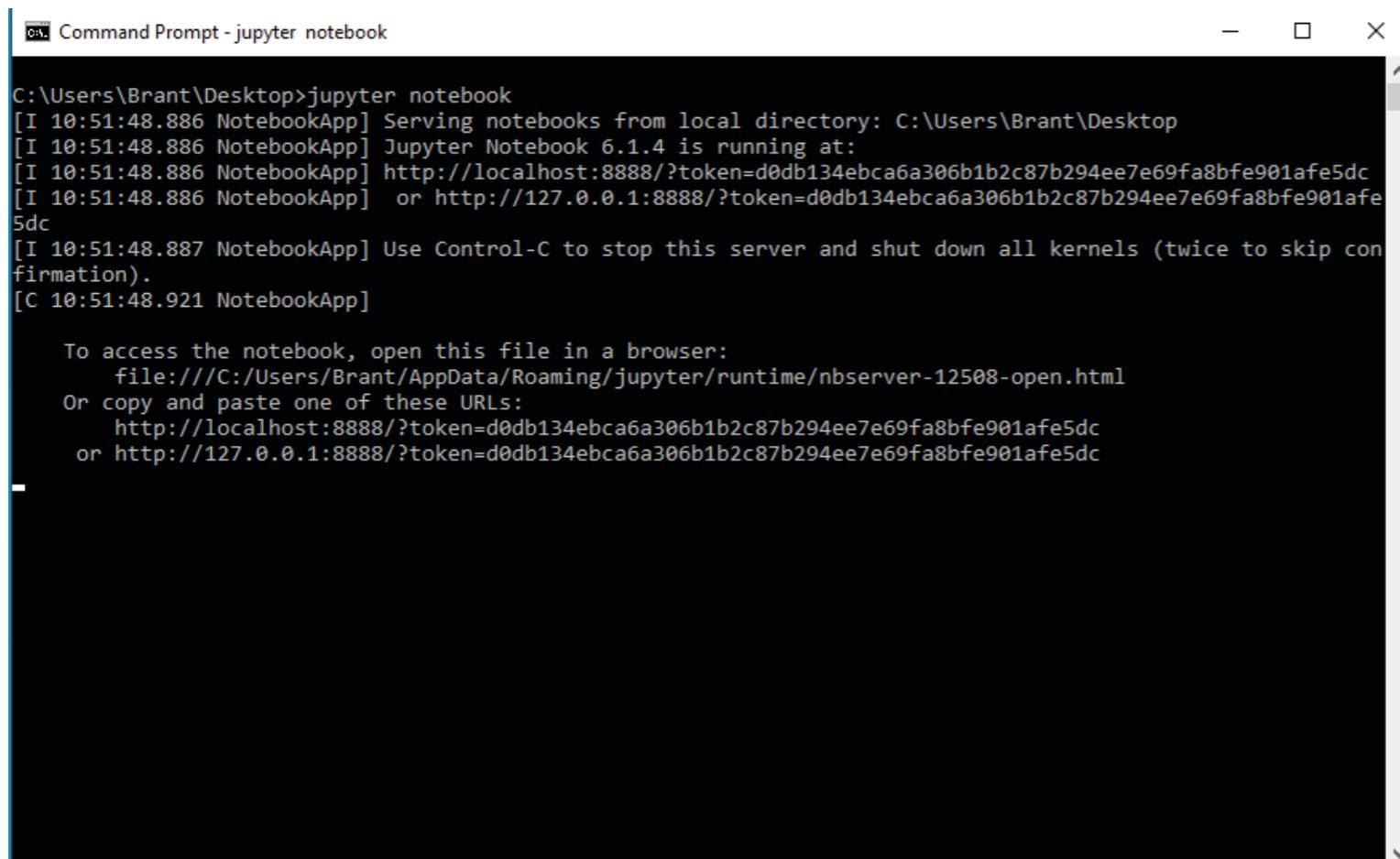
You'll see information about Jupyter Notebook running at a website, and your browser should open the page.



Using Jupyter Notebooks

On Windows in a command prompt:

\$ jupyter notebook



```
Command Prompt - jupyter notebook
C:\Users\Brant\Desktop>jupyter notebook
[I 10:51:48.886 NotebookApp] Serving notebooks from local directory: C:\Users\Brant\Desktop
[I 10:51:48.886 NotebookApp] Jupyter Notebook 6.1.4 is running at:
[I 10:51:48.886 NotebookApp] http://localhost:8888/?token=d0db134ebca6a306b1b2c87b294ee7e69fa8bfe901afe5dc
[I 10:51:48.886 NotebookApp] or http://127.0.0.1:8888/?token=d0db134ebca6a306b1b2c87b294ee7e69fa8bfe901afe5dc
[I 10:51:48.887 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 10:51:48.921 NotebookApp]

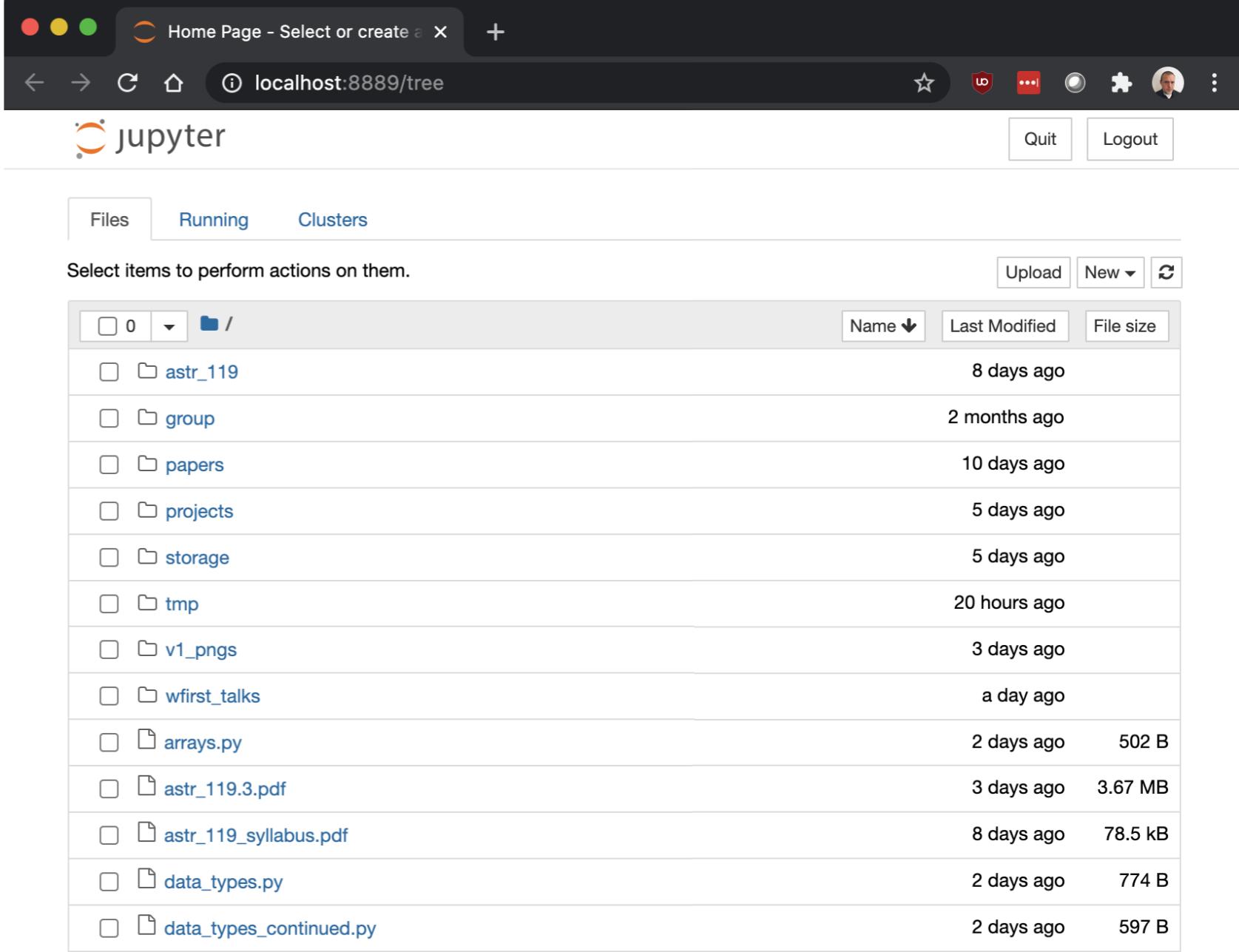
To access the notebook, open this file in a browser:
  file:///C:/Users/Brant/AppData/Roaming/jupyter/runtime/nbserver-12508-open.html
Or copy and paste one of these URLs:
  http://localhost:8888/?token=d0db134ebca6a306b1b2c87b294ee7e69fa8bfe901afe5dc
  or http://127.0.0.1:8888/?token=d0db134ebca6a306b1b2c87b294ee7e69fa8bfe901afe5dc
```

You'll see information about Jupyter Notebook running at a website, and your browser should open the page.



Using Jupyter Notebooks

Jupyter will open a webpage, usually at localhost:8888, which then shows the contents of the current working directory.

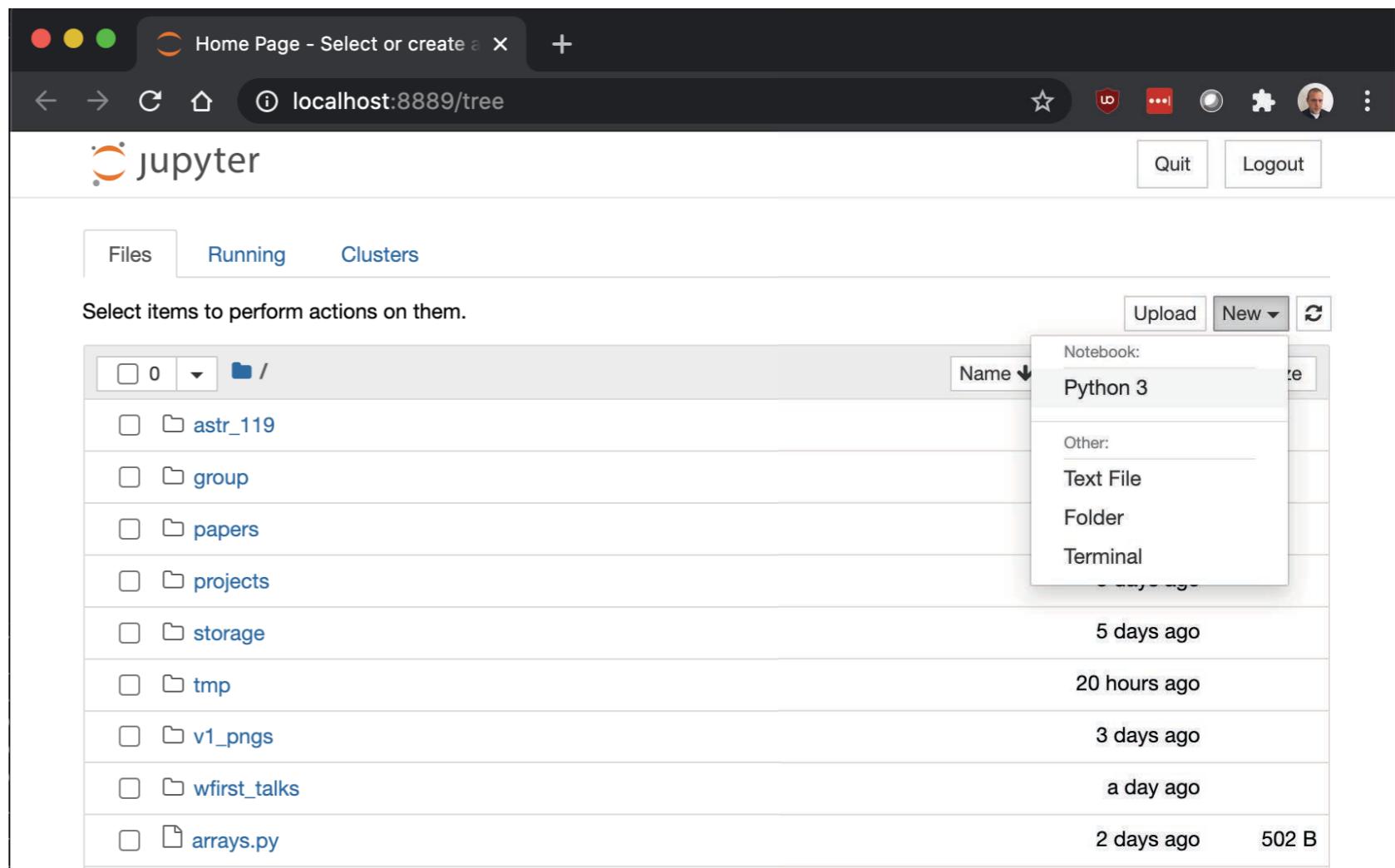


The screenshot shows a web browser window titled "Home Page - Select or create" with the URL "localhost:8889/tree". The page is titled "jupyter" and features a navigation bar with "Files", "Running", and "Clusters" tabs. Below the navigation bar is a message "Select items to perform actions on them." On the right side of the interface are buttons for "Upload", "New", and a refresh icon. The main area displays a file tree with the root folder "/". The contents of the root folder include:

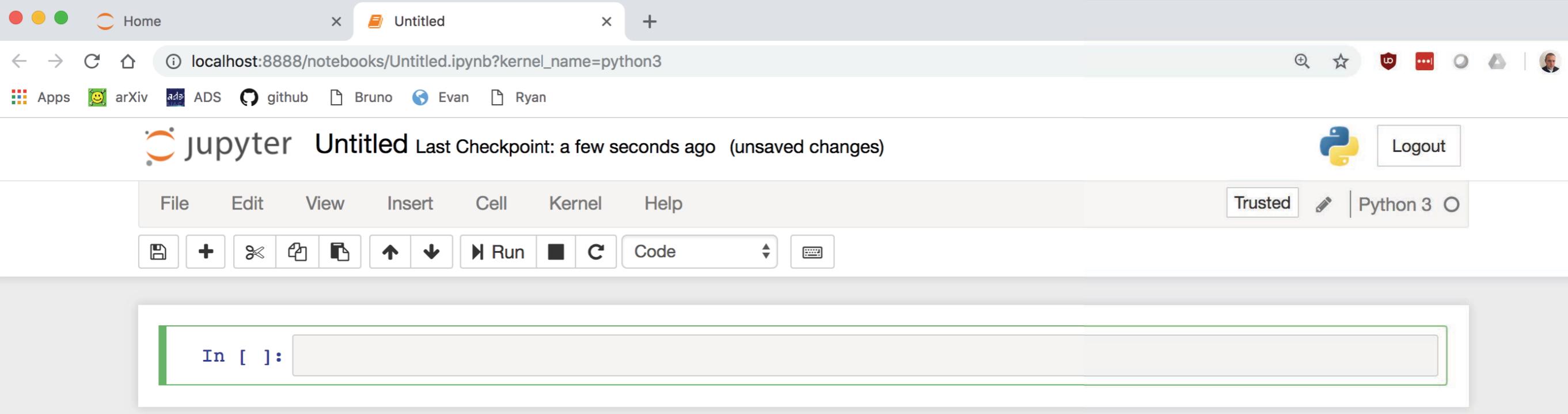
File/Folder	Last Modified	File size
astr_119	8 days ago	
group	2 months ago	
papers	10 days ago	
projects	5 days ago	
storage	5 days ago	
tmp	20 hours ago	
v1_pngs	3 days ago	
wfirst_talks	a day ago	
arrays.py	2 days ago	502 B
astr_119.3.pdf	3 days ago	3.67 MB
astr_119_syllabus.pdf	8 days ago	78.5 kB
data_types.py	2 days ago	774 B
data_types_continued.py	2 days ago	597 B

Using Jupyter Notebooks

Select New->Python 3 to open a new notebook using python.

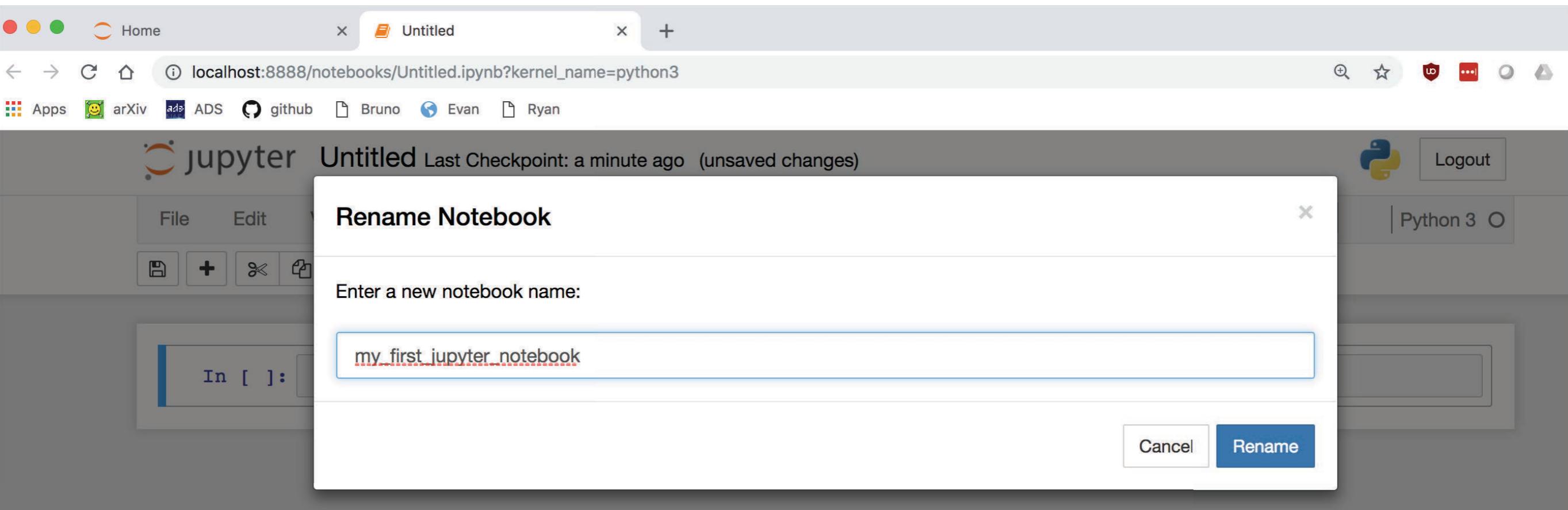


Here's a starting Untitled Notebook



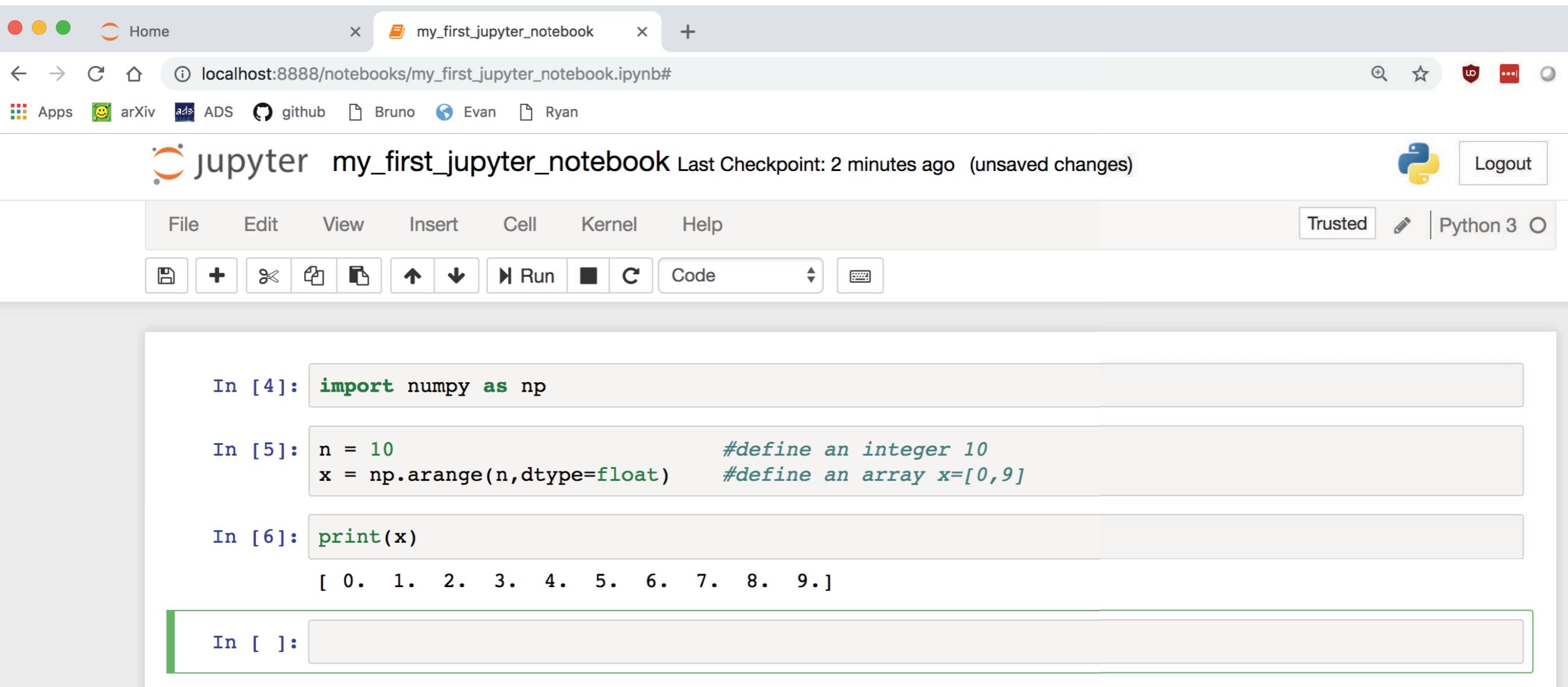
UC SANTA CRUZ

Rename Your Notebook



UC SANTA CRUZ

Our First Notebook Program



The screenshot shows a Jupyter Notebook interface running in a web browser. The title bar indicates the notebook is titled "my_first_jupyter_notebook". The toolbar includes standard browser controls (Home, Back, Forward) and a tab for the current notebook. Below the toolbar, the Jupyter logo and the notebook name are displayed, along with a message about the last checkpoint and unsaved changes. The menu bar offers options like File, Edit, View, Insert, Cell, Kernel, and Help. The toolbar below the menu bar includes icons for saving, running, and other operations. The main content area displays three code cells:

```
In [4]: import numpy as np
In [5]: n = 10
         x = np.arange(n,dtype=float)
In [6]: print(x)
         [ 0.  1.  2.  3.  4.  5.  6.  7.  8.  9.]
```

A new cell, In [], is currently selected and ready for input, indicated by a green border around its input field.



Clear All Output

A screenshot of a Jupyter Notebook interface running in a web browser. The browser's address bar shows the URL `localhost:8888/notebooks/my_first_jupyter_notebook.ipynb`. The notebook title is `my_first_jupyter_notebook`, and it indicates a last checkpoint was saved 2 minutes ago (autosaved). The interface includes a top navigation bar with File, Edit, View, Insert, Cell, Kernel, and Help menus, and a Python 3 kernel selection. A toolbar below the menu bar contains icons for file operations like save, new, cut, copy, paste, and cell navigation. The main workspace displays three code cells:

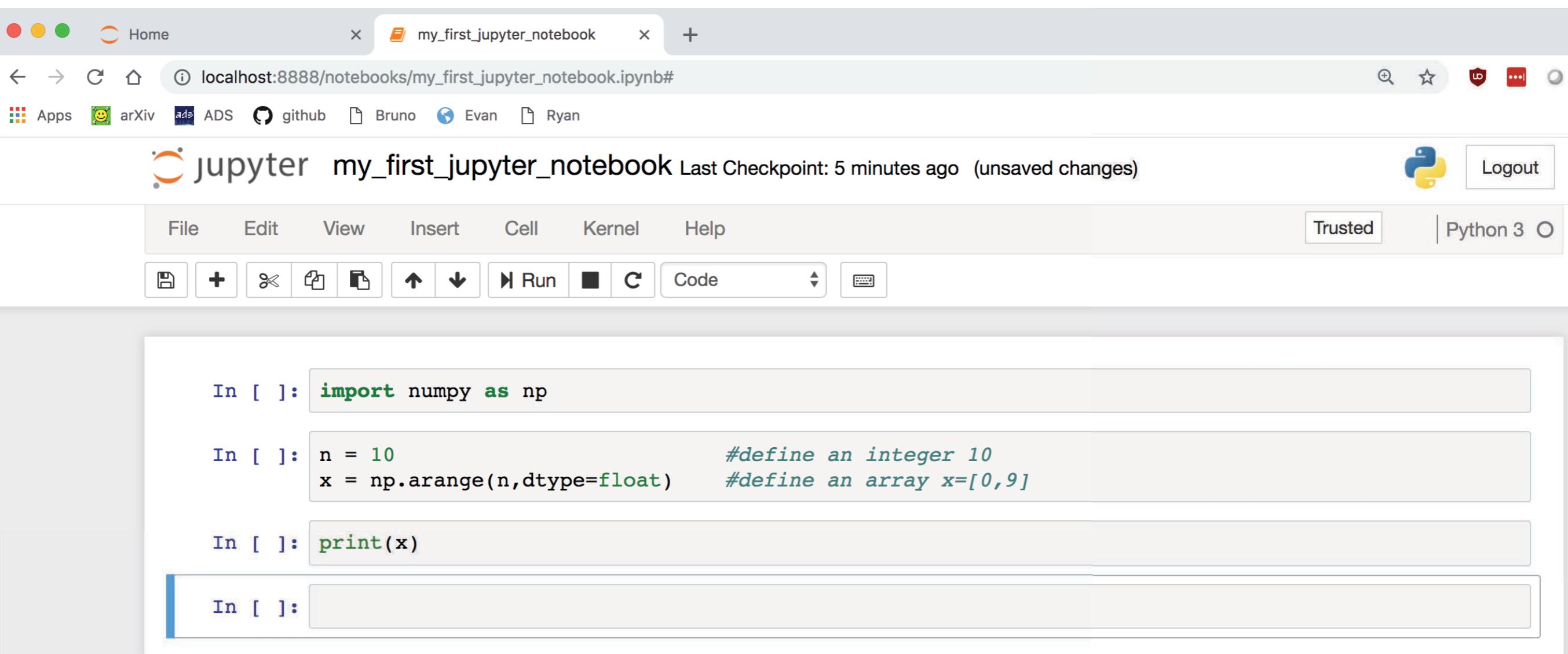
- In [1]: `import numpy`
- In [3]: `n = 10`
`x = np.arange(n)`
- In []: `print(x)`

The output area shows the results of the second cell:
`ne an integer 10`
`ne an array x=[0, 9]`

The `Cell` menu is open, revealing options for running cells, changing cell type, and managing outputs. The `All Output` submenu is currently selected, as indicated by its gray background. This submenu contains three items: `Toggle`, `Toggle Scrolling`, and `Clear`.



When in doubt: Clear All Output, Execute Top to Bottom

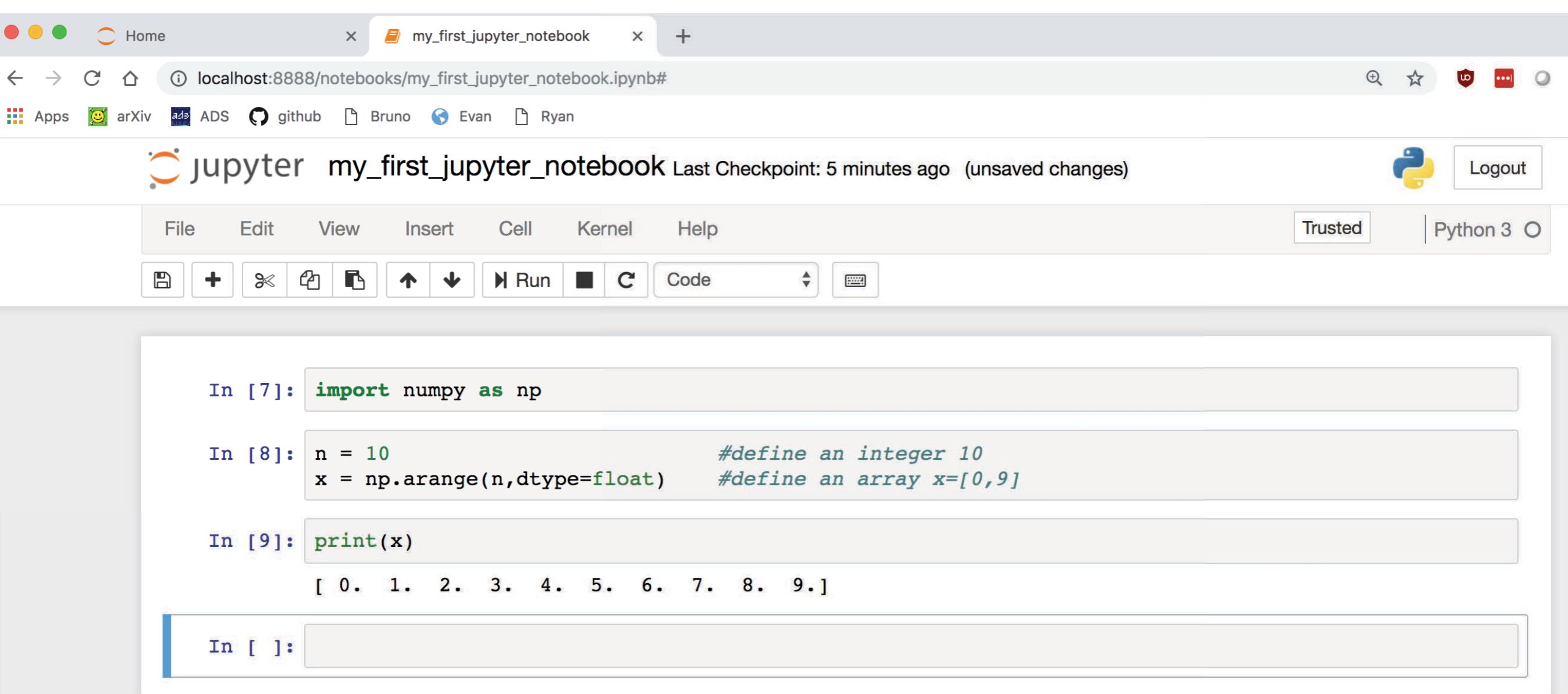


The screenshot shows a Jupyter Notebook interface running in a web browser. The title bar indicates the notebook is titled "my_first_jupyter_notebook". The toolbar includes standard browser controls (Home, Back, Forward) and Jupyter-specific buttons for file operations, cell selection, and code execution. The main area displays four code cells:

- In []: `import numpy as np`
- In []: `n = 10` #define an integer 10
`x = np.arange(n,dtype=float)` #define an array x=[0,9]
- In []: `print(x)`
- In []: (This cell is currently selected and empty.)



When in doubt: Clear All Output, Execute Top to Bottom



The screenshot shows a Jupyter Notebook interface in a web browser. The title bar indicates the notebook is titled "my_first_jupyter_notebook". The toolbar includes standard file operations like File, Edit, View, Insert, Cell, Kernel, and Help, along with specific notebook controls like Run, Cell, and Code. The main area displays four code cells:

- In [7]: `import numpy as np`
- In [8]: `n = 10`
`x = np.arange(n,dtype=float)` *#define an integer 10
#define an array x=[0,9]*
- In [9]: `print(x)`
`[0. 1. 2. 3. 4. 5. 6. 7. 8. 9.]`
- In []: (This cell is currently empty.)

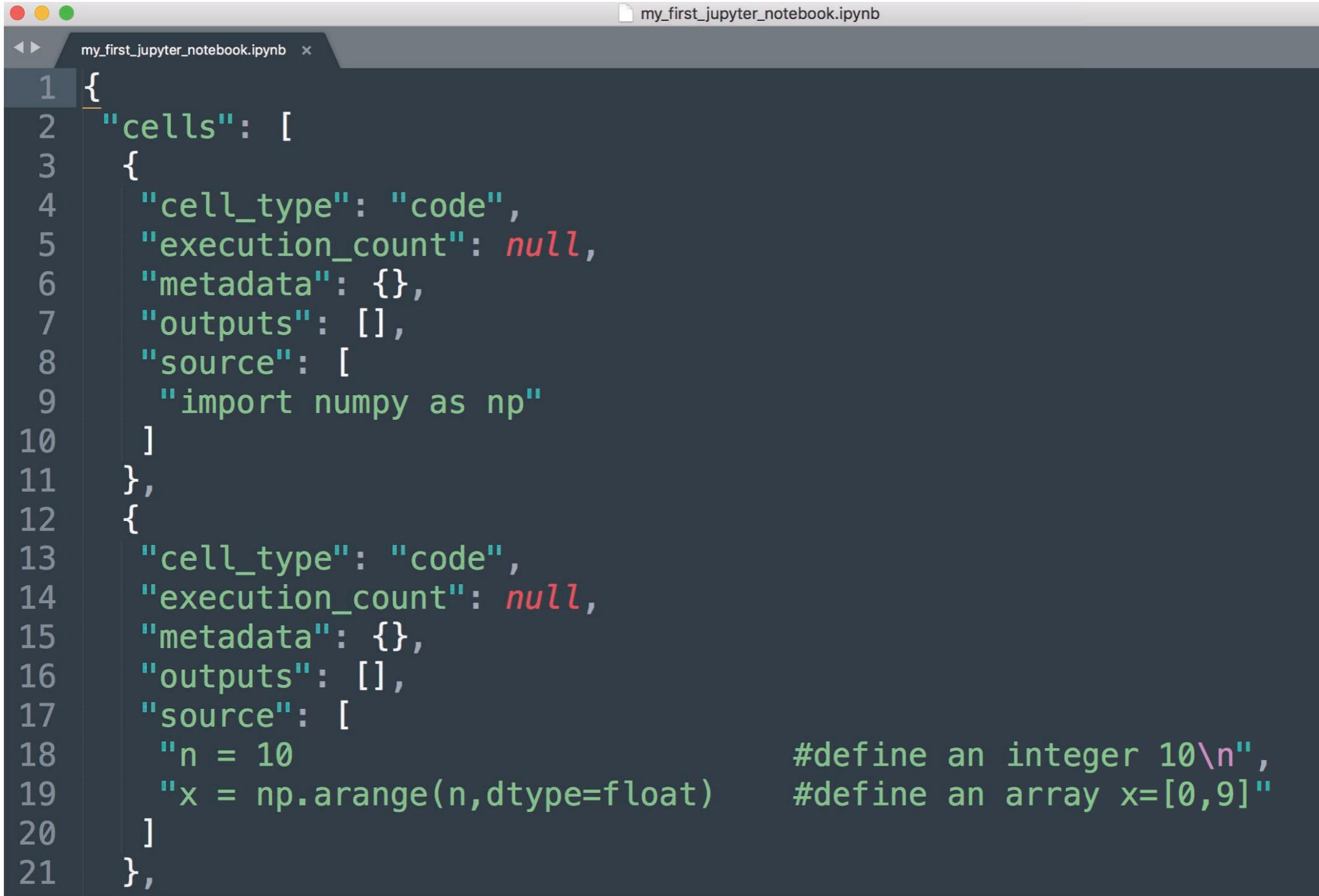


From Jupyter Home: Select Running Notebook and Shutdown



Jupyter Notebooks are JSON

Keep them clean – clear output



The screenshot shows a Jupyter Notebook interface with a dark theme. The title bar says "my_first_jupyter_notebook.ipynb". The left sidebar shows the file path "my_first_jupyter_notebook.ipynb". The main area displays the JSON structure of the notebook:

```
1 {  
2   "cells": [  
3     {  
4       "cell_type": "code",  
5       "execution_count": null,  
6       "metadata": {},  
7       "outputs": [],  
8       "source": [  
9         "import numpy as np"  
10      ]  
11    },  
12    {  
13      "cell_type": "code",  
14      "execution_count": null,  
15      "metadata": {},  
16      "outputs": [],  
17      "source": [  
18        "n = 10  
19        x = np.arange(n,dtype=float)  
20      ]  
21    }]
```

The source code in the second cell is annotated with comments: "#define an integer 10\n", "#define an array x=[0,9]".



Matplotlib Basics



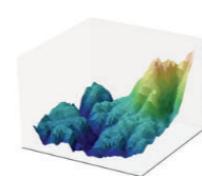
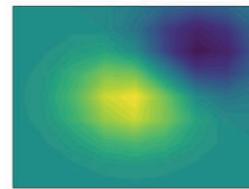
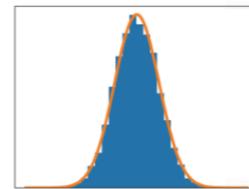
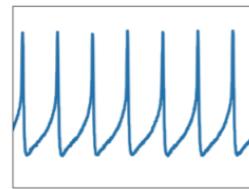
Version 3.0.0

Fork me on GitHub

[home](#) | [examples](#) | [tutorials](#) | [API](#) | [docs](#) »

[modules](#) | [index](#)

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits.



Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc., with just a few lines of code. For examples, see the [sample plots](#) and [thumbnail gallery](#).

For simple plotting the `pyplot` module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

Installation

Visit the [Matplotlib installation instructions](#).

Documentation

Quick search

Go

Matplotlib 3.0 is Python 3 only.

For Python 2 support,
Matplotlib 2.2.x will be
continued as a LTS release
and updated with bugfixes
until January 1, 2020.

[Support Matplotlib](#)

[Support NumFOCUS](#)

matplotlib.org



UC SANTA CRUZ

Matplotlib is on GitHub

The screenshot shows the GitHub repository page for `matplotlib / matplotlib`. The top navigation bar includes links for Pull requests, Issues, Marketplace, and Explore. The repository statistics are displayed: 27,185 commits, 8 branches, 74 releases, and 758 contributors. The commit history lists several recent changes, including a merge pull request from `anntzer/tick-visibility`, updates to CircleCI and GitHub templates, and whitespace updates to documentation. The footer features the UC Santa Cruz logo.

matplotlib / matplotlib

Pull requests Issues Marketplace Explore

Watch 518 Star 8,006 Fork 3,680

Code Issues 1,157 Pull requests 282 Projects 5 Wiki Insights

matplotlib: plotting with Python <http://matplotlib.org/>

27,185 commits 8 branches 74 releases 758 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

jklymak Merge pull request #10088 from anntzer/tick-visibility ... Latest commit 13629a4 8 hours ago

.circleci Add CircleCI cache of user fonts. 3 months ago

.github BLD: update PR template for flake8 a month ago

LICENSE MNT/DOC: whitespace + updated example heading a year ago

ci Simplify travis setup a bit. 5 months ago



Using pip to install matplotlib

Mac OSX / Linux:

```
$ pip3 install matplotlib
```

Windows:

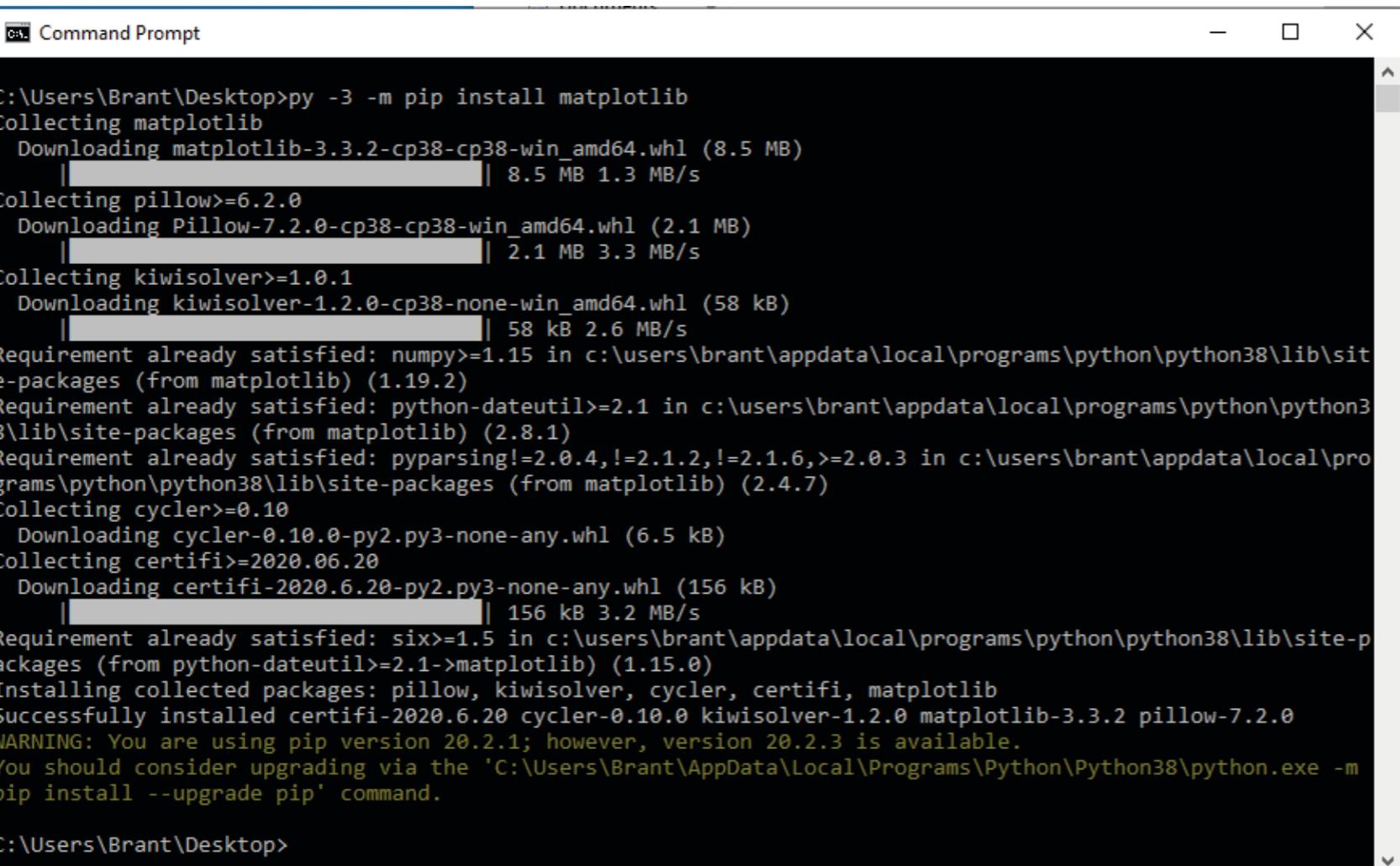
```
$ py -3 -m pip install matplotlib
```



Installing matplotlib on Windows

In a command prompt:

```
$ py -3 -m pip install matplotlib
```



```
Command Prompt

C:\Users\Brant\Desktop>py -3 -m pip install matplotlib
Collecting matplotlib
  Downloading matplotlib-3.3.2-cp38-cp38-win_amd64.whl (8.5 MB)
    ||██████████| 8.5 MB 1.3 MB/s
Collecting pillow>=6.2.0
  Downloading Pillow-7.2.0-cp38-cp38-win_amd64.whl (2.1 MB)
    ||██████████| 2.1 MB 3.3 MB/s
Collecting kiwisolver>=1.0.1
  Downloading kiwisolver-1.2.0-cp38-none-win_amd64.whl (58 kB)
    ||██████████| 58 kB 2.6 MB/s
Requirement already satisfied: numpy>=1.15 in c:\users\brant\appdata\local\programs\python\python38\lib\site-packages (from matplotlib) (1.19.2)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\brant\appdata\local\programs\python\python38\lib\site-packages (from matplotlib) (2.8.1)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in c:\users\brant\appdata\local\programs\python\python38\lib\site-packages (from matplotlib) (2.4.7)
Collecting cycler>=0.10
  Downloading cycler-0.10.0-py2.py3-none-any.whl (6.5 kB)
Collecting certifi>=2020.06.20
  Downloading certifi-2020.6.20-py2.py3-none-any.whl (156 kB)
    ||██████████| 156 kB 3.2 MB/s
Requirement already satisfied: six>=1.5 in c:\users\brant\appdata\local\programs\python\python38\lib\site-packages (from python-dateutil>=2.1->matplotlib) (1.15.0)
Installing collected packages: pillow, kiwisolver, cycler, certifi, matplotlib
Successfully installed certifi-2020.6.20 cycler-0.10.0 kiwisolver-1.2.0 matplotlib-3.3.2 pillow-7.2.0
WARNING: You are using pip version 20.2.1; however, version 20.2.3 is available.
You should consider upgrading via the 'C:\Users\Brant\AppData\Local\Programs\Python\Python38\python.exe -m pip install --upgrade pip' command.

C:\Users\Brant\Desktop>
```

A bunch of packages will likely get installed, but at the end you shouldn't have any errors.



matplotlib.pyplot



Fork me on GitHub

[home](#) | [examples](#) | [tutorials](#) | [API](#) | [docs](#) » [The Matplotlib API](#) » [Pyplot function overview](#) »

[previous](#) | [next](#) | [modules](#) | [index](#)

matplotlib.pyplot

`matplotlib.pyplot` is a state-based interface to matplotlib. It provides a MATLAB-like way of plotting.

`pyplot` is mainly intended for interactive plots and simple cases of programmatic plot generation:

```
import numpy as np
import matplotlib.pyplot as plt

x = np.arange(0, 5, 0.1)
y = np.sin(x)
plt.plot(x, y)
```

The object-oriented API is recommended for more complex plots.

Functions

`acorr(x, *[data])` Plot the autocorrelation of `x`.

`angle_spectrum(x[, Fs, Fc, window, pad_to, ...])` Plot the angle spectrum.

`annotate(text, xy, *args, **kwargs)` Annotate the point `xy` with text `s`.

`arrow(x, y, dx, dy, **kwargs)` Add an arrow to the axes.

`autoscale([enable, axis, tight])` Autoscale the axis view to the data (toggle).

Quick search

 Go

Table Of Contents

`matplotlib.pyplot`

- [Functions](#)

Related Topics

Documentation overview

▪ [The Matplotlib API](#)

- [Pyplot function overview](#)
- [Previous: Pyplot function overview](#)
- [Next: `matplotlib.pyplot.acorr`](#)

[Show Page Source](#)

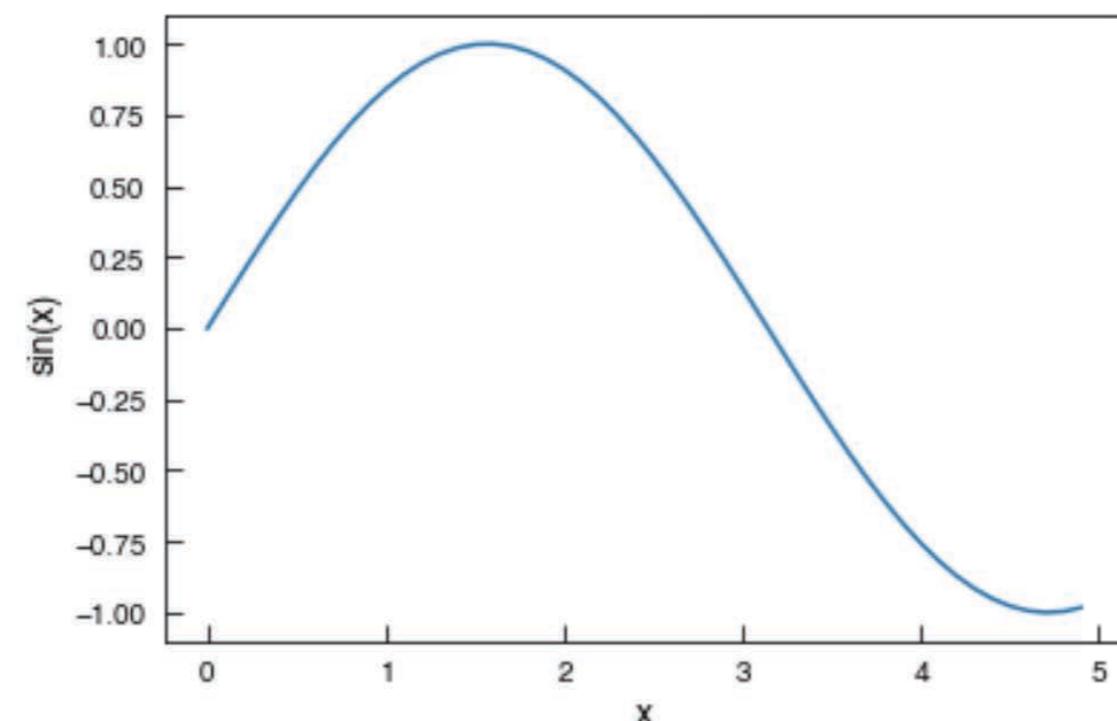


UC SANTA CRUZ



```
In [1]: %matplotlib inline  
import numpy as np  
import matplotlib.pyplot as plt
```

```
In [3]: x = np.arange(0, 5, 0.1) #create x=[0..5] in 0.1 increments  
y = np.sin(x) # y = sin(x)  
plt.plot(x,y) # make a plot  
plt.xlabel('x') # label x axis  
plt.ylabel('sin(x)') # label y axis  
plt.show() # show the plot on the screen
```



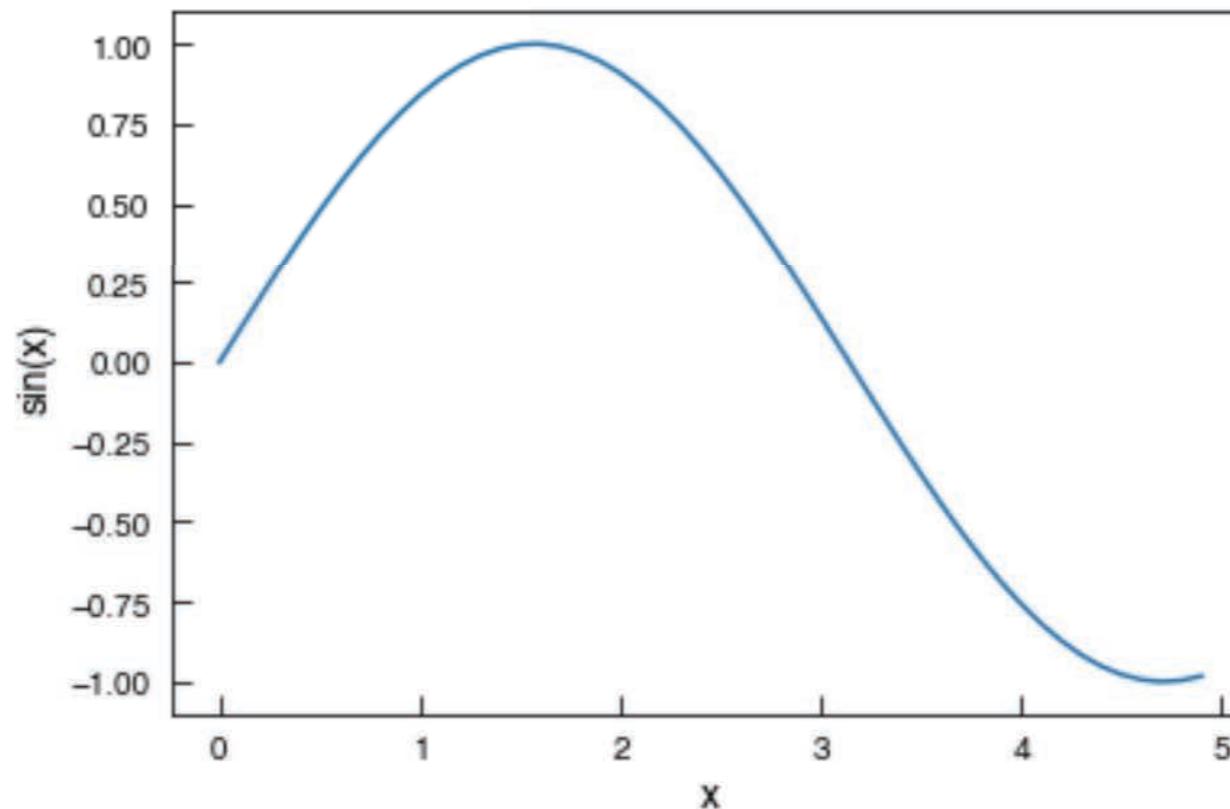
Matplotlib.pyplot



File Edit View Insert Cell Kernel Help



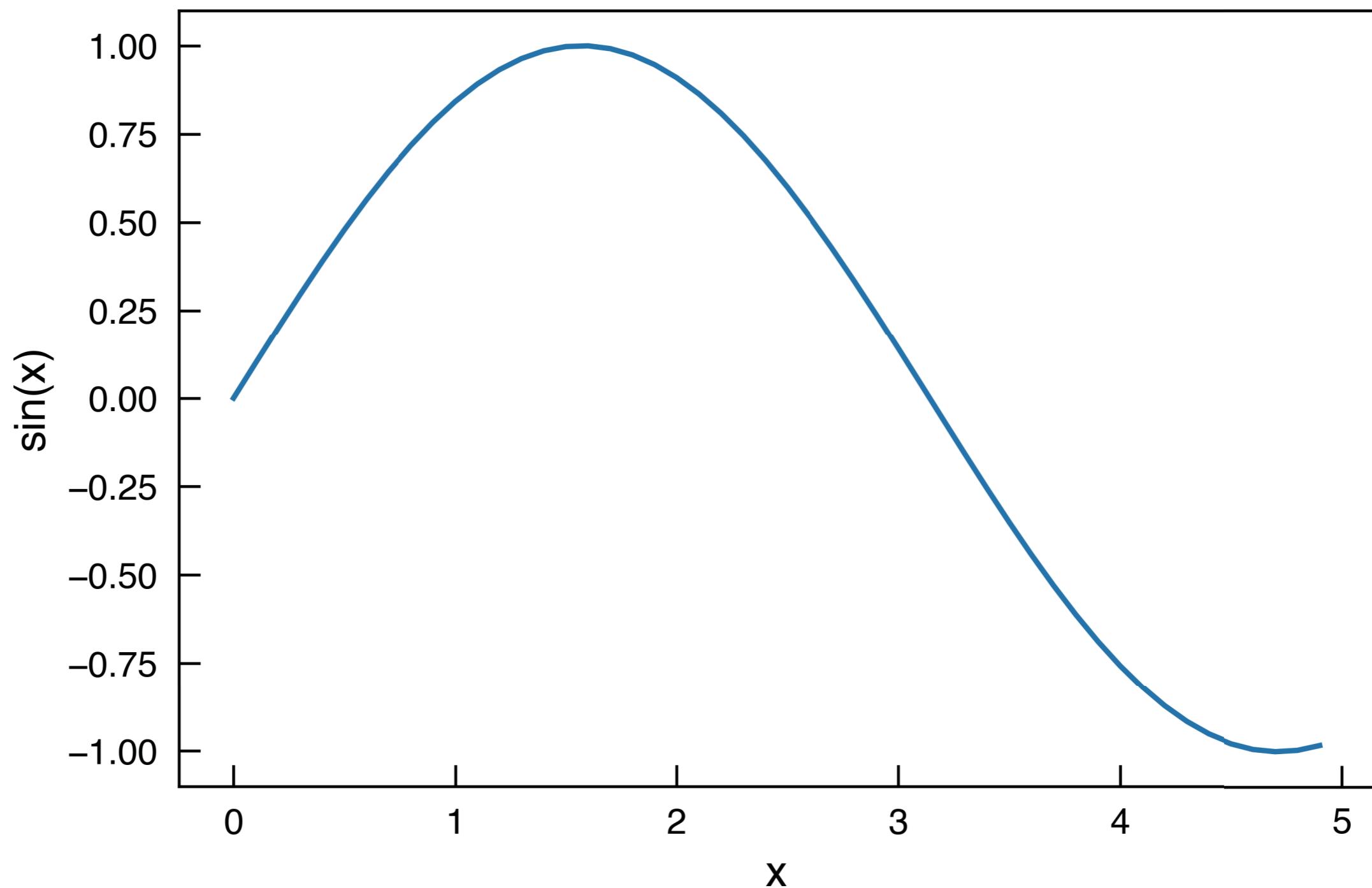
```
In [5]: x = np.arange(0, 5, 0.1) #create x=[0..5] in 0.1 increments
y = np.sin(x)
plt.plot(x,y)
plt.xlabel('x')
plt.ylabel('sin(x)')
plt.savefig('sinx.png',bbox_inches="tight",dpi=600)
```



plt.savefig()



UC SANTA CRUZ



`plt.savefig()`



UC SANTA CRUZ

Clear All Output, Then Save

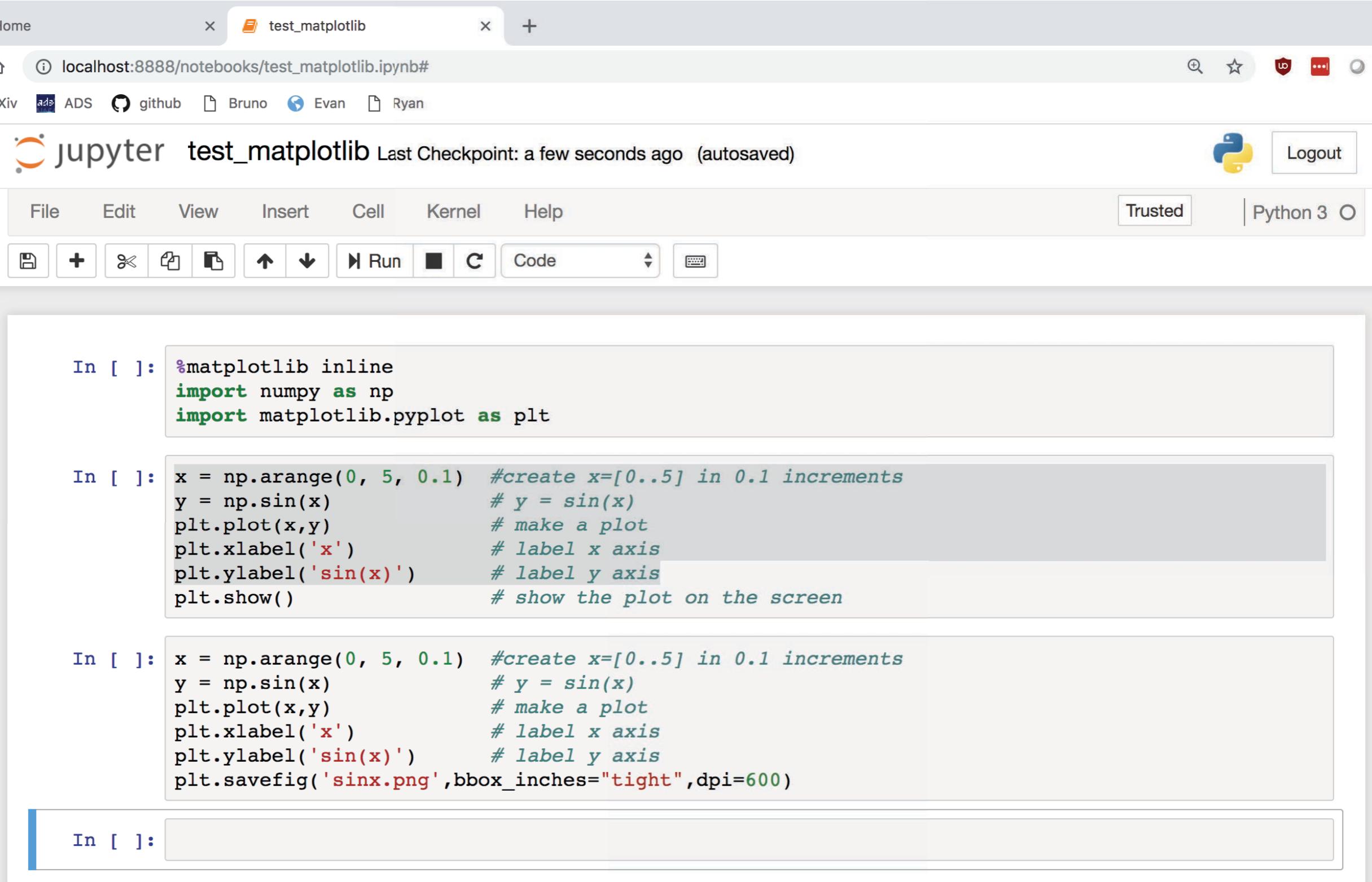
A screenshot of a Jupyter Notebook interface running in a web browser. The browser's address bar shows the URL `localhost:8888/notebooks/my_first_jupyter_notebook.ipynb`. The notebook title is `my_first_jupyter_notebook`. The status bar indicates "Last Checkpoint: 2 minutes ago (autosaved)". The top menu bar includes File, Edit, View, Insert, Cell, Kernel, and Help. The Cell menu is currently open, showing options: Run Cells, Run Cells and Select Below, Run Cells and Insert Below, Run All, Run All Above, Run All Below, Cell Type, Current Outputs, and All Output. The "All Output" option has a submenu with three items: Toggle, Toggle Scrolling, and Clear. The "Clear" option is highlighted with a blue selection bar. On the left, there are three code cells:

- In [1]: `import numpy`
- In [3]: `n = 10`
`x = np.arange`
- In []: `print(x)`

The output area on the right shows the results of the second cell:
`ne an integer 10`
`ne an array x=[0, 9]`



Clear All Output, Then Save



The screenshot shows a Jupyter Notebook interface running in a web browser. The title bar indicates the notebook is titled "test_matplotlib". The toolbar includes standard browser controls (Home, Back, Forward, Refresh) and Jupyter-specific buttons for file operations (Save, New, Cut, Copy, Paste, Run, Cell, Kernel, Help), code editor, and cell selection.

The notebook contains three code cells:

```
In [ ]: %matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
```

```
In [ ]: x = np.arange(0, 5, 0.1)      #create x=[0..5] in 0.1 increments
y = np.sin(x)                      # y = sin(x)
plt.plot(x,y)                      # make a plot
plt.xlabel('x')                    # label x axis
plt.ylabel('sin(x)')               # label y axis
plt.show()                          # show the plot on the screen
```

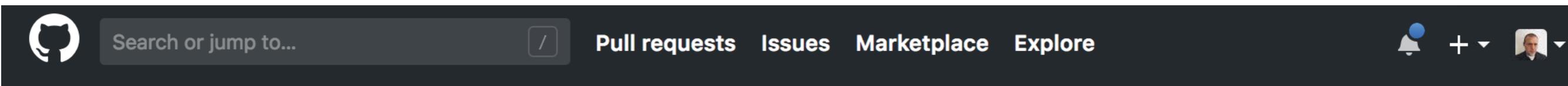
```
In [ ]: x = np.arange(0, 5, 0.1)      #create x=[0..5] in 0.1 increments
y = np.sin(x)                      # y = sin(x)
plt.plot(x,y)                      # make a plot
plt.xlabel('x')                    # label x axis
plt.ylabel('sin(x)')               # label y axis
plt.savefig('sinx.png',bbox_inches="tight",dpi=600)
```

The bottom cell is currently active, indicated by a blue vertical bar on its left.



Save Your Work

Make a GitHub project “astr-119-session-5”, and commit the programs `my_first_jupyter_notebook.ipynb` and `test_matplotlib.ipynb` you made today.



Create a new repository

A repository contains all the files for your project, including the revision history.

Owner Repository name

 brantr / astr-119-session-5 

Great repository names are short and memorable. Need inspiration? How about [fantastic-spork](#).

Description (optional)

We learned a new trick! -- Jupyter notebooks.

 Public

Anyone can see this repository. You choose who can commit.

 Private

You choose who can see and commit to this repository.



UC SANTA CRUZ

Save Your Work

Make a GitHub project “astr-119-session-5”, and commit the programs `my_first_jupyter_notebook.ipynb` and `test_matplotlib.ipynb` you made today.

The screenshot shows a GitHub repository page for the user 'brantr' named 'astr-119-session-5'. The repository has 2 commits, 1 branch, 0 releases, 1 contributor, and is licensed under MIT. The latest commit was made just now by 'brantr' with the message 'First two Jupyter notebooks for ASTR 119.' The repository also contains files for 'LICENSE', 'README.md', 'my_first_jupyter_notebook.ipynb', and 'test_matplotlib.ipynb'.

brantr / **astr-119-session-5**

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

We learned a new trick! -- Jupyter notebooks. Edit

Manage topics

2 commits 1 branch 0 releases 1 contributor MIT

Branch: master ▾ New pull request Create new file Upload files Find file Clone or download ▾

File	Commit Message	Time
LICENSE	Initial commit	43 seconds ago
README.md	Initial commit	43 seconds ago
my_first_jupyter_notebook.ipynb	First two Jupyter notebooks for ASTR 119.	just now
test_matplotlib.ipynb	First two Jupyter notebooks for ASTR 119.	just now

