Python @ SIAM CSE 2015

Jon Woodring, Aron Ahmadia, Andy Bauer, Joseph Cottam & Andy Terrel

March 15, 2015

Getting Started

Three paths:

1. Watch the slides
2. Interactive with a virtual machine
3. Download the git repository and use Anaconda/

Path 1: Follow along with the slides

Getting Started

Three paths:

1. Watch the slides
2. Interactive with a virtual machine
3. Download the git repository and use Anaconda/

Path 1: Follow along with the slides

Open in your web browser:

https://github.com/JosephCottam/PythonSIAM2014

Scroll down on the page.

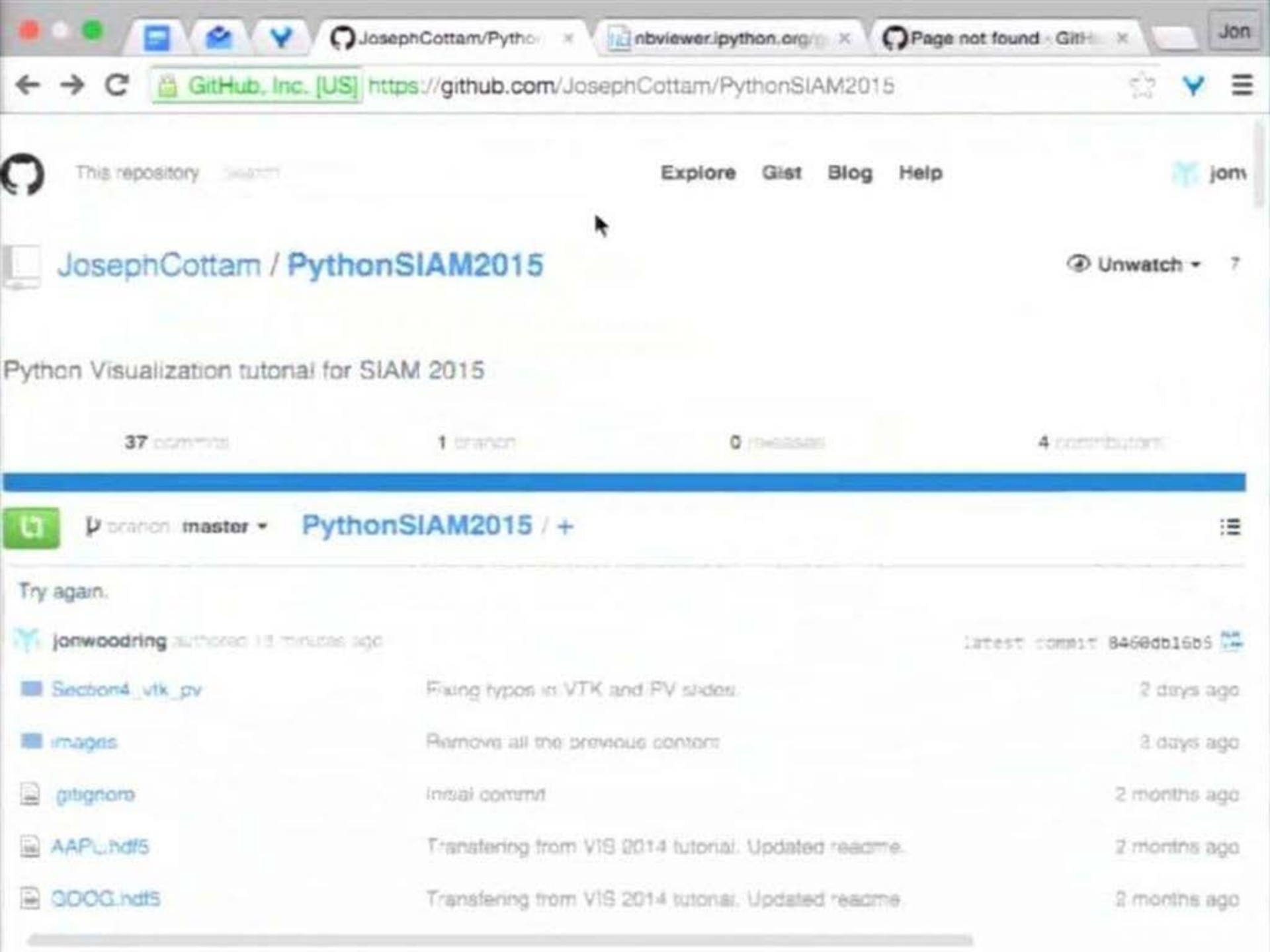
Click the "Section X" links to the notebooks and slides as needed.

If you have section 0 open, here are the links to the static materials:

- Section 0 Introduction
  - Additional ipython links:
    - https://github.com/barbagroup/CFDPython

Find code, projects, and people on GitHub:

Search


JosephCottam/Pytho
nbviewer.ipython.org/g
Page not found · GitH
Jon
GitHub, Inc. [US] https://github.com/JosephCottam/PythonSIAM2015
This repository
Explore
Gist
Blog
Help
jonw
JosephCottam / PythonSIAM2015
Unwatch
Python Visualization tutorial for SIAM 2015
37
1
0
4
branch: master
PythonSIAM2015 / +
Try again.
jonwoodring
Section4_vtk_pv
Fixing typos in VTK and PV slides
2 days ago
images
Removes all the previous content
2 days ago
gitignore
Initial commit
2 months ago
AAPL.hdf5
Transfering from VIS 2014 tutorial. Updated readme
2 months ago
GOOG.hdf5
Transfering from VIS 2014 tutorial. Updated readme
2 months ago

Introduction to IPython Notebook

Why do I use IPython notebook:

- Interactive like a REPL
- Durrable artifacts in cells
- Shareable like a document...with built-in annotations!

A motivating example...

In [7]:
```
from __future__ import division, print_function
import numpy as np
import pandas as pd
```

In [8]:
```
!ls
```

```
881.png                                Section3_bokeh.ipynb                heat
_3-2.png
AAPL.hdf5                              Section4_vtk.ipynb                  inte
restingBugs.jpg
```

- [redacted]
- Apache Libcloud

Path 2: Follow along interactively ¶

Our setup involves using a virtual machine (VM) to ease the process because everything will be prepackaged right from the start, and launched after you start the VM. We will have copies of the VM on USB drives, or you can download the VM directly from the web.

The virtual machine is available at: http://goo.gl/V6Ryk0

Virtualbox software download link is at:
https://www.virtualbox.org/wiki/Downloads

This link is available for clicking (so you don't have to type the long link) by going to
https://github.com/JosephCottam/PythonSIAM2014

We will provide VirtualBox to be able to run the virtual machine, on-site at SIAM, (or you can download it directly from Oracle) but other virtualization systems (VMWare, libvirt/qemu, etc.) should be able to run the images. The virtual machine is packaged in a OVA (Open Virtualization Format) for easy installation on your machine.

"We are running a 64-bit Linux on the VM, which should work even if you are running a 32-bit OS, as long as you have a modern 64-bit processor."

Open the installer and go through the default installation to install VirtualBox on your computer.

Step Two: Open the Virtual Machine

Copy pyvis_siam_2015.ova from USB drives available on-site at SIAM or download it from the link below.

The virtual machine is available at: http://goo.gl/V6Rvk0

Start VirtualBox, and in the main menu, go to "File->Import Appliance"

Open the "pyvis_siam_2015.ova" that is provided. Continue through the default options for the virtual machine.

This will unpack and install the virtual machine on your computer.

Step Three: Launch the Virtual Machine

In VirtualBox, in the virtual machine lists, open the new virtual machine. This will start Arch Linux in a virtual machine on your computer.

It will automatically launch into an graphical session with two open terminals and a ipython notebook session.

Step Three: Launch the Virtual Machine

In VirtualBox, in the virtual machine lists, open the new virtual machine. This will start Arch Linux in a virtual machine on your computer.

It will automatically launch into an graphical session with two open terminals and a ipython notebook session.

In case you happen to need it, the default user is "pyvis" with a password of "pyvis". It has sudo access, and the root password is "pyvis".

Step Four: You're Ready!

If all went well, a window should have launched in the browser on the VM, showing the ipython notebook interface.

Path 3. Install on your computer without VM

Alternatively, if you can't get the VM to work, you can download Anaconda, ParaView, VTK & the materials:

Step Four: You're Ready!

If all went well, a window should have launched in the browser on the VM, showing the ipython notebook interface.

Path 3. Install on your computer without VM

Alternatively, if you can't get the VM to work, you can download Anaconda, ParaView, VTK & the materials:

Tutorial Materials -- https://github.com/JosephCottam/PythonSIAM2015/archive/master.zip

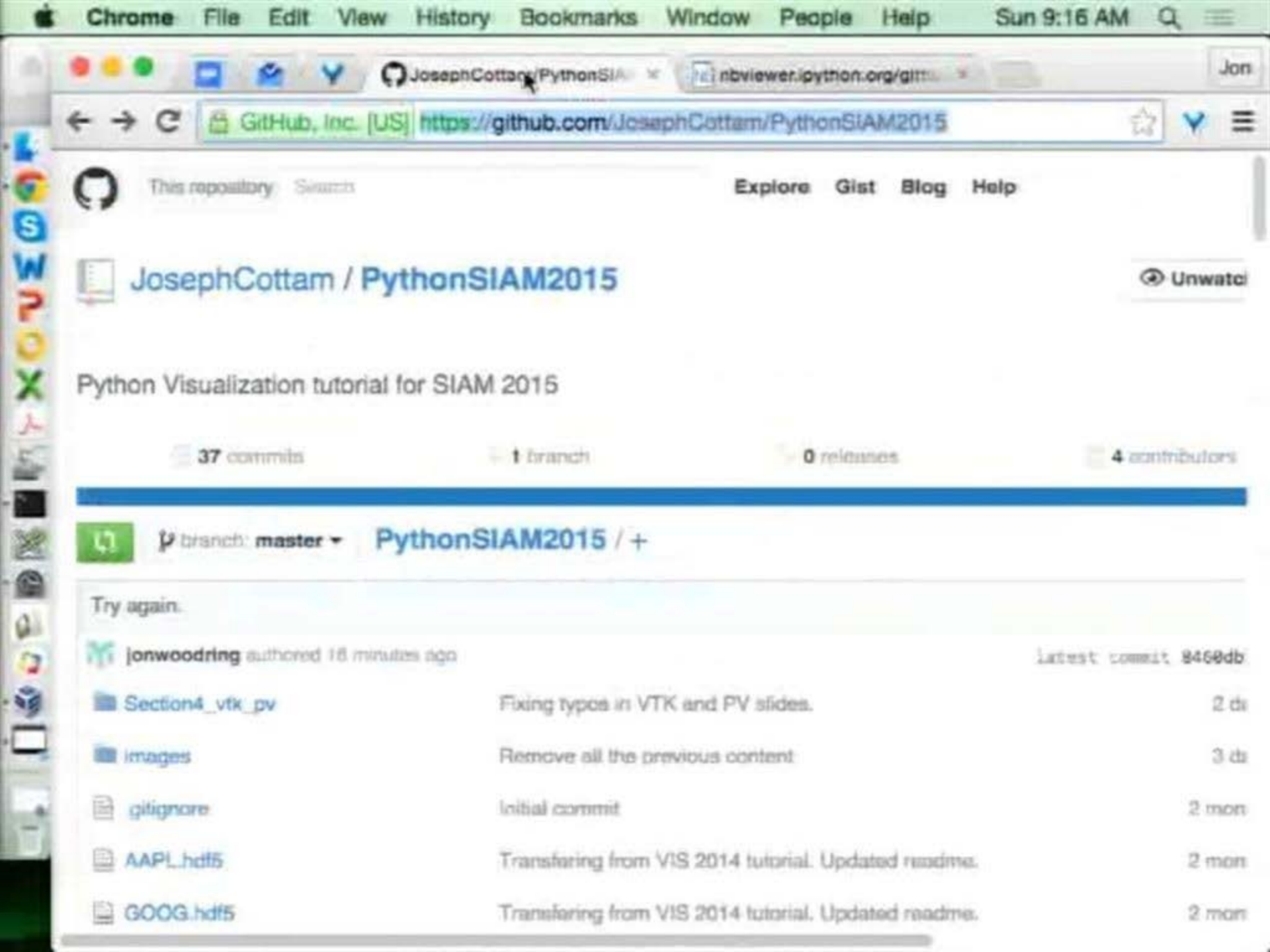
Anaconda Python -- http://continuum.io/downloads

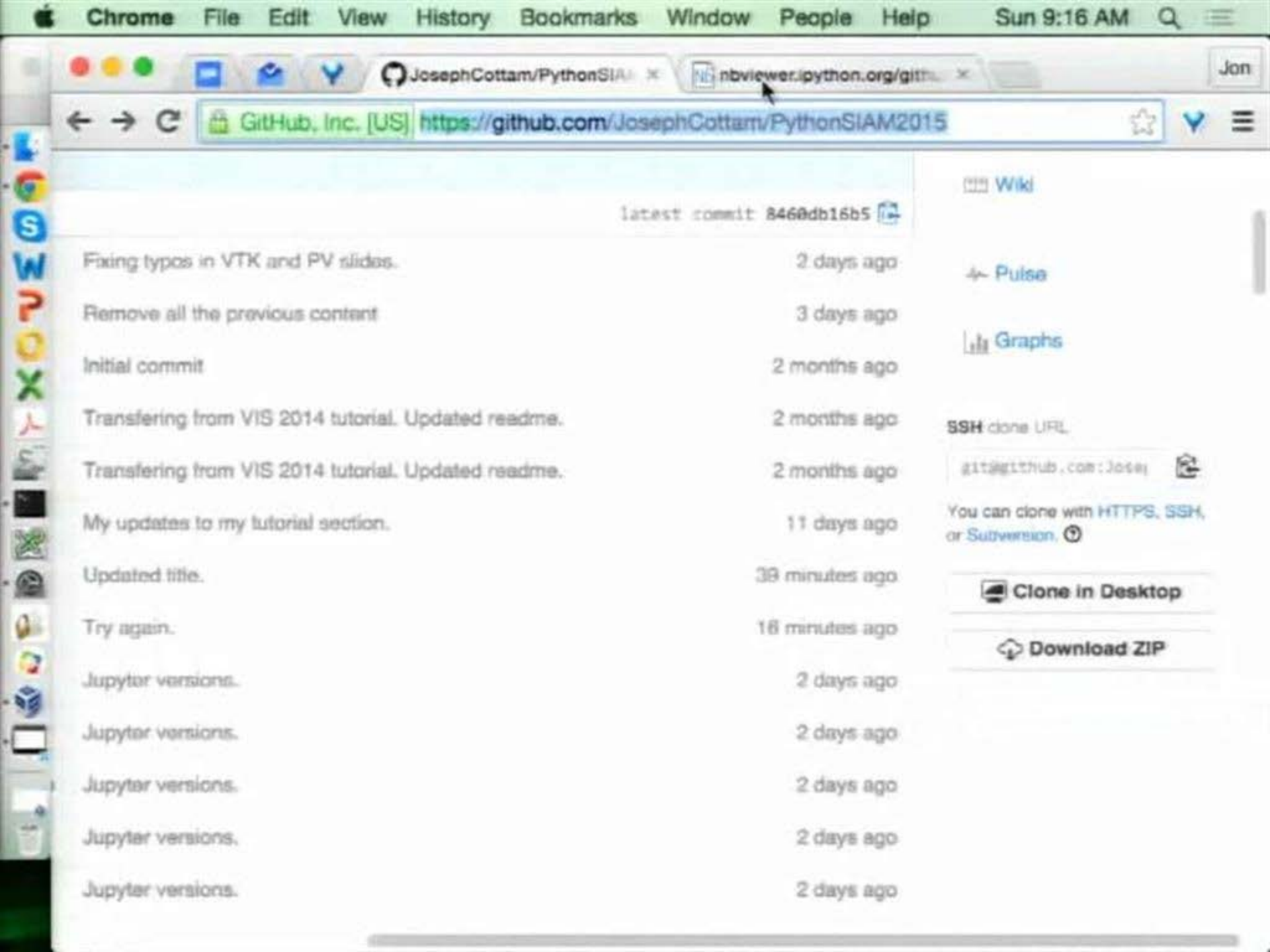
ParaView -- http://www.paraview.org/download/

VTK -- http://www.vtk.org/VTK/resources/software.html

3 Reasons why I like Python

1. It's a fun language


Chrome File Edit View History Bookmarks Window People Help Sun 9:16 AM
GitHub, Inc. [US] https://github.com/JosephCottam/PythonSIAM2015
Jon
This repository
Explore Gist Blog Help
JosephCottam / PythonSIAM2015
Unwatch
Python Visualization tutorial for SIAM 2015
37 commits 1 branch 0 releases 4 contributors
branch: master PythonSIAM2015 / +
Try again.
jonwoodring authored 16 minutes ago
Latest commit 8460db
Section4_vtk_pv Fixing typos in VTK and PV slides. 2 da
images Remove all the previous content 3 da
.gitignore Initial commit 2 mon
AAPL.hdf5 Transfering from VIS 2014 tutorial. Updated readme. 2 mon
GOOG.hdf5 Transfering from VIS 2014 tutorial. Updated readme. 2 mon


Chrome File Edit View History Bookmarks Window People Help Sun 9:16 AM
JosephCottam/PythonSIA
nbviewer.ipython.org/gith
Jon
GitHub, Inc. [US] https://github.com/JosephCottam/PythonSIAM2015
latest commit 8460db16b5
Fixing typos in VTK and PV slides. 2 days ago
Remove all the previous content 3 days ago
Initial commit 2 months ago
Transfering from VIS 2014 tutorial. Updated readme. 2 months ago
Transfering from VIS 2014 tutorial. Updated readme. 2 months ago
My updates to my tutorial section. 11 days ago
Updated title. 39 minutes ago
Try again. 18 minutes ago
Jupyter versions. 2 days ago
Jupyter versions. 2 days ago
Jupyter versions. 2 days ago
Jupyter versions. 2 days ago
Jupyter versions. 2 days ago
Wiki
Pulse
Graphs
SSH clone URL
git@github.com:Jose
You can clone with HTTPS, SSH, or Subversion.
Clone in Desktop
Download ZIP

If all went well, a window should have launched in the browser on the VM, showing the ipython notebook interface.

Path 3. Install on your computer without VM

Alternatively, if you can't get the VM to work, you can download Anaconda, ParaView, VTK & the materials:

Tutorial Materials -- https://github.com/JosephCottam/PythonSIAM2015/archive/master.zip

Anaconda Python -- http://continuum.io/downloads

ParaView -- http://www.paraview.org/download/

VTK -- http://www.vtk.org/VTK/resources/software.html

3 Reasons why I like Python

1. It's a fun language

o do anything. It's kind of functional, it's

1. It's a fun language

It is so easy to do anything. It's kind of functional, it's kind of procedural, it's all around fun.

Many of you here may already use Matlab and R, and Python is another tool in that vein. Python may not replace those tools for you, but stick around, maybe something here piques your interest.

I personally prefer Python to both of those or I wouldn't be teaching this tutorial :)

2. Rapid prototyping

I have written volume renderers, high-dimensional

2. Rapid prototyping

I have written volume renderers, high-dimensional visualizations, analysis tools, web servers, benchmarks, parallel processing, data manipulation, demonstrations, etc. all in Python.

I'm a big fan of dynamic typing, rather than static typing, because I'm a "bottom-up developer" & "experimental computer scientist." Python and ipython/jupyter fit my workflow really well. ¶

Premature optimization is the downfall of productivity. There is no good reason to start with C/C++ when 95% of the time in analytics, the bottleneck is I/O. If you need to make it go fast, find your bottlenecks, and optimize LATER not at the start.

3. Batteries are included

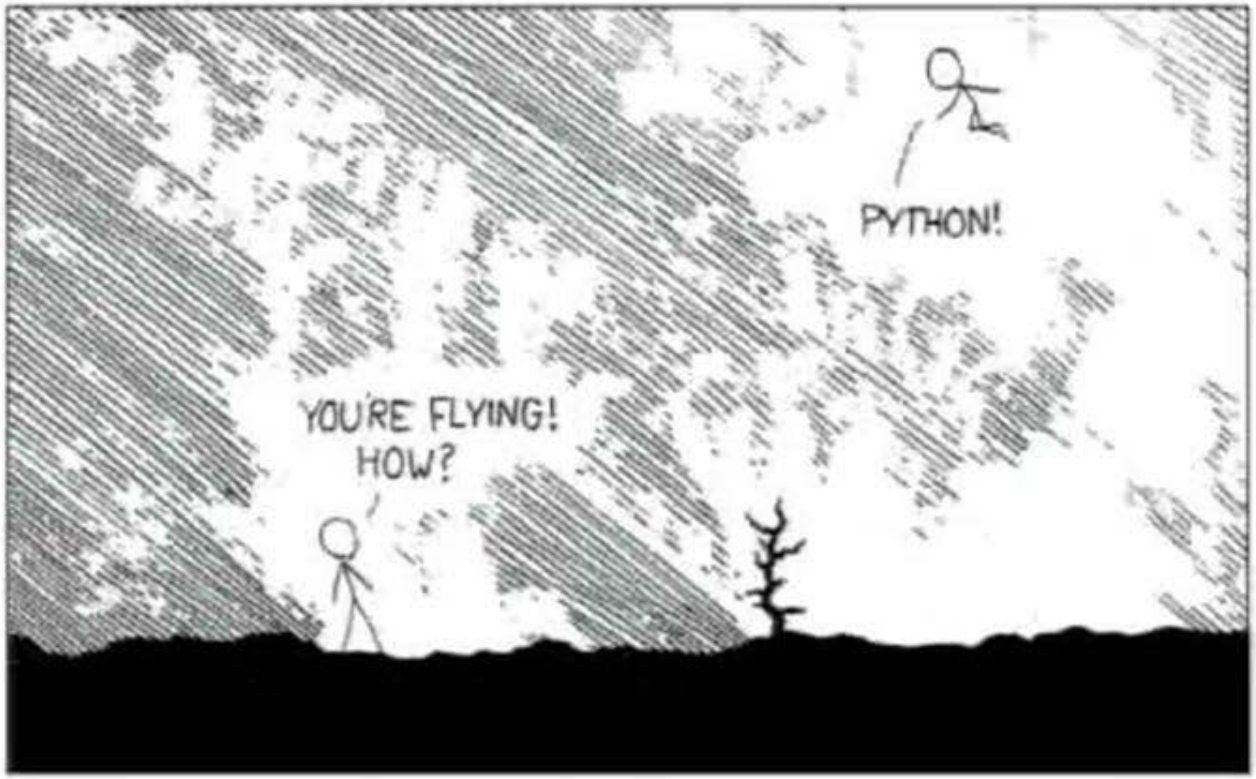
Chrome File Edit View History Bookmarks Window People Help Sun 9:18 AM

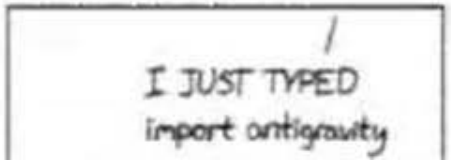
JosephCottam/PythonSIA nbviewer.ipython.org/gith Jon

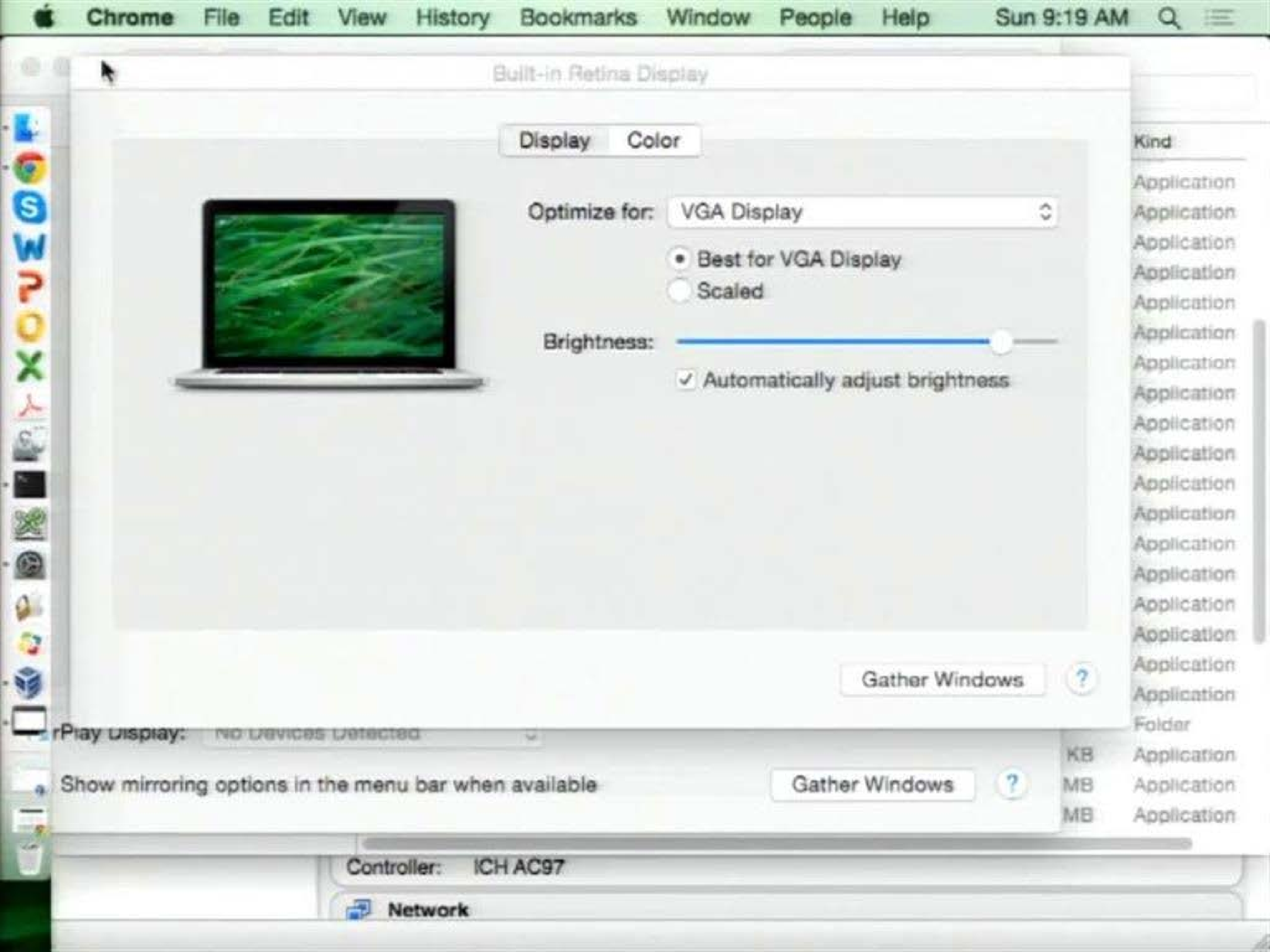
nbviewer.ipython.org/github/JosephCottam/PythonSIAM2015/blob/master/Section0_...

nbviewer FAQ IPython Jupyter

XKCD: import antigravity

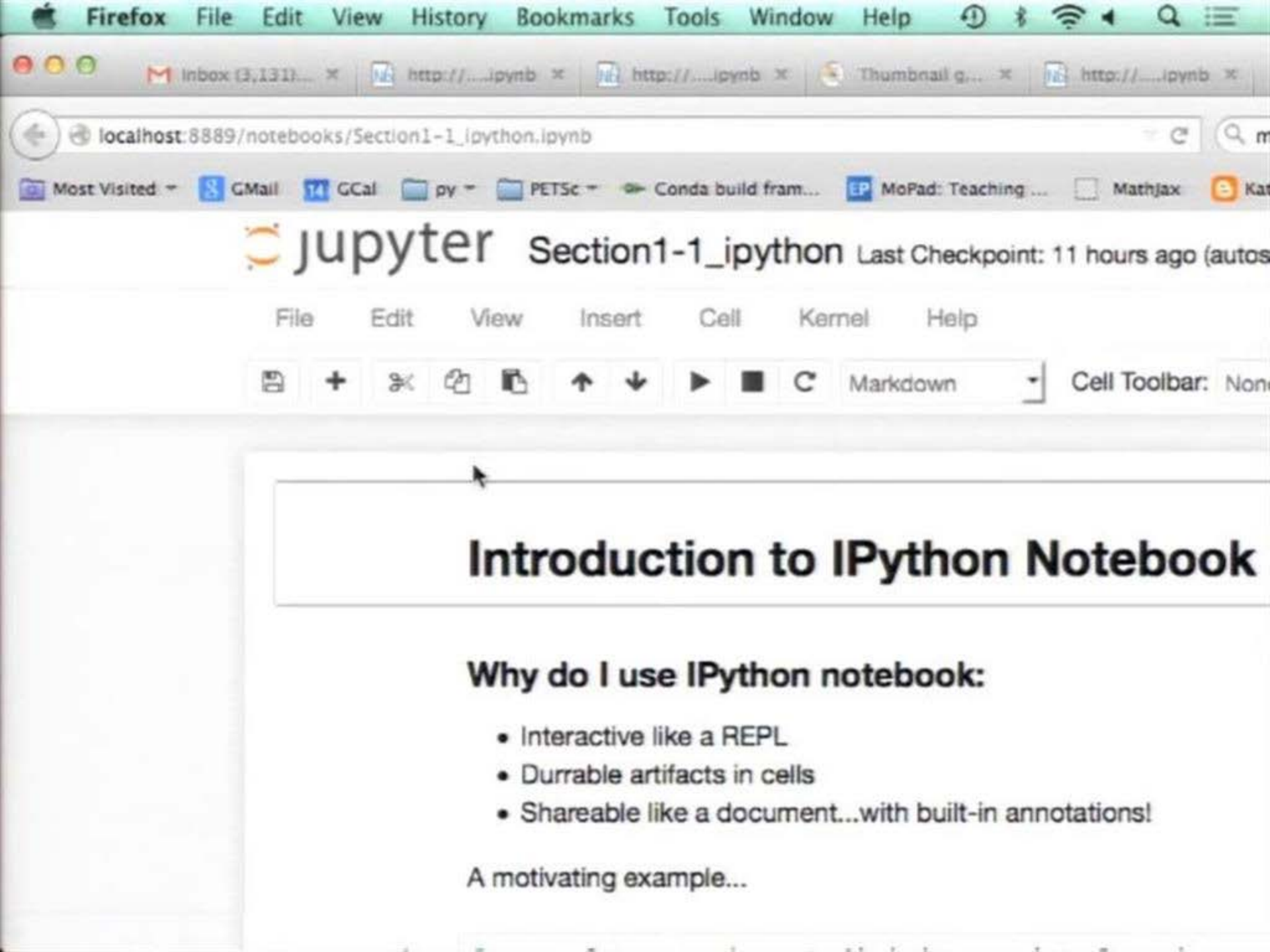


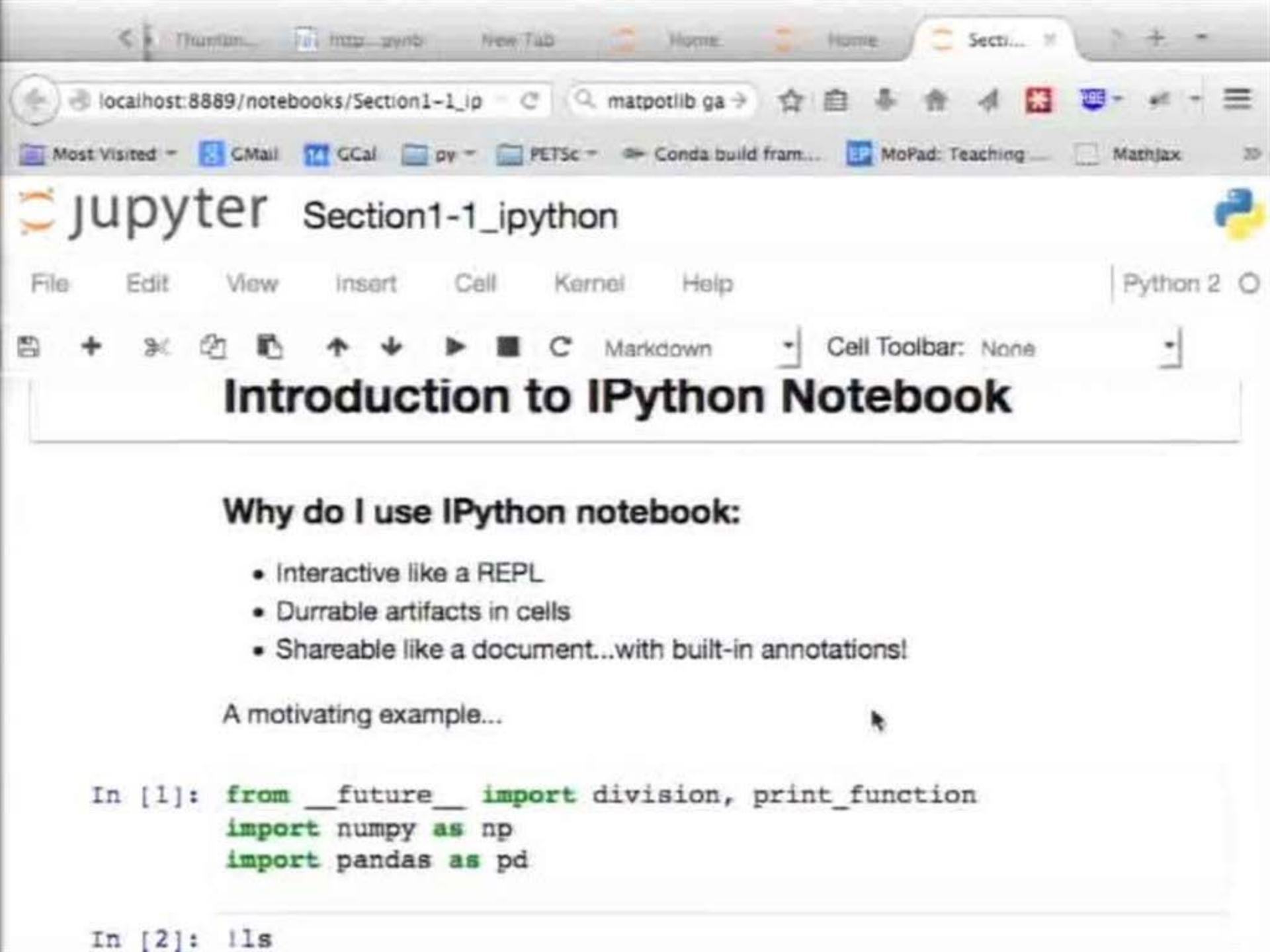


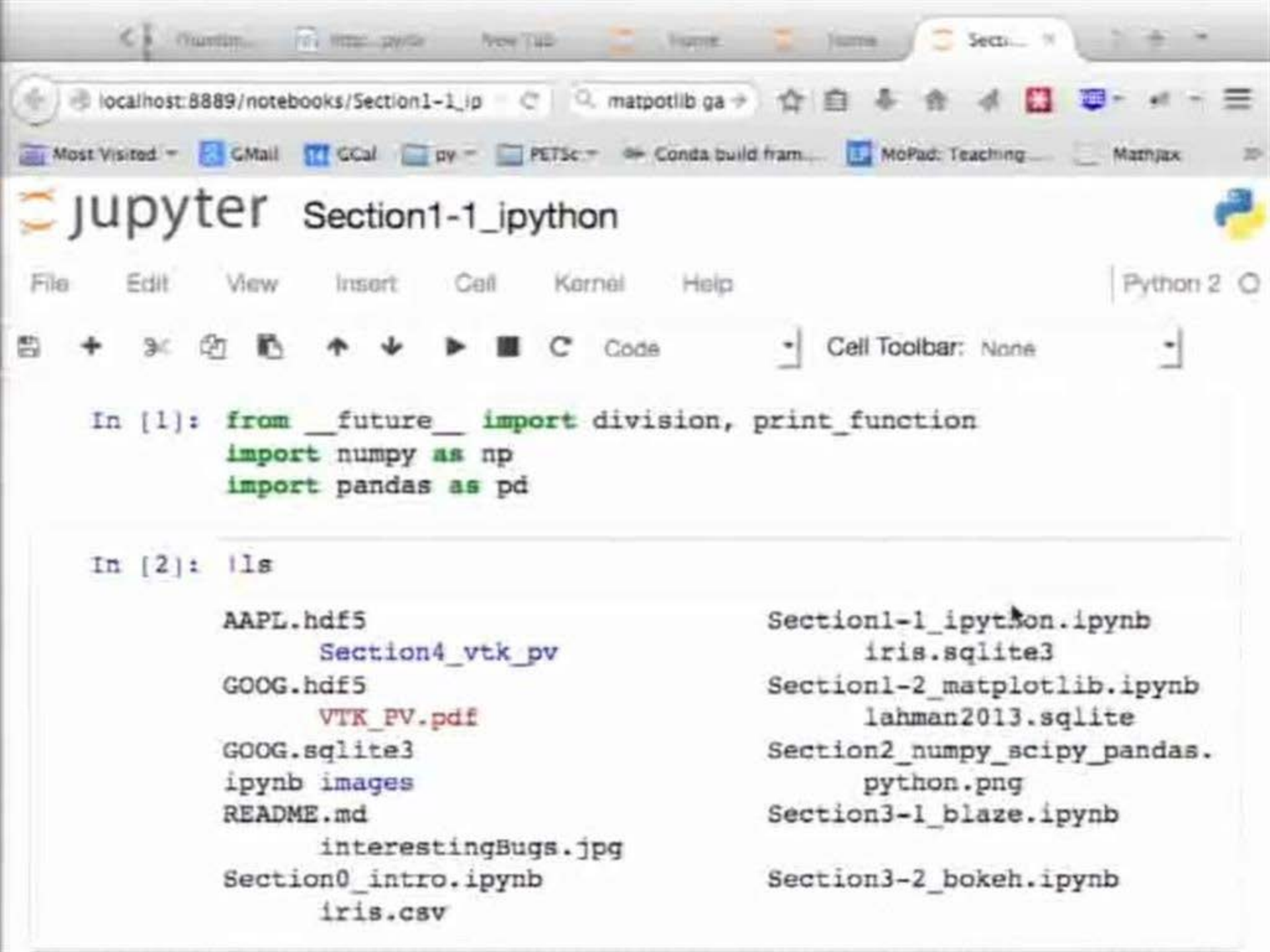


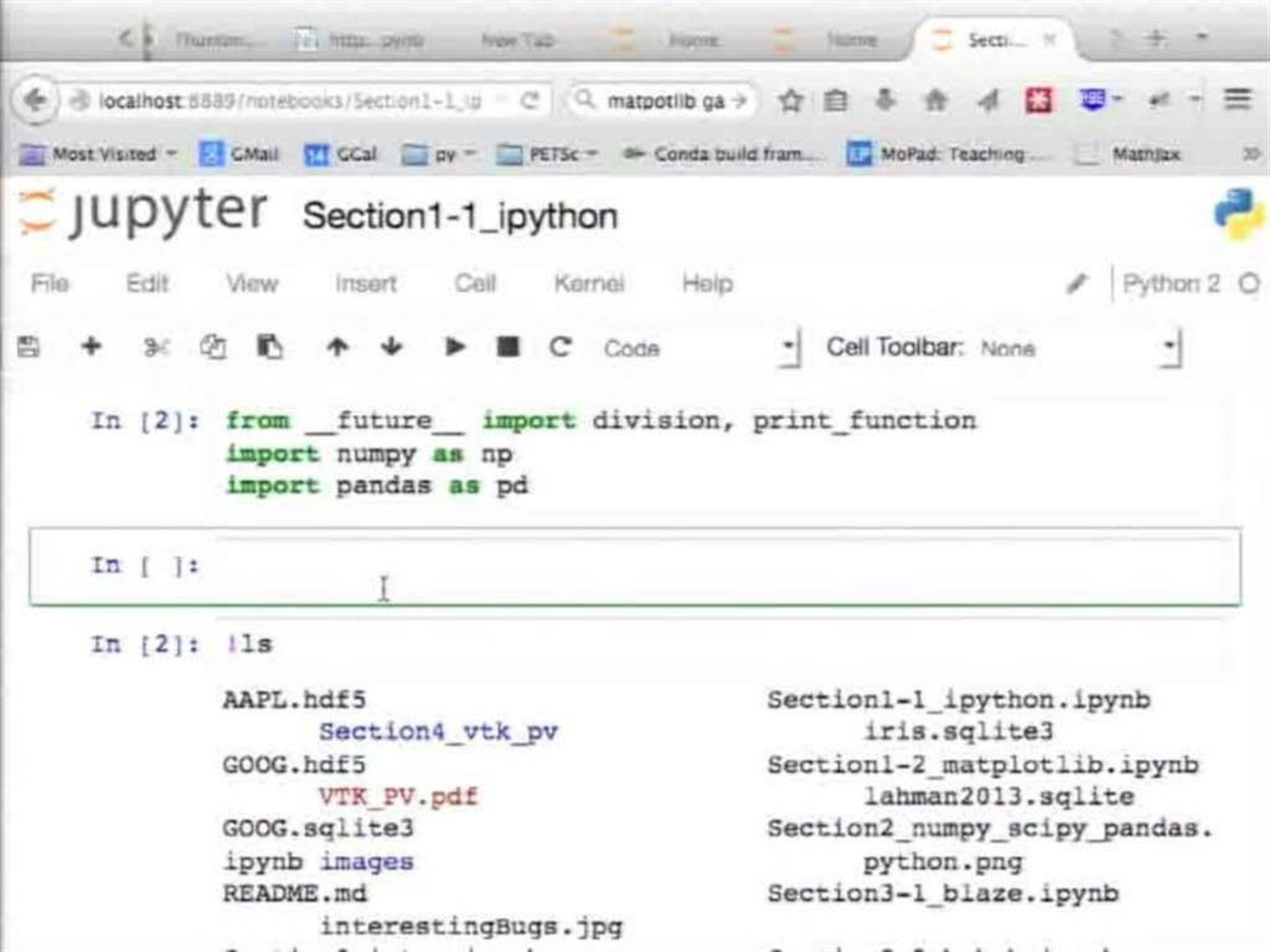
Chrome File Edit View History Bookmarks Window People Help Sun 9:19 AM
Built-in Retina Display
Display Color
Optimize for: VGA Display
Best for VGA Display
Scaled
Brightness:
Automatically adjust brightness
Gather Windows
?
Kind
Application
Folder
KB
MB
Show mirroring options in the menu bar when available
Gather Windows
Controller: ICH AC97
Network

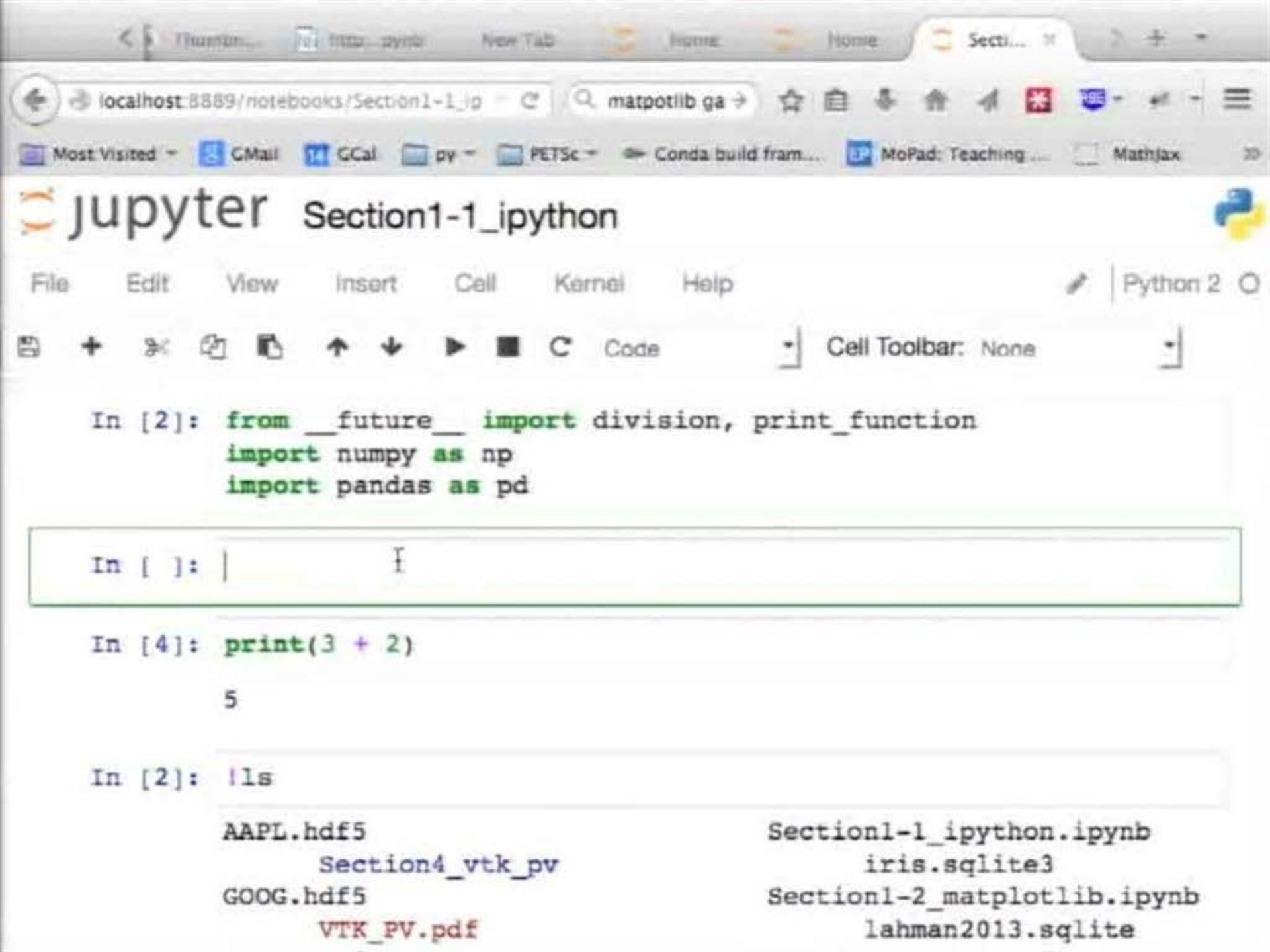

iTerm Shell Edit View Profiles Toolbelt Window Help
1. ~/sandbox/PythonSIAM2015
[Sun 15/03/15 09:20 MDT][s000][x86_64/darwin13.0.0/13.4.0][5.0.5]
<aron@Arons-MacBook-Pro ~/sandbox/PythonSIAM2015>
zsh 10007 (git)-[master]-%

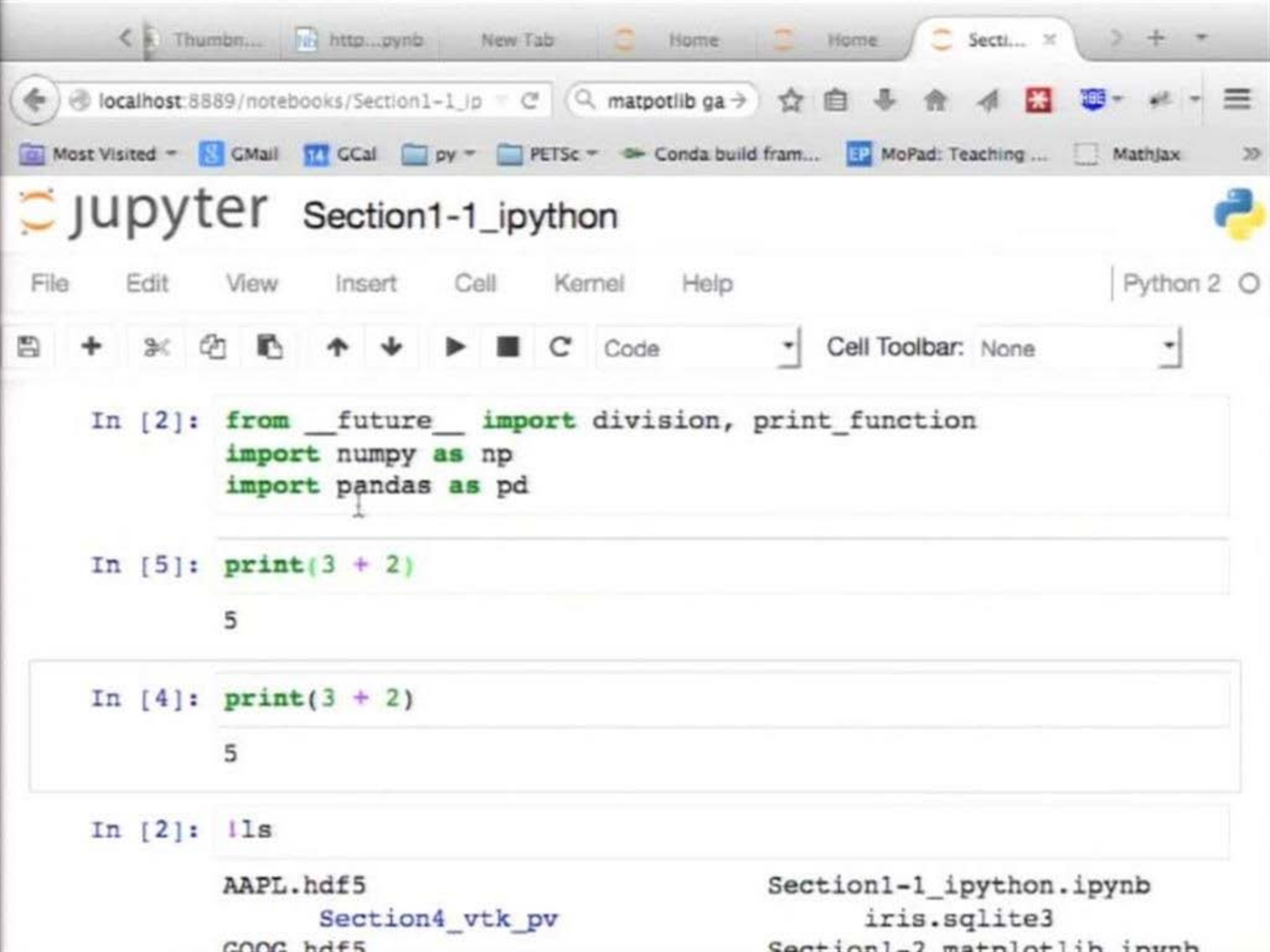

Firefox File Edit View History Bookmarks Tools Window Help
localhost:8889/notebooks/Section1-1_ipython.ipynb
Most Visited GMail GCal py PETSc Conda build fram... MoPad: Teaching ... MathJax
jupyter Section1-1_ipython Last Checkpoint: 11 hours ago (autos
File Edit View Insert Cell Kernel Help
Markdown
Cell Toolbar:
Introduction to IPython Notebook
Why do I use IPython notebook:
Interactive like a REPL
Durrable artifacts in cells
Shareable like a document...with built-in annotations!
A motivating example...

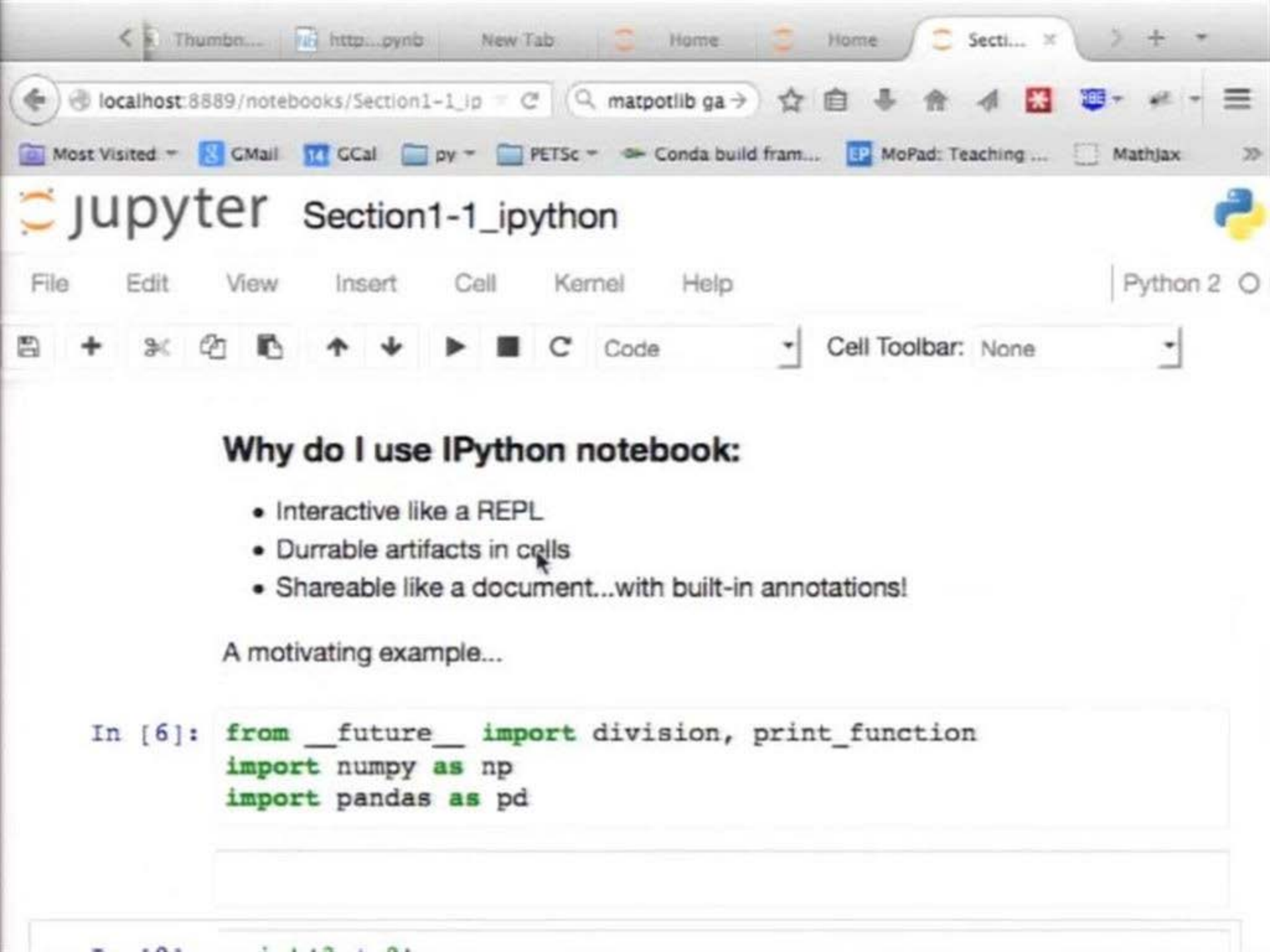

localhost:8889/notebooks/Section1-1_ip
matplotlib ga
Most Visited
GMail
GCal
py
PETSc
Conda build fram...
MoPad: Teaching
MathJax
jupyter
Section1-1_ipython
File
Edit
View
Insert
Cell
Kernel
Help
Python 2
Markdown
Cell Toolbar: None
Introduction to IPython Notebook
Why do I use IPython notebook:
Interactive like a REPL
Durrable artifacts in cells
Shareable like a document...with built-in annotations!
A motivating example...
In [1]:
from __future__ import division, print_function
import numpy as np
import pandas as pd
In [2]:
!ls

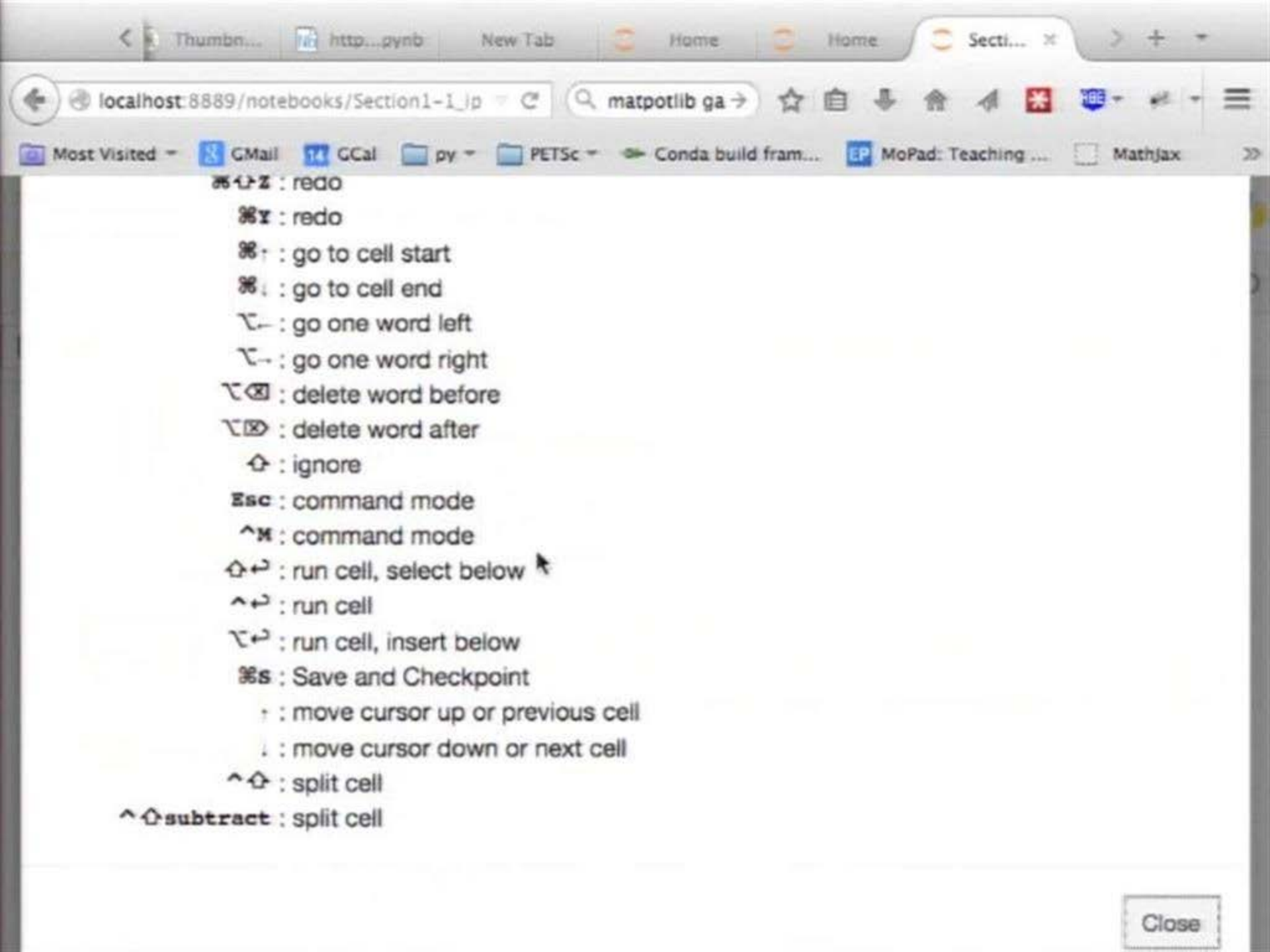

localhost:8889/notebooks/Section1-1_ip
matplotlib ga
Most Visited
GMail
GCal
py
PETSc
Conda build fram...
MoPad: Teaching
Mathjax
jupyter Section1-1_ipython
File Edit View Insert Cell Kernel Help
Python 2
Code
Cell Toolbar: None
In [1]:
from __future__ import division, print_function
import numpy as np
import pandas as pd
In [2]:
!ls
AAPL.hdf5
Section4_vtk_pv
GOOG.hdf5
VTK_PV.pdf
GOOG.sqlite3
ipynb images
README.md
interestingBugs.jpg
Section0_intro.ipynb
iris.csv
Section1-1_ipython.ipynb
iris.sqlite3
Section1-2_matplotlib.ipynb
lahman2013.sqlite
Section2_numpy_scipy_pandas.
python.png
Section3-1_blaze.ipynb
Section3-2_bokeh.ipynb

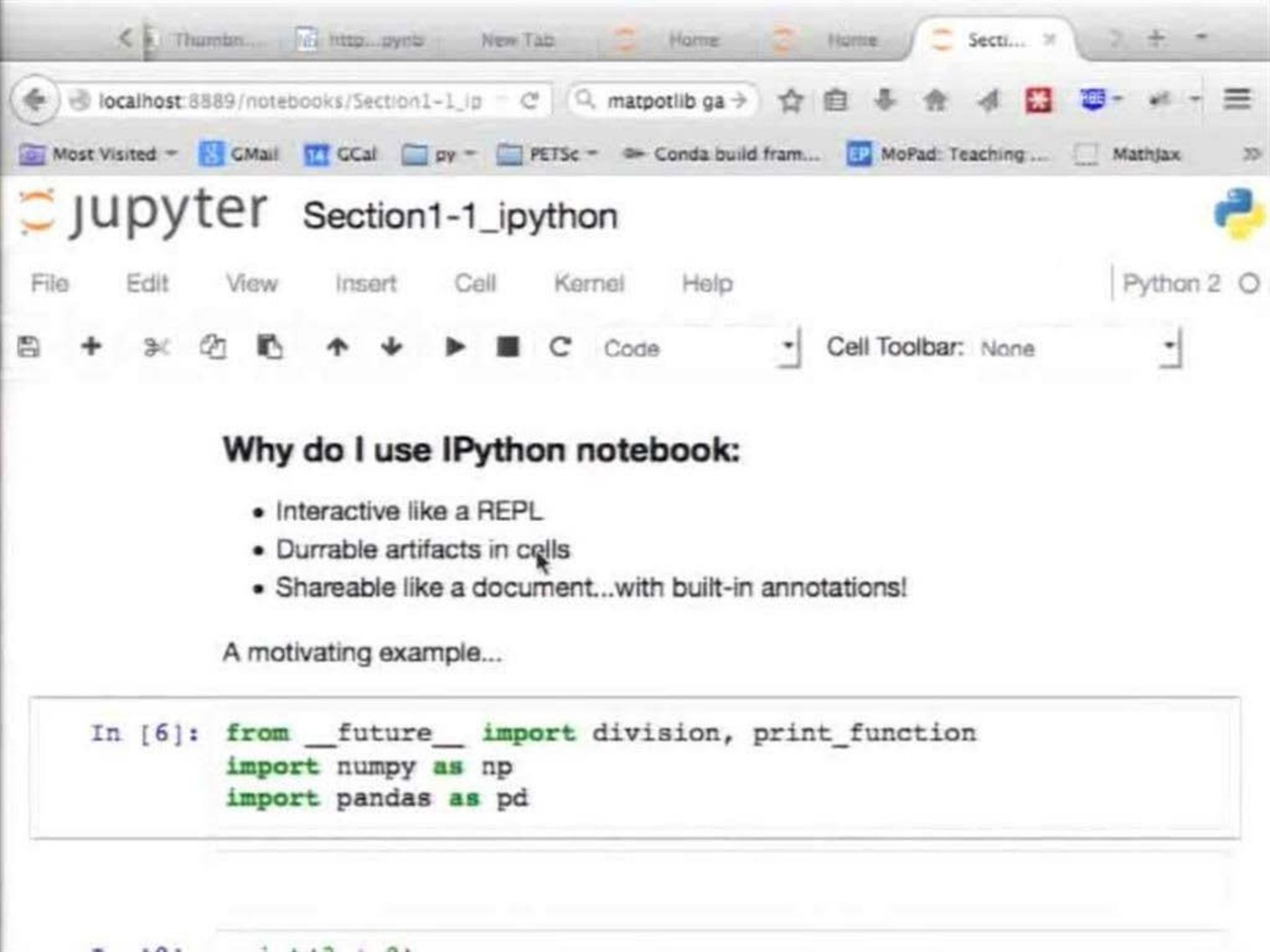

localhost:8889/notebooks/Section1-1_ip
matplotlib ga
Most Visited
GMail
GCal
py
PETSc
Conda build fram...
MoPad: Teaching ...
MathJax
jupyter
Section1-1_ipython
File
Edit
View
Insert
Cell
Kernel
Help
Python 2
Code
Cell Toolbar: None
In [2]: from __future__ import division, print_function
import numpy as np
import pandas as pd
In [ ]:
In [2]: !ls
AAPL.hdf5
Section4_vtk_pv
GOOG.hdf5
VTK_PV.pdf
GOOG.sqlite3
ipynb images
README.md
interestingBugs.jpg
Section1-1_ipython.ipynb
iris.sqlite3
Section1-2_matplotlib.ipynb
lahman2013.sqlite
Section2_numpy_scipy_pandas.
python.png
Section3-1_blaze.ipynb

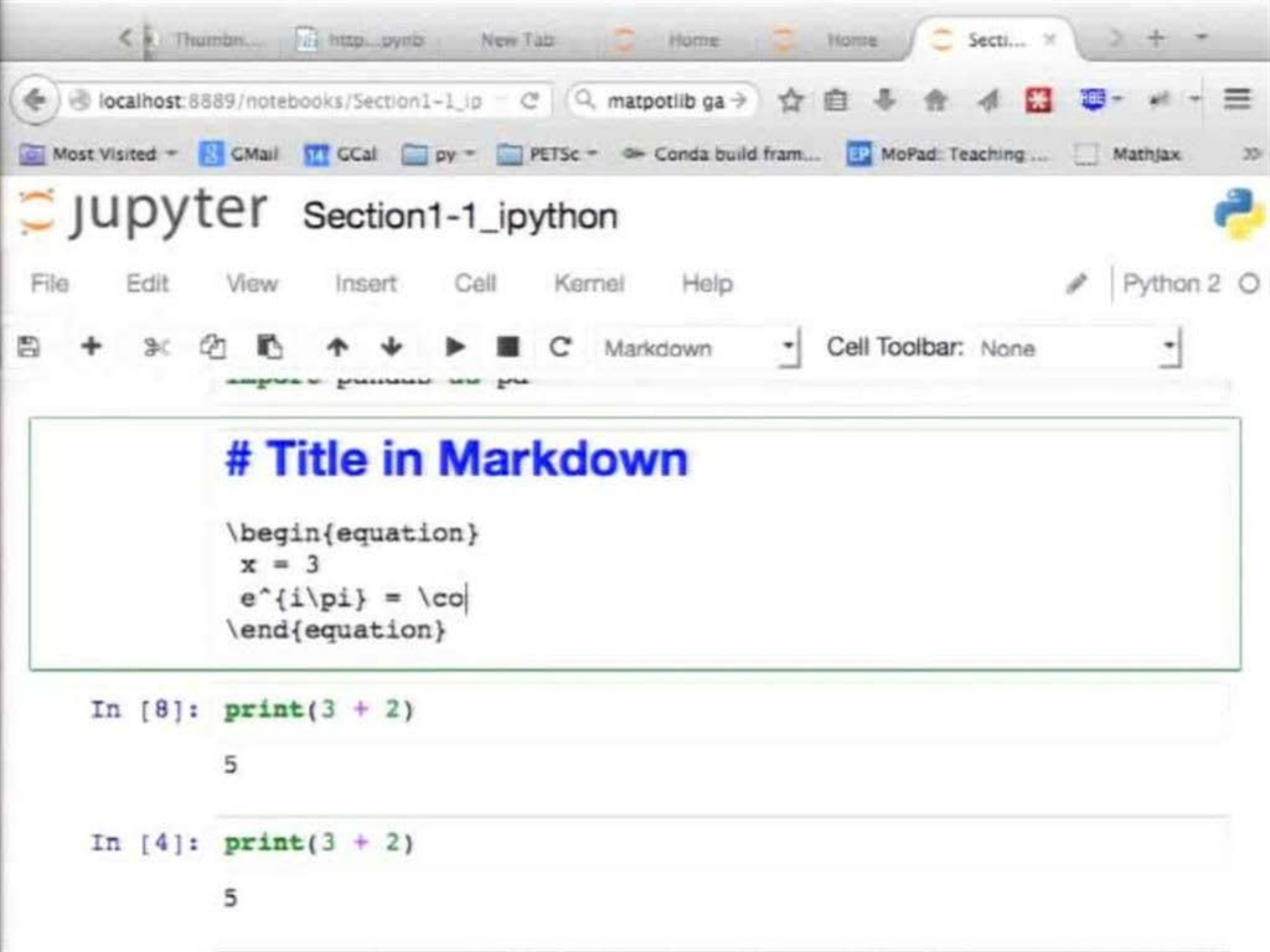

localhost:8889/notebooks/Section1-1_ip
matpotlib ga
Most Visited
GMail
GCal
py
PETSc
Conda build fram...
MoPad: Teaching ...
Mathjax
jupyter
Section1-1_ipython
File
Edit
View
Insert
Cell
Kernel
Help
Python 2
Code
Cell Toolbar: None
In [2]: from __future__ import division, print_function
import numpy as np
import pandas as pd
In [ ]:
In [4]: print(3 + 2)
5
In [2]: !ls
AAPL.hdf5
Section4_vtk_pv
GOOG.hdf5
VTK_PV.pdf
Section1-1_ipython.ipynb
iris.sqlite3
Section1-2_matplotlib.ipynb
lahman2013.sqlite


Thumbn...
http...pynb
New Tab
Home
Home
Secti...
localhost:8889/notebooks/Section1-1_ip
matplotlib ga
Most Visited
GMail
GCal
py
PETSc
Conda build fram...
MoPad: Teaching ...
MathJax
jupyter
Section1-1_ipython
File
Edit
View
Insert
Cell
Kernel
Help
Python 2
Code
Cell Toolbar:
None
In [2]:
from __future__ import division, print_function
import numpy as np
import pandas as pd
In [5]:
print(3 + 2)
5
In [4]:
print(3 + 2)
5
In [2]:
!ls
AAPL.hdf5
Section4_vtk_pv
Section1-1_ipython.ipynb
iris.sqlite3


Thumbn...
http...pynb
New Tab
Home
Home
Secti...
localhost:8889/notebooks/Section1-1_ip
matpotlib ga
Most Visited
GMail
GCal
py
PETSc
Conda build fram...
MoPad: Teaching ...
Mathjax
jupyter
Section1-1_ipython
File
Edit
View
Insert
Cell
Kernel
Help
Python 2
Code
Cell Toolbar:
None
Why do I use IPython notebook:
Interactive like a REPL
Durrable artifacts in cells
Shareable like a document...with built-in annotations!
A motivating example...
In [6]:
from __future__ import division, print_function
import numpy as np
import pandas as pd


Thumbn...
http...pynb
New Tab
Home
Home
Secti...
localhost:8889/notebooks/Section1-1_jp
matpotlib ga
Most Visited
GMail
GCal
py
PETSc
Conda build fram...
MoPad: Teaching ...
Mathjax
⌘⇧Z : redo
⌘Y : redo
⌘↑ : go to cell start
⌘↓ : go to cell end
⌥← : go one word left
⌥→ : go one word right
⌥⌫ : delete word before
⌥⌦ : delete word after
⇧ : ignore
Esc : command mode
^M : command mode
⇧↵ : run cell, select below
^↵ : run cell
⌥↵ : run cell, insert below
⌘S : Save and Checkpoint
↑ : move cursor up or previous cell
↓ : move cursor down or next cell
^⇧ : split cell
^⇧subtract : split cell
Close


localhost:8889/notebooks/Section1-1_ip
matpotlib ga
Most Visited
GMail
GCal
py
PETSc
Conda build fram...
MoPad: Teaching ...
Mathjax
jupyter
Section1-1_ipython
File
Edit
View
Insert
Cell
Kernel
Help
Python 2
Code
Cell Toolbar: None
Why do I use IPython notebook:
Interactive like a REPL
Durrable artifacts in cells
Shareable like a document...with built-in annotations!
A motivating example...
In [6]:
from __future__ import division, print_function
import numpy as np
import pandas as pd


Thumbn...
http...pynb
New Tab
Home
Home
Secti...
localhost:8889/notebooks/Section1-1_ip
matpotlib ga
Most Visited
GMail
GCal
py
PETSc
Conda build fram...
MoPad: Teaching ...
Mathjax
jupyter
Section1-1_ipython
File
Edit
View
Insert
Cell
Kernel
Help
Python 2
Markdown
Cell Toolbar: None
Title in Markdown
\begin{equation}
x = 3
e^{i\pi} = \co
\end{equation}
In [8]:
print(3 + 2)
5
In [4]:
print(3 + 2)
5

Thumbn... | http...pynb | New Tab | Home | Home | Secti...

localhost:8889/notebooks/Section1-1_ip | matpotlib ga

Most Visited | GMail | GCal | py | PETSc | Conda build fram... | MoPad: Teaching ... | MathJax

jupyter Section1-1_ipython

File Edit View Insert Cell Kernel Help | Python 2

Code | Cell Toolbar: None

Title in Markdown

$$x = 3e^{i\pi} = \cos x + i \sin x$$

In [8]:
```
print(3 + 2)
```
```
5
```

In [4]:
```
print(3 + 2)
```
```
5
```

In [2]:
```
!ls
```

Thumbn... | http...pynb | New Tab | Home | Home | Secti...

localhost:8889/notebooks/Section1-1_ip | matpotlib ga

Most Visited | GMail | GCal | py | PETSc | Conda build fram... | MoPad: Teaching ... | Mathjax

jupyter Section1-1_ipython

File Edit View Insert Cell Kernel Help | Python 2

Markdown | Cell Toolbar: None

```
\begin{equation}
 x = 3
 e^{i\pi} = \cos x + i \sin x
\end{equation}
```

In [8]:
```
print(3 + 2)
```

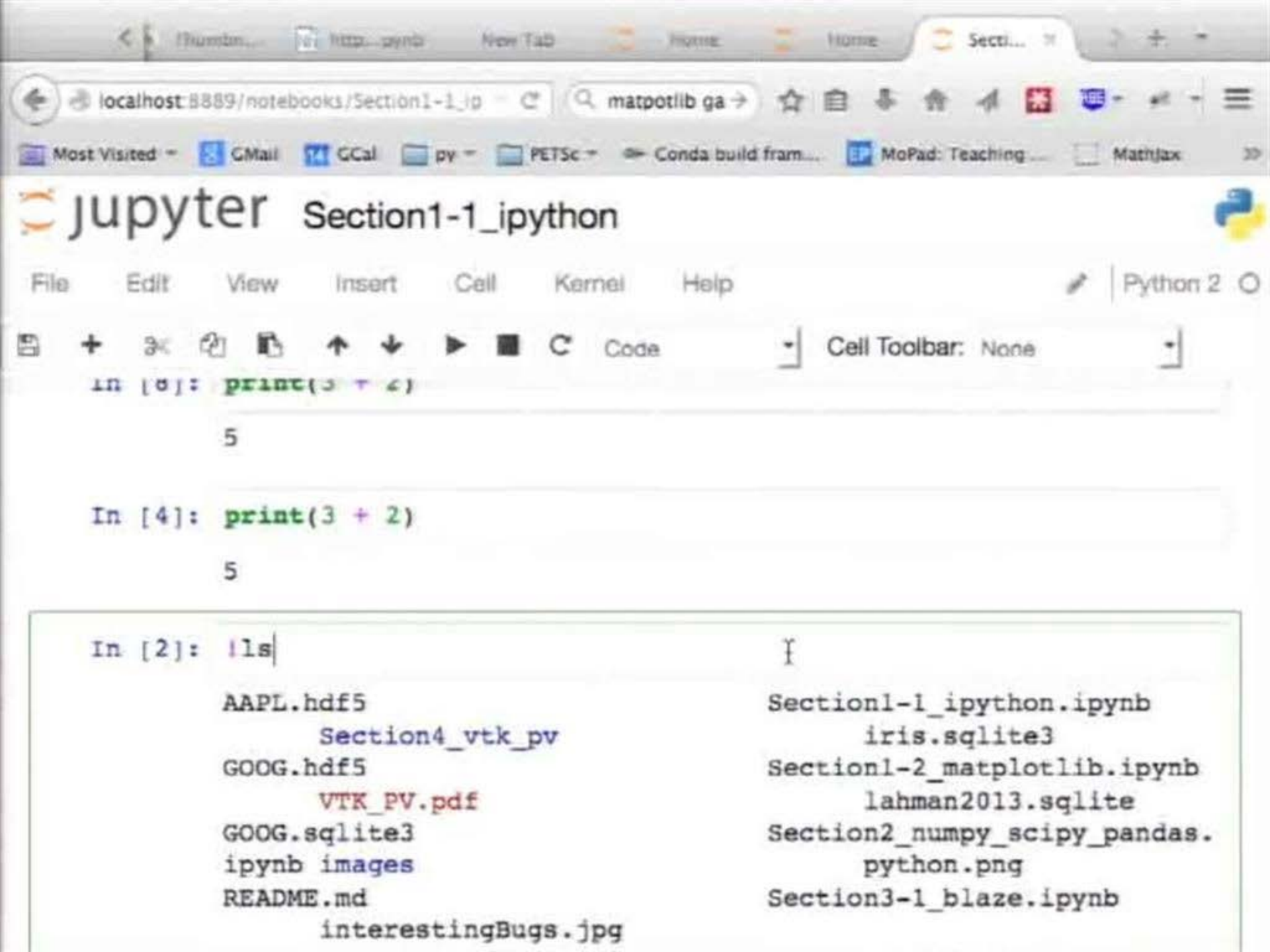
```
5
```

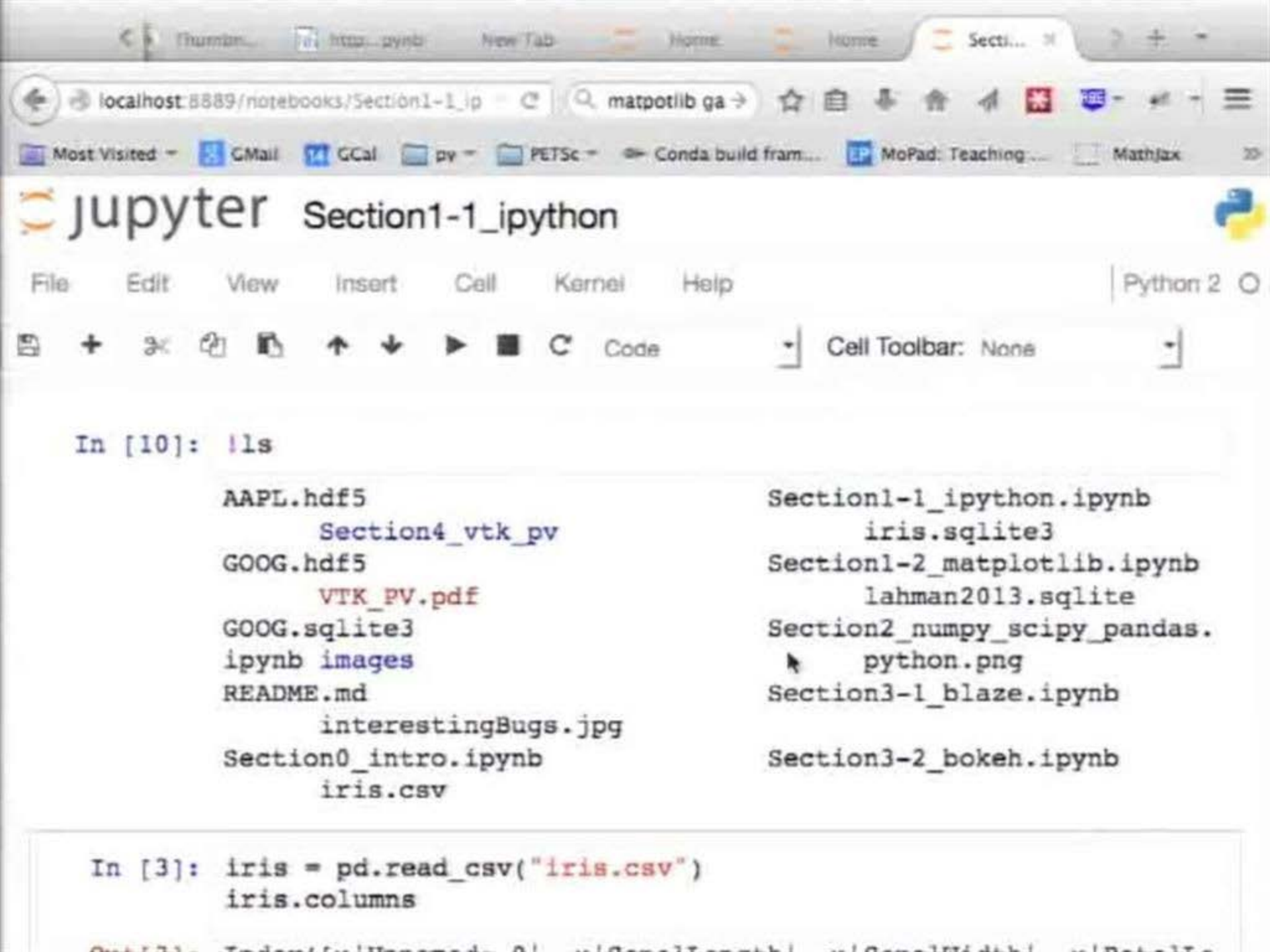
In [4]:
```
print(3 + 2)
```

```
5
```

In [2]:
```
!ls
```

```
AAPL.hdf5                    Section1-1_ipython.ipynb
```


localhost:8889/notebooks/Section1-1_ip
matpotlib ga
Most Visited
GMail
GCal
py
PETSc
Conda build fram...
MoPad: Teaching ...
MathJax
jupyter
Section1-1_ipython
File
Edit
View
Insert
Cell
Kernel
Help
Python 2
Code
Cell Toolbar: None
In [3]: print(3 + 2)
5
In [4]: print(3 + 2)
5
In [2]: !ls
AAPL.hdf5
Section4_vtk_pv
GOOG.hdf5
VTK_PV.pdf
GOOG.sqlite3
ipynb images
README.md
interestingBugs.jpg
Section1-1_ipython.ipynb
iris.sqlite3
Section1-2_matplotlib.ipynb
lahman2013.sqlite
Section2_numpy_scipy_pandas.
python.png
Section3-1_blaze.ipynb


localhost:8889/notebooks/Section1-1_ip
matpotlib ga
Most Visited
GMail
GCal
py
PETSc
Conda build fram...
MoPad: Teaching ...
MathJax
jupyter
Section1-1_ipython
File
Edit
View
Insert
Cell
Kernel
Help
Python 2
Code
Cell Toolbar: None
In [10]: !ls
AAPL.hdf5
Section4_vtk_pv
GOOG.hdf5
VTK_PV.pdf
GOOG.sqlite3
ipynb images
README.md
interestingBugs.jpg
Section0_intro.ipynb
iris.csv
Section1-1_ipython.ipynb
iris.sqlite3
Section1-2_matplotlib.ipynb
lahman2013.sqlite
Section2_numpy_scipy_pandas.
python.png
Section3-1_blaze.ipynb
Section3-2_bokeh.ipynb
In [3]: iris = pd.read_csv("iris.csv")
iris.columns

jupyter Section1-1_ipython

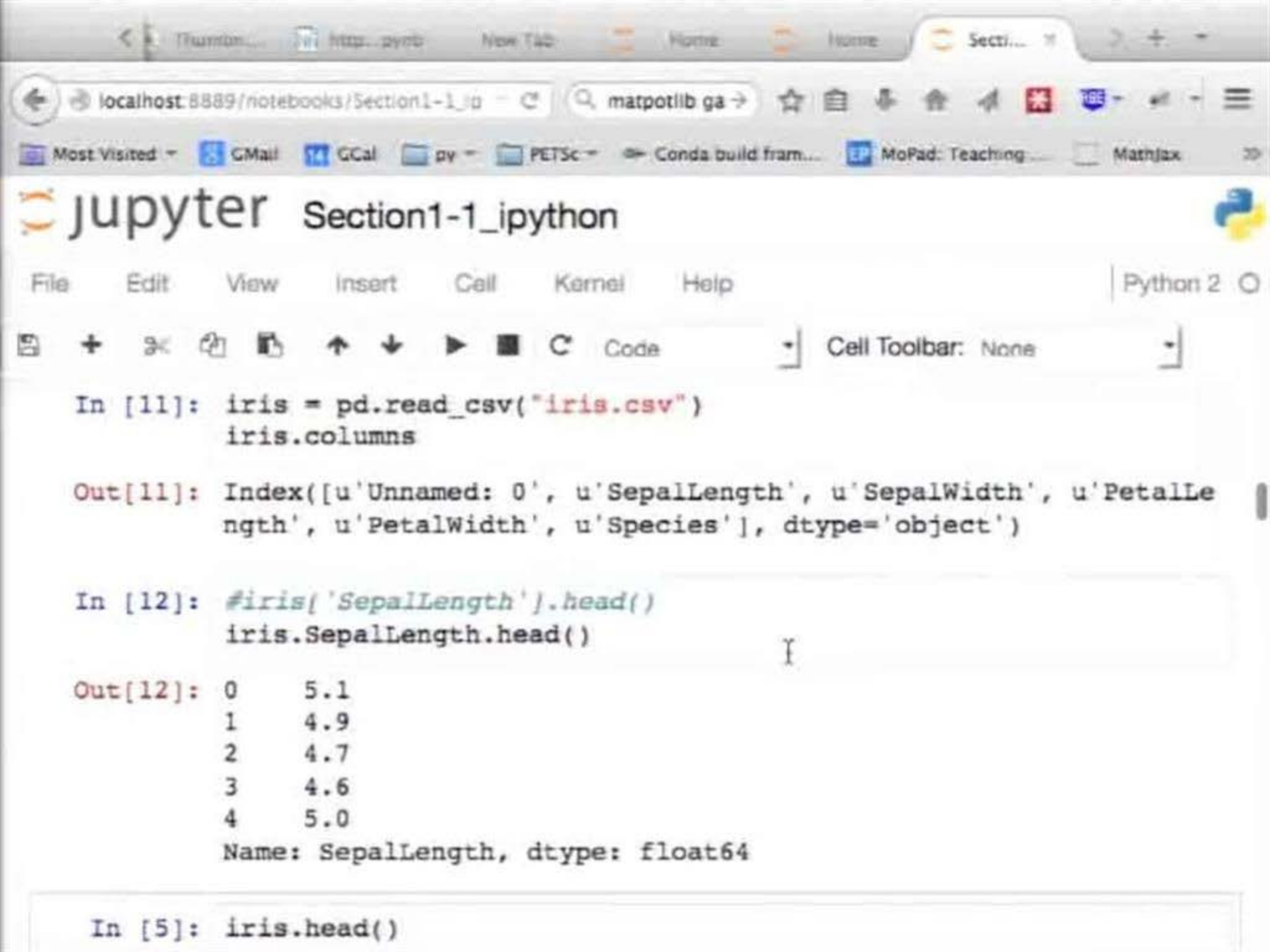

```
README.md                          Section3-1_blaze.ipynb
      interestingBugs.jpg
Section0_intro.ipynb               Section3-2_bokeh.ipynb
      iris.csv
```

In [11]:
```
iris = pd.read_csv("iris.csv")
iris.columns
```

Out[11]:
```
Index([u'Unnamed: 0', u'SepalLength', u'SepalWidth', u'PetalLe
ngth', u'PetalWidth', u'Species'], dtype='object')
```

In [4]:
```
#iris['SepalLength'].head()
iris.SepalLength.head()
```

Out[4]:
```
0    5.1
1    4.9
2    4.7
3    4.6
```


localhost:8889/notebooks/Section1-1_ip
matplotlib ga
Most Visited
GMail
GCal
py
PETSc
Conda build fram...
MoPad: Teaching ...
MathJax
jupyter
Section1-1_ipython
File
Edit
View
Insert
Cell
Kernel
Help
Python 2
Code
Cell Toolbar: None
In [11]: iris = pd.read_csv("iris.csv")
iris.columns
Out[11]: Index([u'Unnamed: 0', u'SepalLength', u'SepalWidth', u'PetalLe
ngth', u'PetalWidth', u'Species'], dtype='object')
In [12]: #iris['SepalLength'].head()
iris.SepalLength.head()
Out[12]: 0 5.1
1 4.9
2 4.7
3 4.6
4 5.0
Name: SepalLength, dtype: float64
In [5]: iris.head()

localhost:8889/notebooks/Section1-1_ip

matpotlib ga

Most Visited | GMail | GCal | py | PETSc | Conda build fram... | MoPad: Teaching ... | MathJax

jupyter Section1-1_ipython

File Edit View Insert Cell Kernel Help Python 2

Code Cell Toolbar: None

In [13]: `iris.head()`

Out[13]:

	Unnamed: 0	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
0	1	5.1	3.5	1.4	0.2	setosa
1	2	4.9	3.0	1.4	0.2	setosa
2	3	4.7	3.2	1.3	0.2	setosa
3	4	4.6	3.1	1.5	0.2	setosa
4	5	5.0	3.6	1.4	0.2	setosa

In [6]:
```
import matplotlib.pyplot as plt
%matplotlib inline
```

In [7]:
```
plt.scatter(iris.PetalLength, iris.PetalWidth)
```

Thumbn... | http...pynb | New Tab | Home | Home | Secti...

localhost:8889/notebooks/Section1-1_ip | matpotlib ga

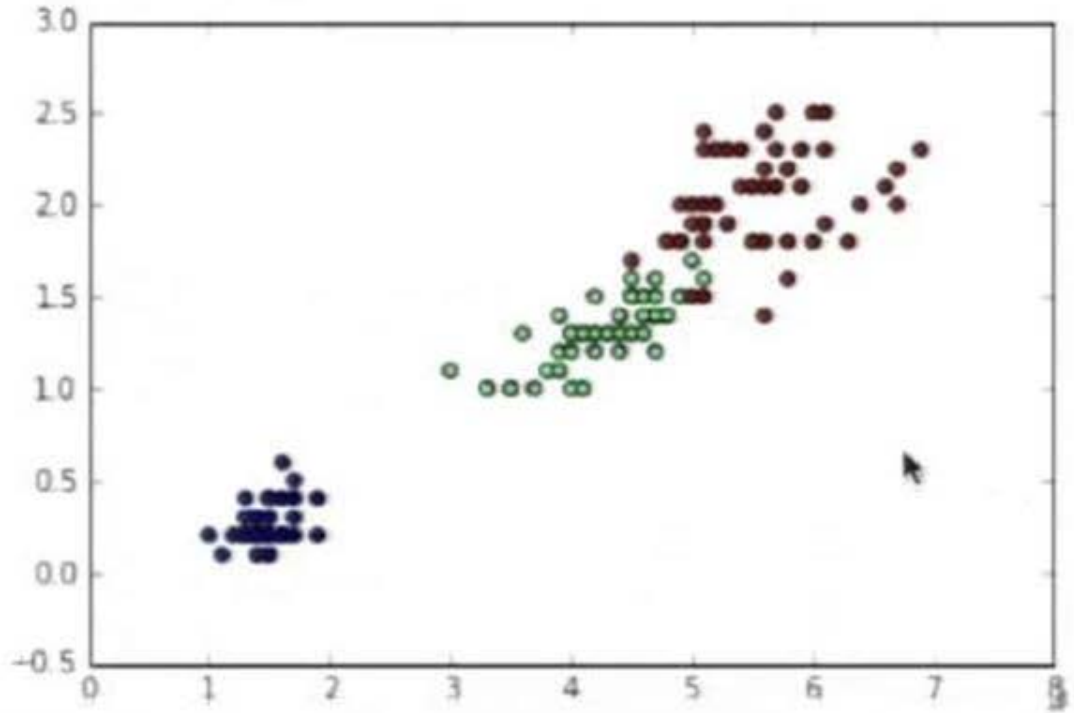
Most Visited | GMail | GCal | py | PETSc | Conda build fram... | MoPad: Teaching ... | Mathjax

jupyter Section1-1_ipython

File Edit View Insert Cell Kernel Help | Python 2

Markdown | Cell Toolbar: None

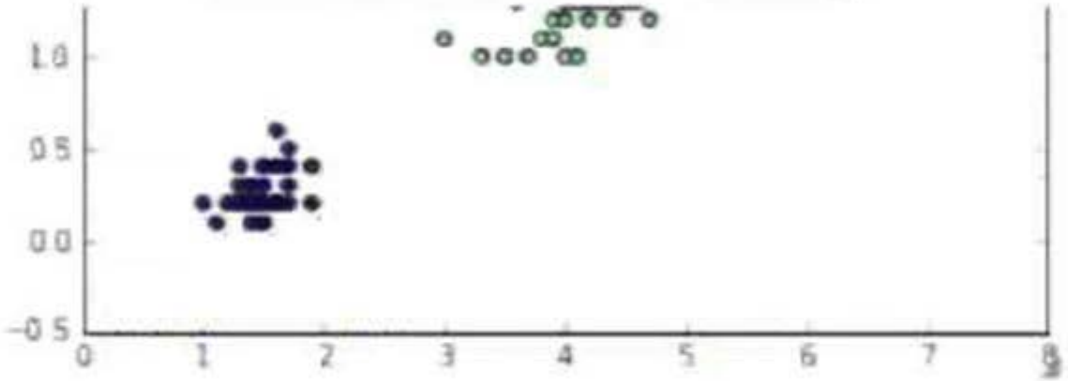
```
Out[15]: <matplotlib.collections.PathCollection at 0x10b50bf50>
```



jupyter Section1-1_ipython

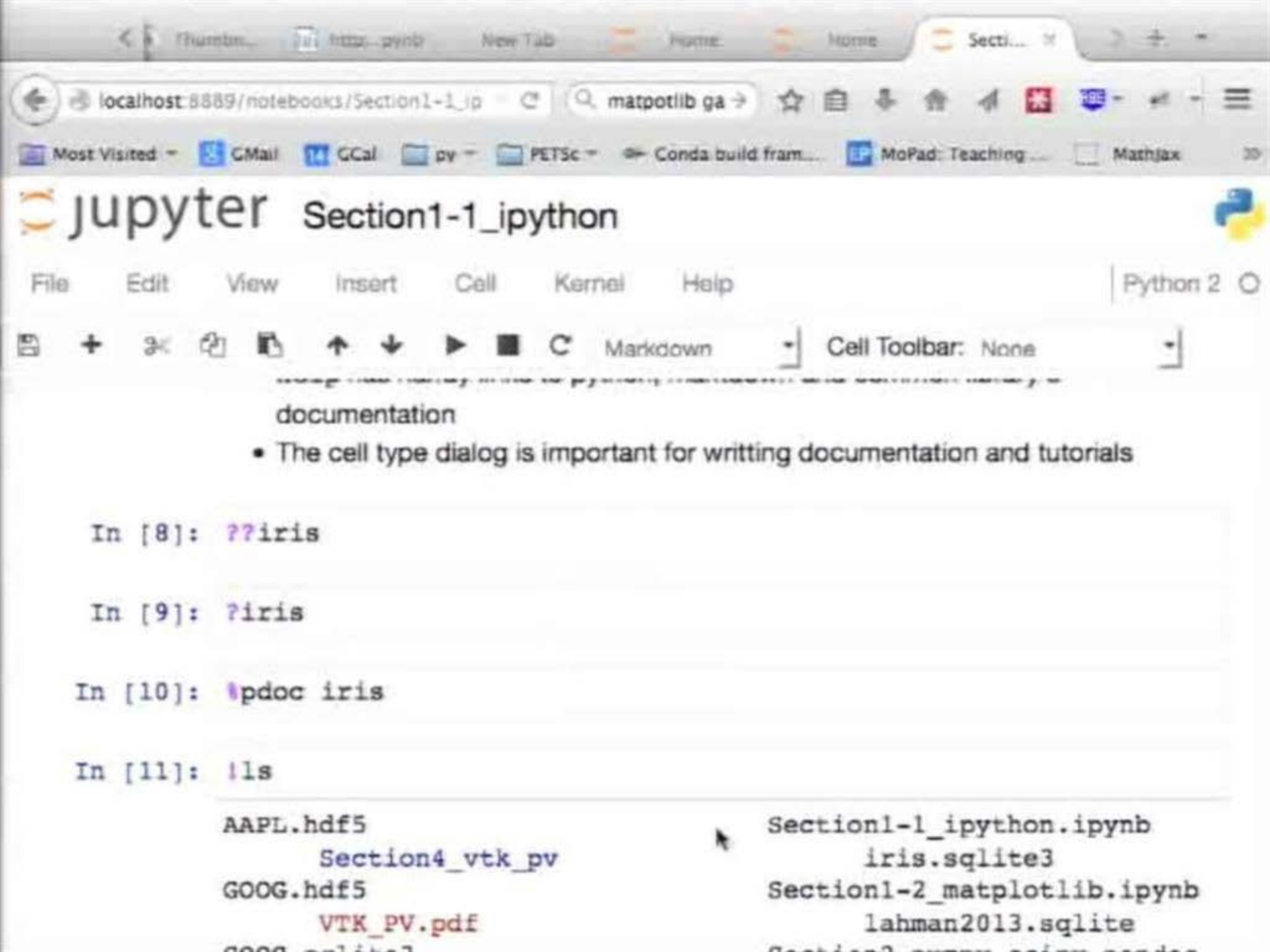
File Edit View Insert Cell Kernel Help Python 2

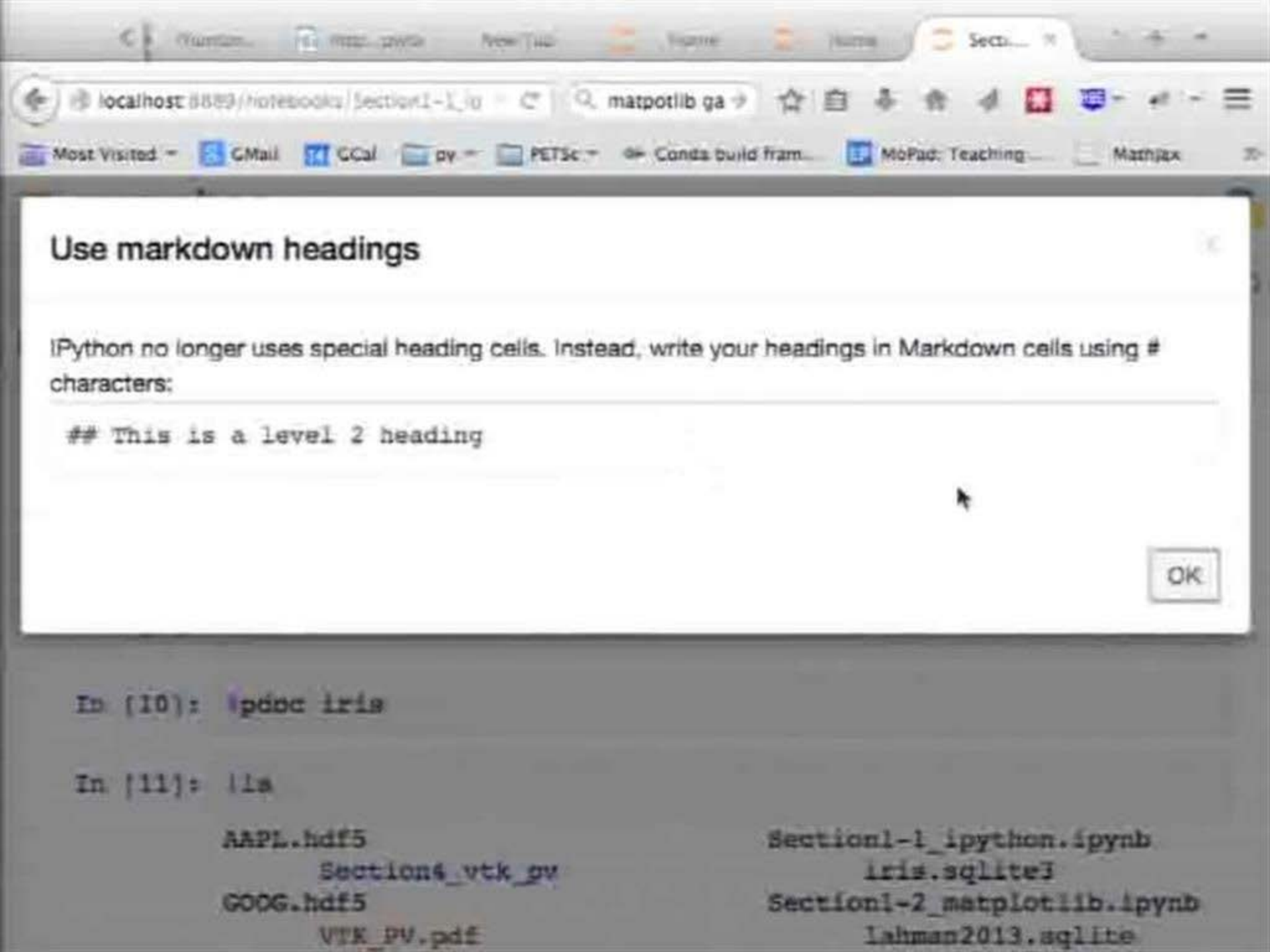
Markdown Cell Toolbar: None

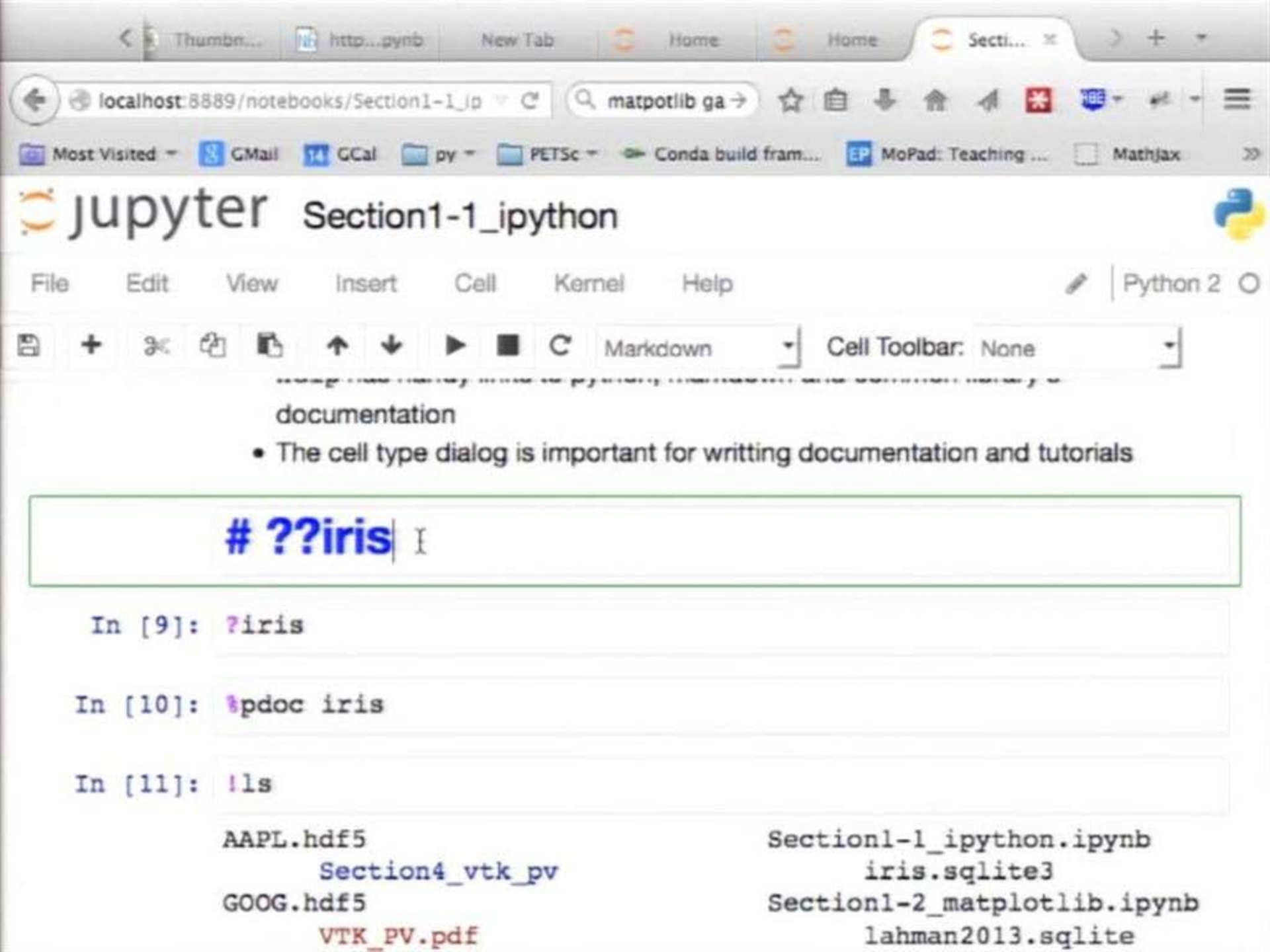


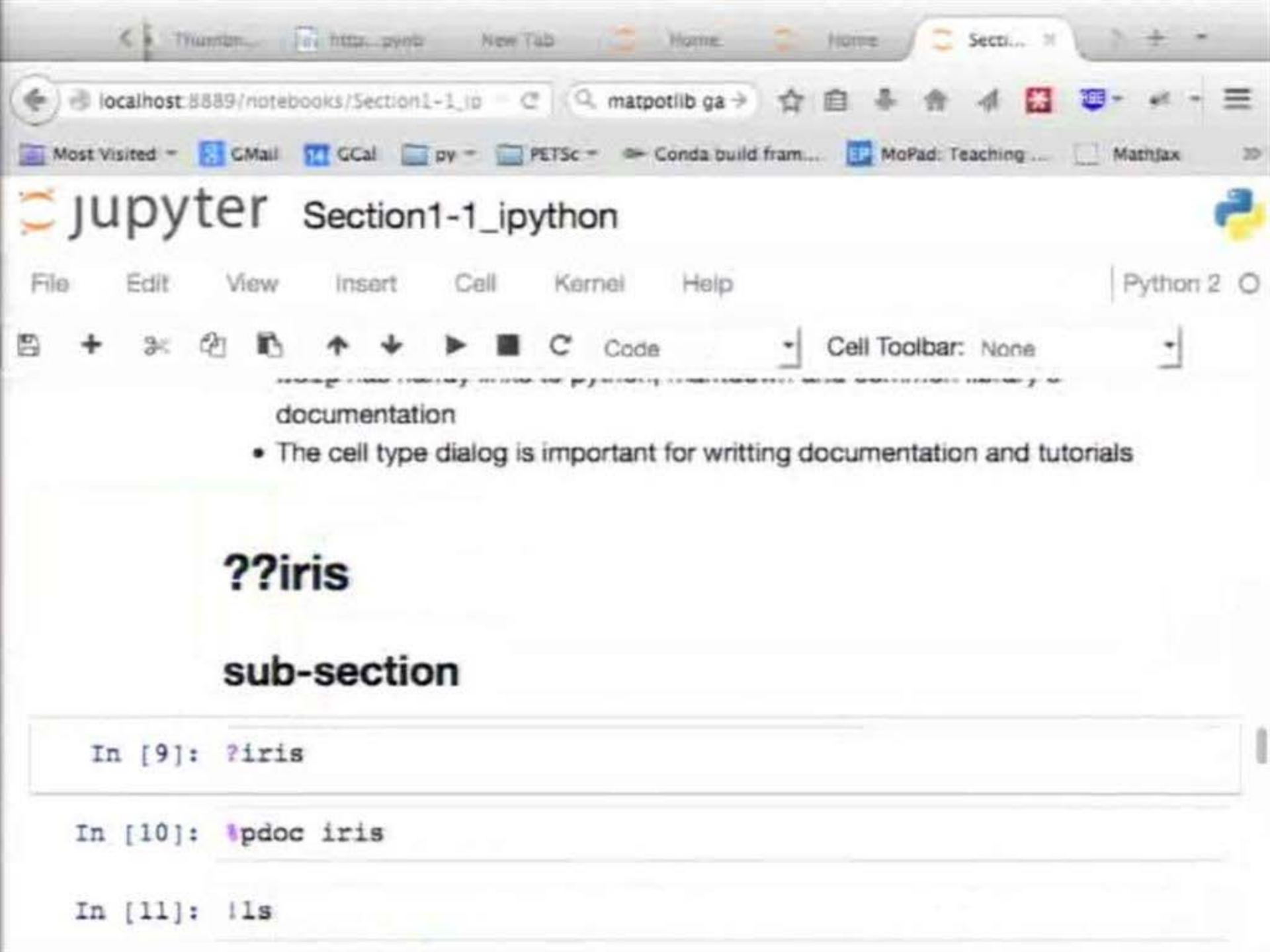
Notebook startup

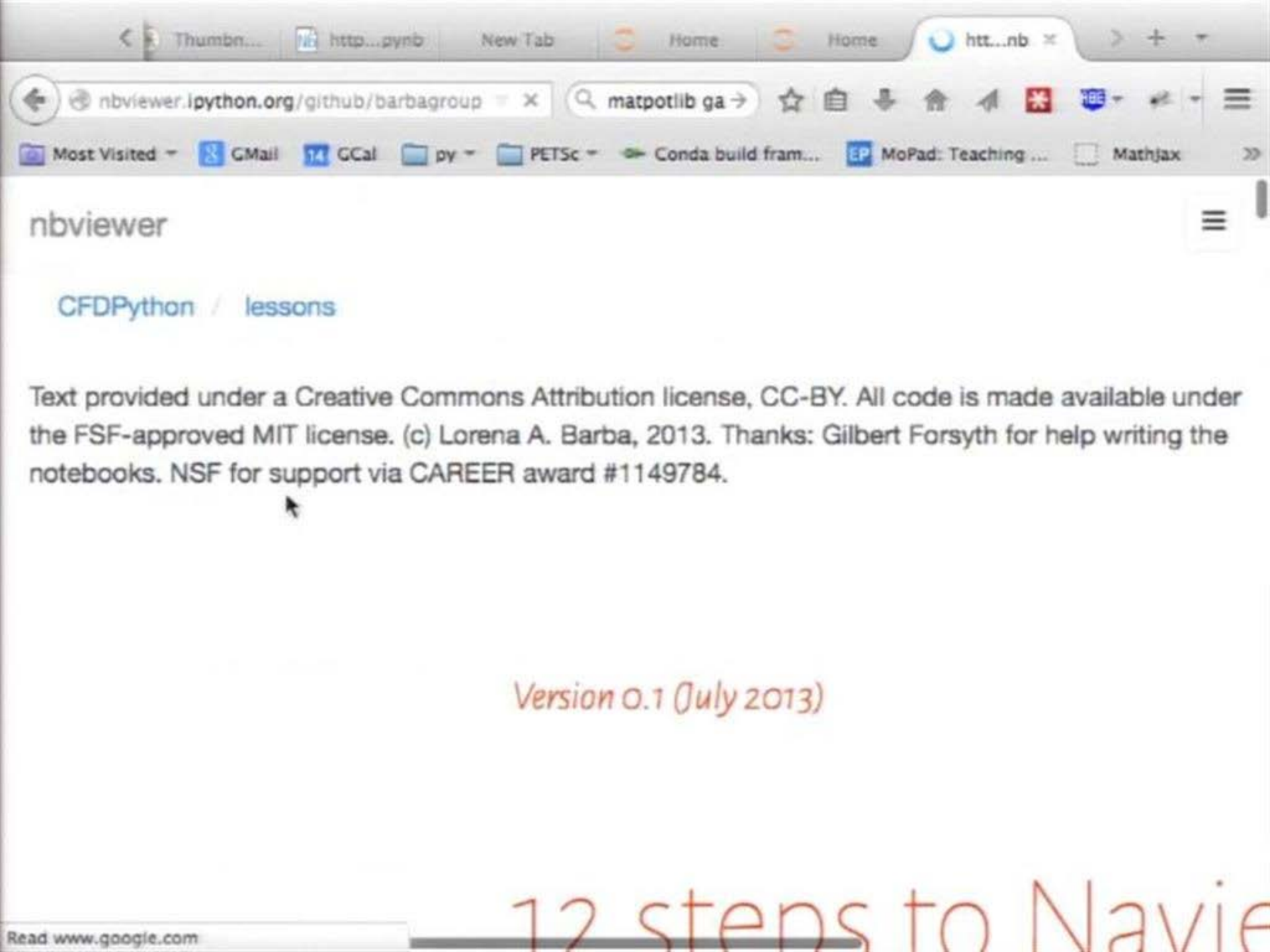
IPython notebook uses a client/server application. It can be accessed remotely if it lives on a accessible server. However, it is commonly used through localhost.


localhost:8889/notebooks/Section1-1_ip
matpotlib ga
Most Visited
GMail
GCal
py
PETSc
Conda build fram...
MoPad: Teaching ...
MathJax
jupyter Section1-1_ipython
File Edit View Insert Cell Kernel Help
Python 2
Markdown
Cell Toolbar: None
documentation
• The cell type dialog is important for writting documentation and tutorials
In [8]: ??iris
In [9]: ?iris
In [10]: %pdoc iris
In [11]: !ls
AAPL.hdf5
Section4_vtk_pv
GOOG.hdf5
VTK_PV.pdf
Section1-1_ipython.ipynb
iris.sqlite3
Section1-2_matplotlib.ipynb
lahman2013.sqlite


Use markdown headings
IPython no longer uses special heading cells. Instead, write your headings in Markdown cells using # characters:
This is a level 2 heading
OK
In [10]: %pdoc iris
In [11]: ls
AAPL.hdf5
Section4_vtk_gv
GOOG.hdf5
VTK_PV.pdf
Section1-1_ipython.ipynb
iris.sqlite3
Section1-2_matplotlib.ipynb
lahman2013.sqlite


Thumbn...
http...pynb
New Tab
Home
Home
Secti...
localhost:8889/notebooks/Section1-1_ip
matplotlib ga
Most Visited
GMail
GCal
py
PETSc
Conda build fram...
MoPad: Teaching ...
Mathjax
jupyter
Section1-1_ipython
File
Edit
View
Insert
Cell
Kernel
Help
Python 2
Markdown
Cell Toolbar: None
documentation
• The cell type dialog is important for writting documentation and tutorials
??iris
In [9]: ?iris
In [10]: %pdoc iris
In [11]: !ls
AAPL.hdf5
Section4_vtk_pv
GOOG.hdf5
VTK_PV.pdf
Section1-1_ipython.ipynb
iris.sqlite3
Section1-2_matplotlib.ipynb
lahman2013.sqlite


localhost:8889/notebooks/Section1-1_ip
matplotlib ga
Most Visited
GMail
GCal
py
PETSc
Conda build fram...
MoPad: Teaching ...
MathJax
jupyter
Section1-1_ipython
File
Edit
View
Insert
Cell
Kernel
Help
Python 2
Code
Cell Toolbar: None
documentation
• The cell type dialog is important for writting documentation and tutorials
??iris
sub-section
In [9]: ?iris
In [10]: %pdoc iris
In [11]: !ls


Thumbn...
http...pynb
New Tab
Home
Home
htt...nb
nbviewer.ipython.org/github/barbagroup
matpotlib ga
Most Visited
GMail
GCal
py
PETSc
Conda build fram...
MoPad: Teaching ...
Mathjax
nbviewer
CFDPython / lessons
Text provided under a Creative Commons Attribution license, CC-BY. All code is made available under the FSF-approved MIT license. (c) Lorena A. Barba, 2013. Thanks: Gilbert Forsyth for help writing the notebooks. NSF for support via CAREER award #1149784.
Version 0.1 (July 2013)
12 steps to Navie
Read www.google.com

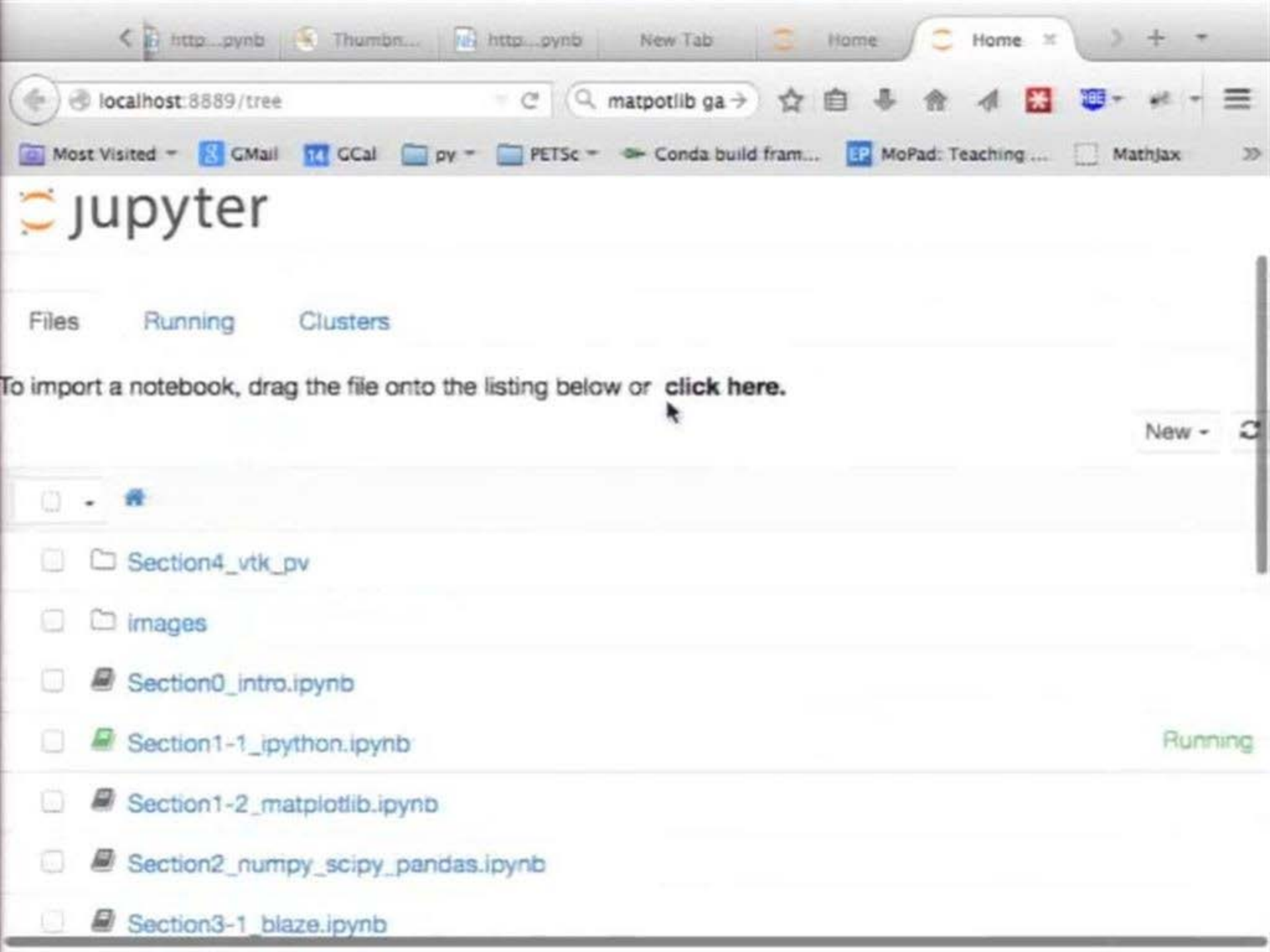
[2]:
```
#this makes matplotlib plots appear in the notebook (instead of a separa
%matplotlib inline
```

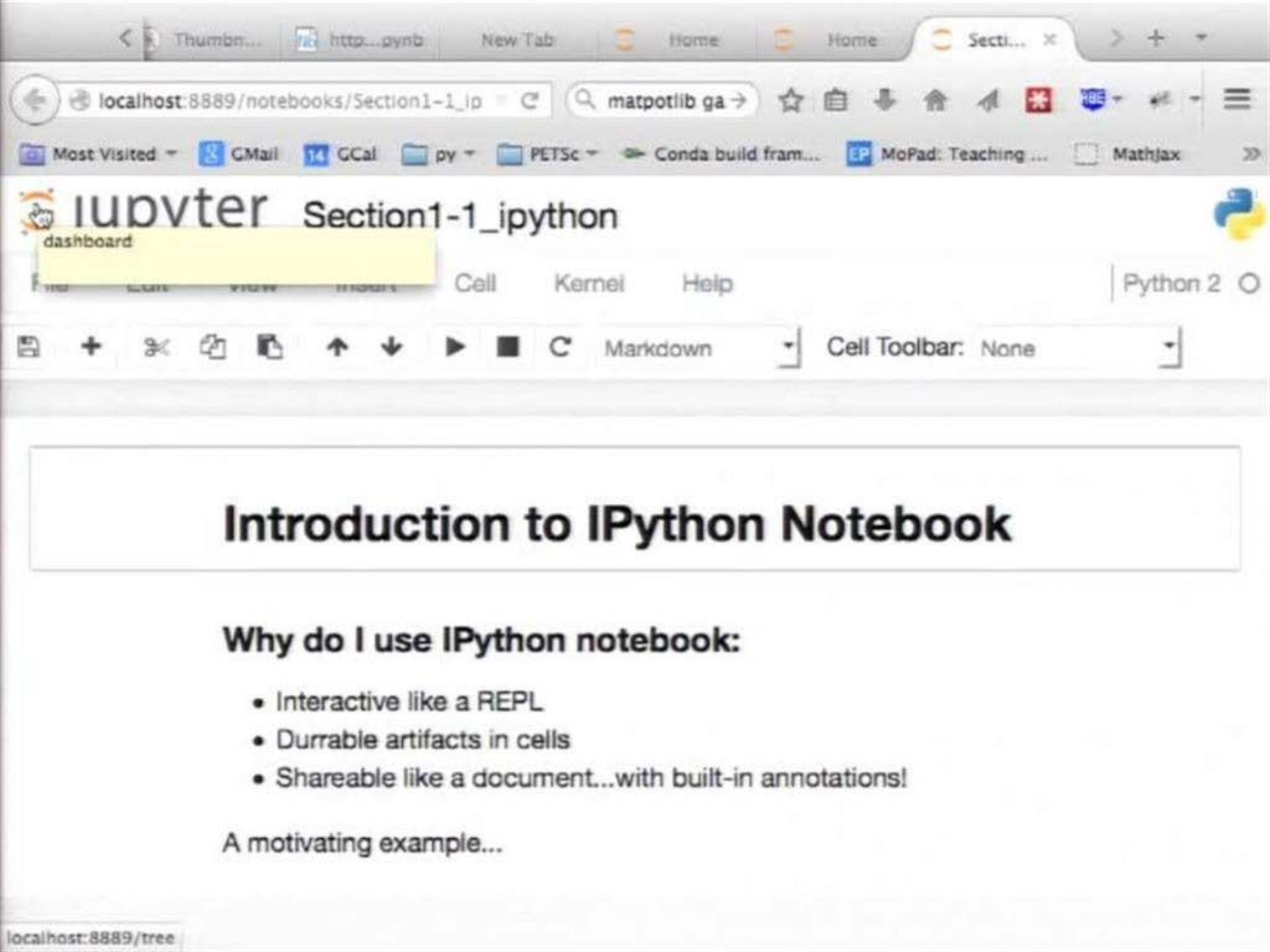
Now let's define a few variables; we want to define an evenly spaced grid of points with spatial domain that is 2 units of length wide, i.e., $x_i \in (0, 2)$. We'll define a variable **nx** which will be the number of grid points we want and **dx** will be the distance between a pair of adjacent grid points.

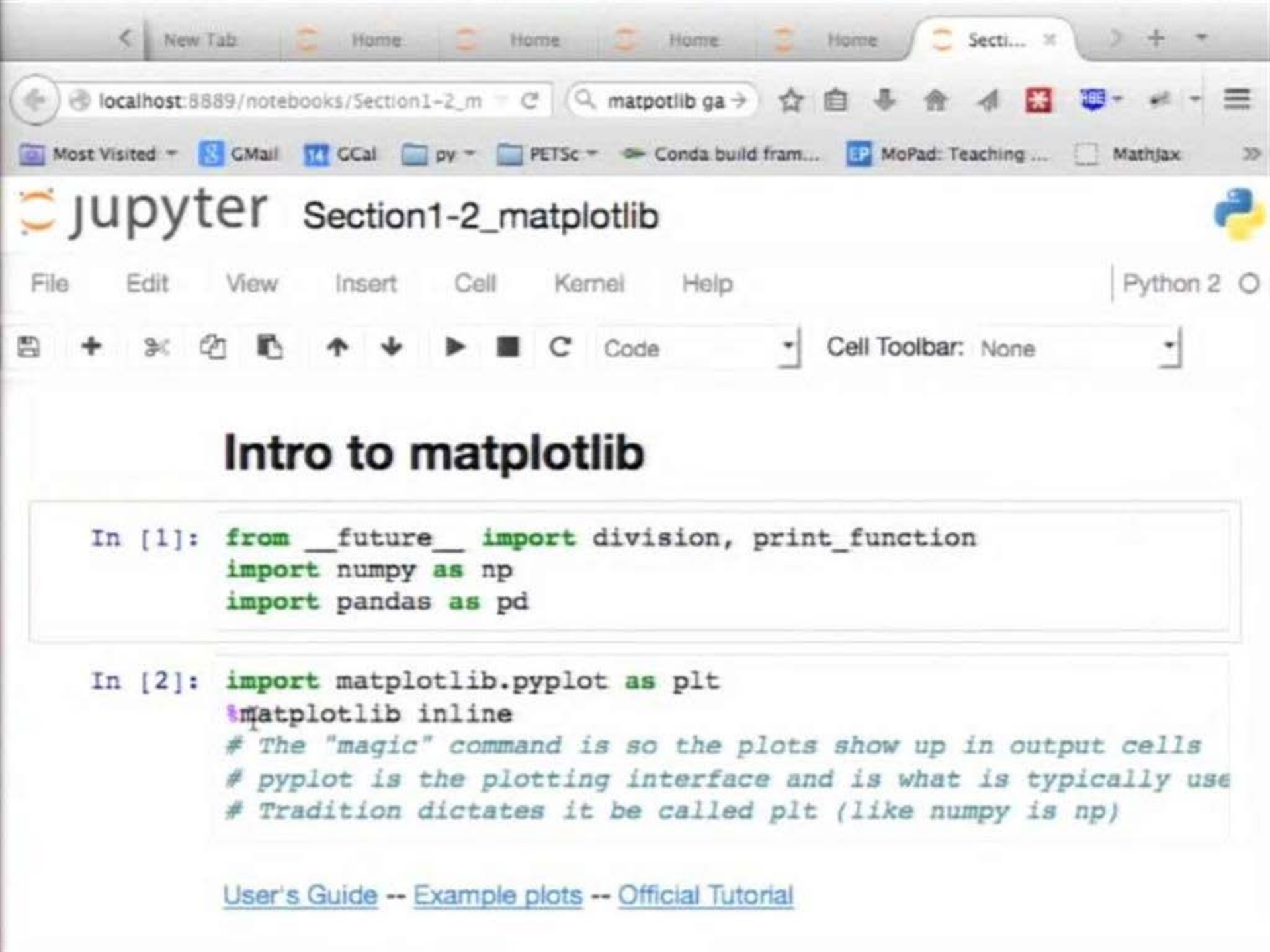
[3]:
```
nx = 41  # try changing this number from 41 to 81 and Run All ... what h
dx = 2./(nx-1)
nt = 25    #nt is the number of timesteps we want to calculate
dt = .025  #dt is the amount of time each timestep covers (delta t)
c = 1.      #assume wavespeed of c = 1
```

We also need to set up our initial conditions. The initial velocity u_0 is given as $u = 2$ in interval $0.5 \leq x \leq 1$ and $u = 1$ everywhere else in $(0, 2)$ (i.e., a hat function).

Here, we use the function **ones ( )** defining a **numpy** array which is **nx** elements long


http...pynb
Thumbn...
http...pynb
New Tab
Home
Home
localhost:8889/tree
matpotlib ga
Most Visited
GMail
GCal
py
PETSc
Conda build fram...
MoPad: Teaching ...
Mathjax
jupyter
Files
Running
Clusters
To import a notebook, drag the file onto the listing below or click here.
New
Section4_vtk_pv
images
Section0_intro.ipynb
Section1-1_ipython.ipynb
Running
Section1-2_matplotlib.ipynb
Section2_numpy_scipy_pandas.ipynb
Section3-1_blaze.ipynb


Thumbn...
http...ipynb
New Tab
Home
Home
Secti...
localhost:8889/notebooks/Section1-1_ip
matplotlib ga
Most Visited
GMail
GCal
py
PETSc
Conda build fram...
MoPad: Teaching ...
Mathjax
jupyter
Section1-1_ipython
dashboard
Cell
Kernel
Help
Python 2
Markdown
Cell Toolbar: None
Introduction to IPython Notebook
Why do I use IPython notebook:
Interactive like a REPL
Durrable artifacts in cells
Shareable like a document...with built-in annotations!
A motivating example...
localhost:8889/tree


New Tab
Home
Home
Home
Home
Secti...
localhost:8889/notebooks/Section1-2_m
matpotlib ga
Most Visited
GMail
GCal
py
PETSc
Conda build fram...
MoPad: Teaching ...
Mathjax
jupyter
Section1-2_matplotlib
File
Edit
View
Insert
Cell
Kernel
Help
Python 2
Code
Cell Toolbar: None
Intro to matplotlib
In [1]:
from __future__ import division, print_function
import numpy as np
import pandas as pd
In [2]:
import matplotlib.pyplot as plt
%matplotlib inline
The "magic" command is so the plots show up in output cells
pyplot is the plotting interface and is what is typically use
Tradition dictates it be called plt (like numpy is np)
User's Guide -- Example plots -- Official Tutorial

New Tab | Home | Home | Home | Home | Secti...

localhost:8889/notebooks/Section1-2_m | matpotlib ga

Most Visited | GMail | GCal | py | PETSc | Conda build fram... | MoPad: Teaching ... | Mathjax

jupyter Section1-2_matplotlib

File Edit View Insert Cell Kernel Help | Python 2

Markdown | Cell Toolbar: None

In [1]:
```
from __future__ import division, print_function
import numpy as np
import pandas as pd
```

In [2]:
```
import matplotlib.pyplot as plt
%matplotlib inline
# The "magic" command is so the plots show up in output cells
# pyplot is the plotting interface and is what is typically use
# Tradition dictates it be called plt (like numpy is np)
```

User's Guide -- Example plots -- Official Tutorial

Scatterplot (and state)

jupyter Section1-2_matplotlib

File Edit View Insert Cell Kernel Help Python 2

Markdown Cell Toolbar: None

```
%matplotlib inline
# The "magic" command is so the plots show up is output cells
# pyplot is the plotting interface and is what is typically use
# Tradition dictates it be called plt (like numpy is np)
```

User's Guide -- Example plots -- Official Tutorial

Scatterplot (and state)

```
In [3]: iris = pd.read_csv("iris.csv")
        iris.columns

Out[3]: Index([u'Unnamed: 0', u'SepalLength', u'SepalWidth', u'PetalLe
```

New Tab | Home | Home | Home | Home | Secti...

localhost:8889/notebooks/Section1-2_m | matpotlib ga

Most Visited | GMail | GCal | py | PETSc | Conda build fram... | MoPad: Teaching ... | MathJax

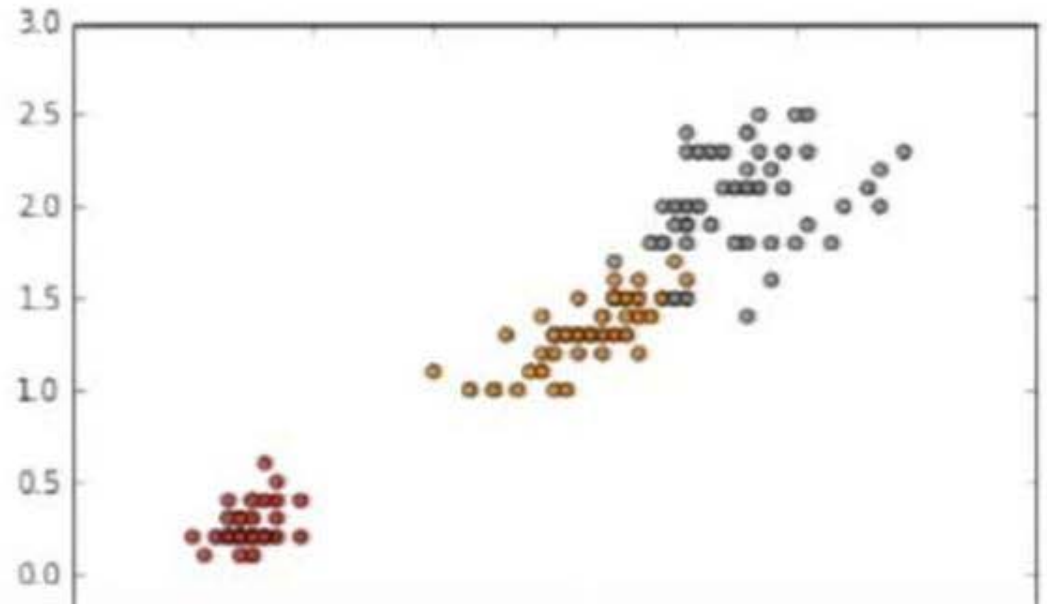
jupyter Section1-2_matplotlib

File Edit View Insert Cell Kernel Help | Python 2

Code | Cell Toolbar: None

In [4]:
```
#plt.scatter(iris.SepalLength, iris.SepalWidth)
#plt.scatter(iris.PetalLength, iris.PetalWidth, c=iris.Species.
plt.scatter(iris.PetalLength, iris.PetalWidth, c=iris.Species.f
```

Out[4]:
```
<matplotlib.collections.PathCollection at 0x10a5c8510>
```



localhost:8889/notebooks/Section1-2_m

matpotlib ga

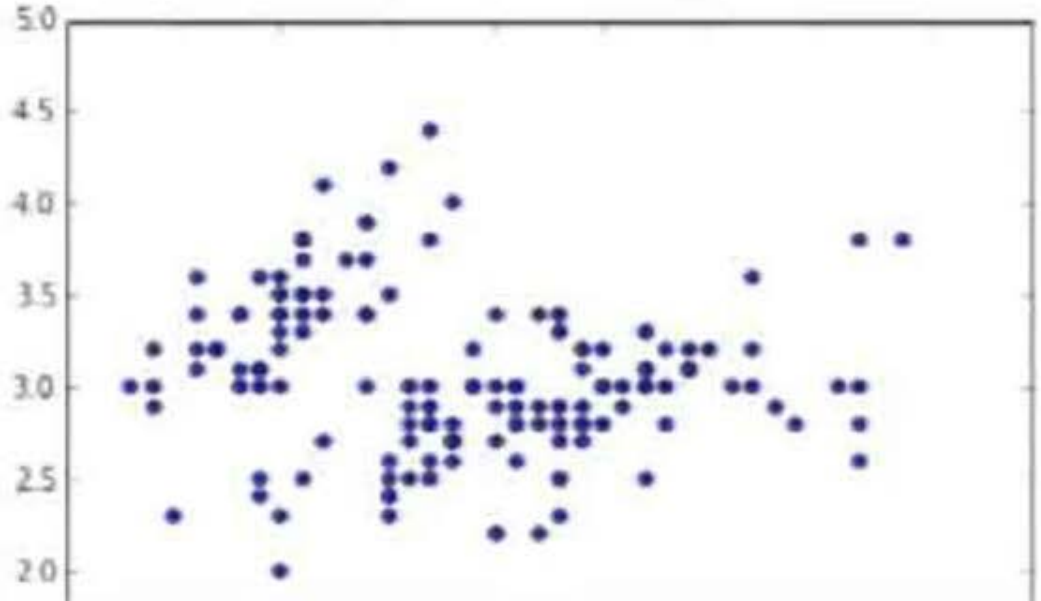
Most Visited | GMail | GCal | py | PETSc | Conda build fram... | MoPad: Teaching ... | MathJax

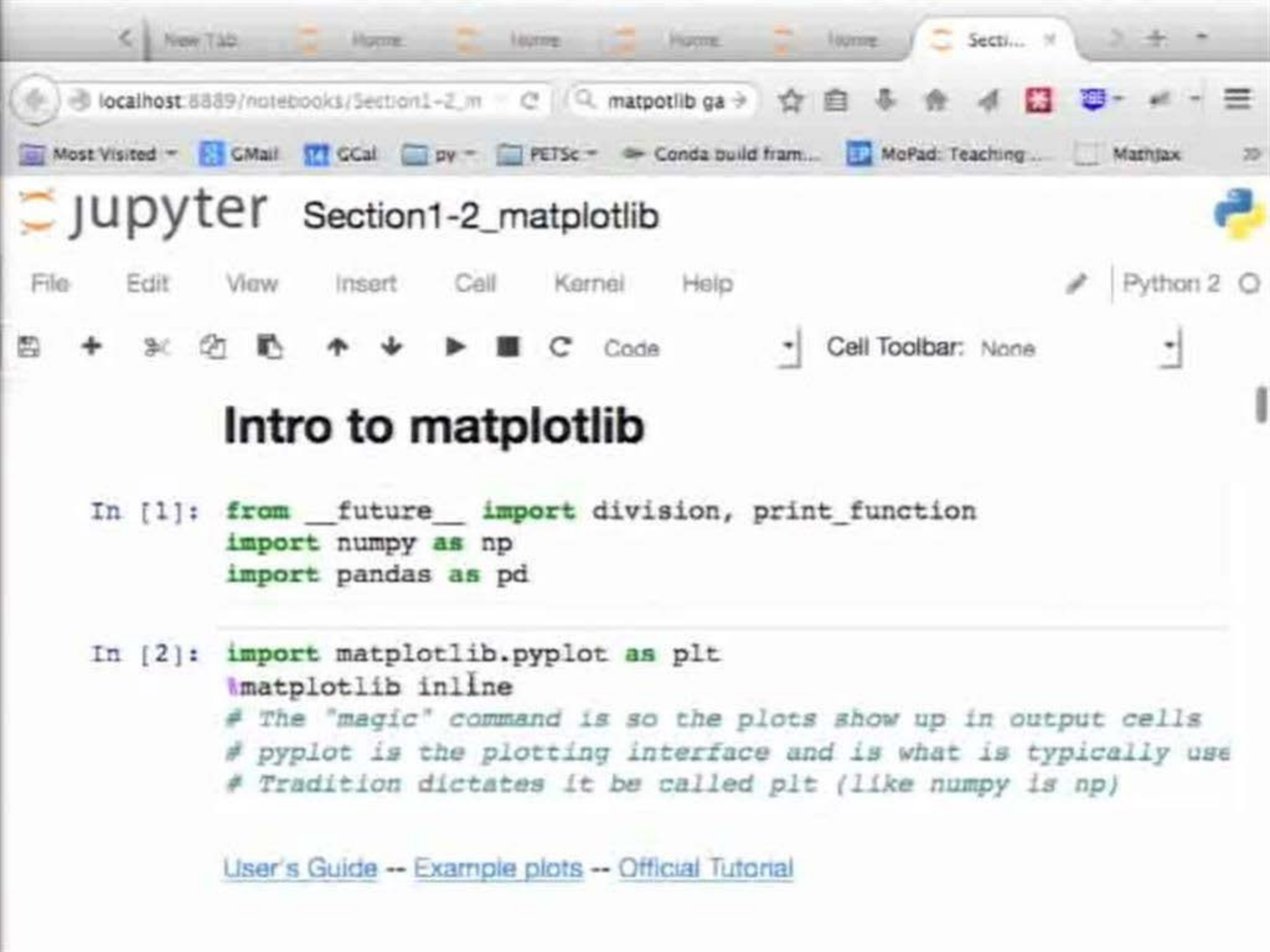
jupyter Section1-2_matplotlib

File Edit View Insert Cell Kernel Help Python 2

Code Cell Toolbar: None

```
In [5]: plt.scatter(iris.SepalLength, iris.SepalWidth)
        #plt.scatter(iris.PetalLength, iris.PetalWidth, c=iris.Species.
        #plt.scatter(iris.PetalLength, iris.PetalWidth, c=iris.Species.

Out[5]: <matplotlib.collections.PathCollection at 0x10b72c0d0>
```




localhost:8889/notebooks/Section1-2_m
matpotlib ga
Most Visited
GMail
GCal
py
PETSc
Conda build fram...
MoPad: Teaching ...
Mathjax
jupyter
Section1-2_matplotlib
File
Edit
View
Insert
Cell
Kernel
Help
Python 2
Code
Cell Toolbar: None
Intro to matplotlib
In [1]:
from __future__ import division, print_function
import numpy as np
import pandas as pd
In [2]:
import matplotlib.pyplot as plt
%matplotlib inline
The "magic" command is so the plots show up in output cells
pyplot is the plotting interface and is what is typically use
Tradition dictates it be called plt (like numpy is np)
User's Guide -- Example plots -- Official Tutorial

New Tab | Home | Home | Home | Home | Secti...

localhost:8889/notebooks/Section1-2_m

matpotlib ga

Most Visited | GMail | GCal | py | PETSc | Conda build fram... | MoPad: Teaching ... | Mathjax

jupyter Section1-2_matplotlib

File Edit View Insert Cell Kernel Help Python 2

Code Cell Toolbar: None

In [7]:
```
iris.Species.factorize()
```

Out[7]:
```
(array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0,
           0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0,
           0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1,
           1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1,
           1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
        2, 2, 2, 2, 2,
           2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
        2, 2, 2, 2, 2,
           2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]),
 Index([u'setosa', u'versicolor', u'virginica'],
dtype='object'))
```

jupyter Section1-2_matplotlib

File Edit View Insert Cell Kernel Help Python 2

Code Cell Toolbar: None

```
1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2,
        2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2,
        2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]),
 Index([u'setosa', u'versicolor', u'virginica'],
dtype='object'))
```

In [6]:
```
ris.SepalLength, iris.SepalWidth)
is.PetalLength, iris.PetalWidth, c=iris.Species.factorize()[0])
ris.PetalLength, iris.PetalWidth, c=iris.Species.factorize()[0],
```

Out[6]: `<matplotlib.collections.PathCollection at 0x10b946650>`

jupyter Section1-2_matplotlib

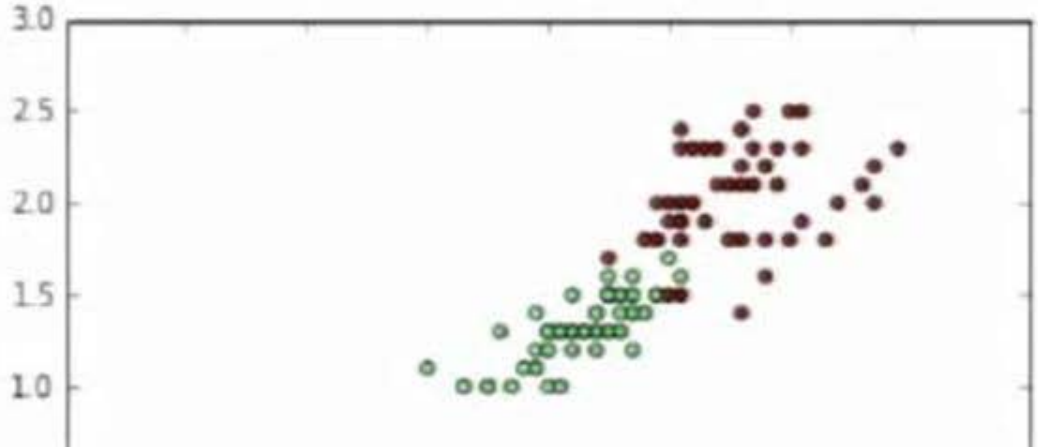
File Edit View Insert Cell Kernel Help Python 2

Code Cell Toolbar: None

```
dtype='object'))
```

In [6]:
```
s.SepalLength, iris.SepalWidth)
.PetalLength, iris.PetalWidth, c=iris.Species.factorize()[0])
s.PetalLength, iris.PetalWidth, c=iris.Species.factorize()[0],
```

Out[6]: <matplotlib.collections.PathCollection at 0x10b946650>

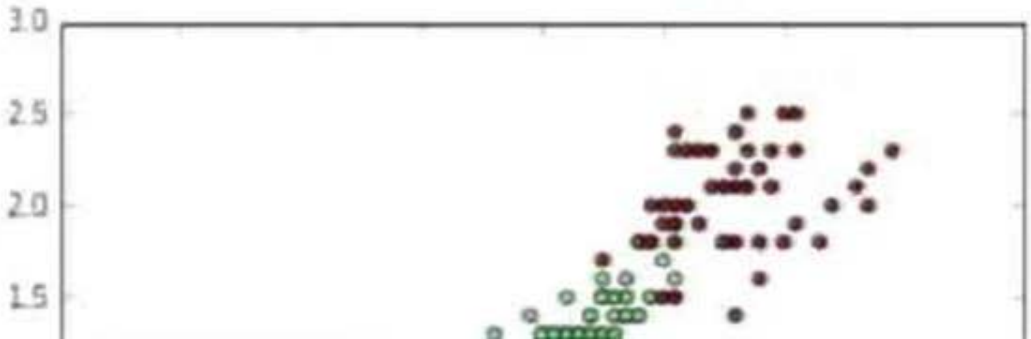


jupyter Section1-2_matplotlib


```
        2, 2, 2, 2, 2,
               2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]},
 Index([u'setosa', u'versicolor', u'virginica'],
dtype='object'))
```

In [6]:
```
s.SepalLength, iris.SepalWidth)
.PetalLength, iris.PetalWidth, c=iris.Species.factorize()[0])
s.PetalLength, iris.PetalWidth, c=iris.Species.factorize()[0],
```

Out[6]: `<matplotlib.collections.PathCollection at 0x10b946650>`



jupyter Section1-2_matplotlib

In [5]:
```
iris.Species.unique()
```

Out[5]:
```
array(['setosa', 'versicolor', 'virginica'], dtype=object)
```

matplotlib is a **stateful** library. State by default accumulates within the cell, so it is actually making several plots at once back in the prior cell. That is why there are **four** colors on the above plot, one supplied by the earlier plot!

In [6]:
```
#plt.scatter(iris.SepalLength, iris.SepalWidth)
#plt.scatter(iris.PetalLength, iris.PetalWidth, c=iris.Species.
```

jupyter Section1-2_matplotlib

File Edit View Insert Cell Kernel Help Python 2

Code Cell Toolbar: None

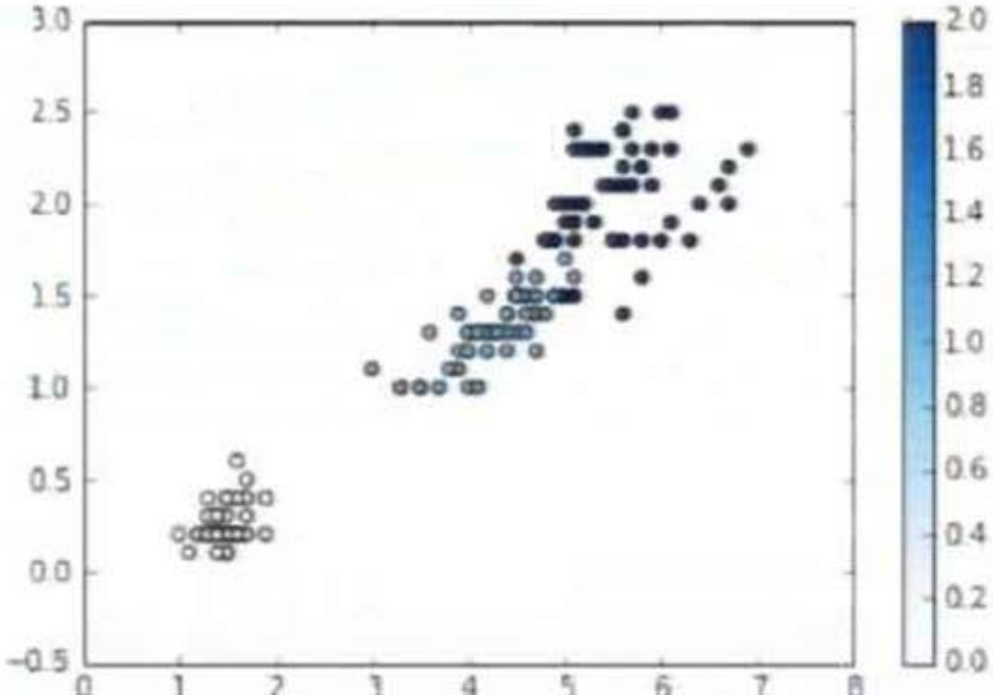
```
In [10]: gth, iris.SepalWidth)
         gth, iris.PetalWidth, c=iris.Species.factorize()[0])
         th, iris.PetalWidth, c=iris.Species.factorize()[0], cmap="Blus"


         ---------------------------------------------------------------
         -------------
         ValueError                                Traceback (most rece
         nt call last)
         <ipython-input-10-def73b910b93> in <module>()
               1 #plt.scatter(iris.SepalLength, iris.SepalWidth)
               2 #plt.scatter(iris.PetalLength, iris.PetalWidth, c=iris
         .Species.factorize()[0])
         ----> 3 plt.scatter(iris.PetalLength, iris.PetalWidth, c=iris.
         Species.factorize()[0], cmap="Blus")
               4 plt.colorbar()
```

jupyter Section1-2_matplotlib

File Edit View Insert Cell Kernel Help | Python 2

Markdown | Cell Toolbar: None

Out[11]: <matplotlib.colorbar.Colorbar instance at 0x10c898560>



jupyter Section1-2_matplotlib

In [5]:
```
iris.Species.unique()
```

Out[5]:
```
array(['setosa', 'versicolor', 'virginica'], dtype=object)
```

matplotlib is a **stateful** library. State by default accumulates within the cell, so it is actually making several plots at once back in the prior cell. That is why there are **four** colors on the above plot, one supplied by the earlier plot!

In [11]:
```
th, iris.SepalWidth)
th, iris.PetalWidth, c=iris.Species.factorize()[0])
h, iris.PetalWidth, c=iris.Species.factorize()[0], cmap="Blues")
```

Out[11]:
```
<matplotlib.colorbar.Colorbar instance at 0x10c898560>
```

Home | Home | Home | Home | Secti... | http://...g.html

localhost:8889/notebooks/Section1-2_m | matpotlib ga

Most Visited | GMail | GCal | py | PETSc | Conda build fram... | MoPad: Teaching ... | Mathjax

jupyter Section1-2_matplotlib

File Edit View Insert Cell Kernel Help | Python 2

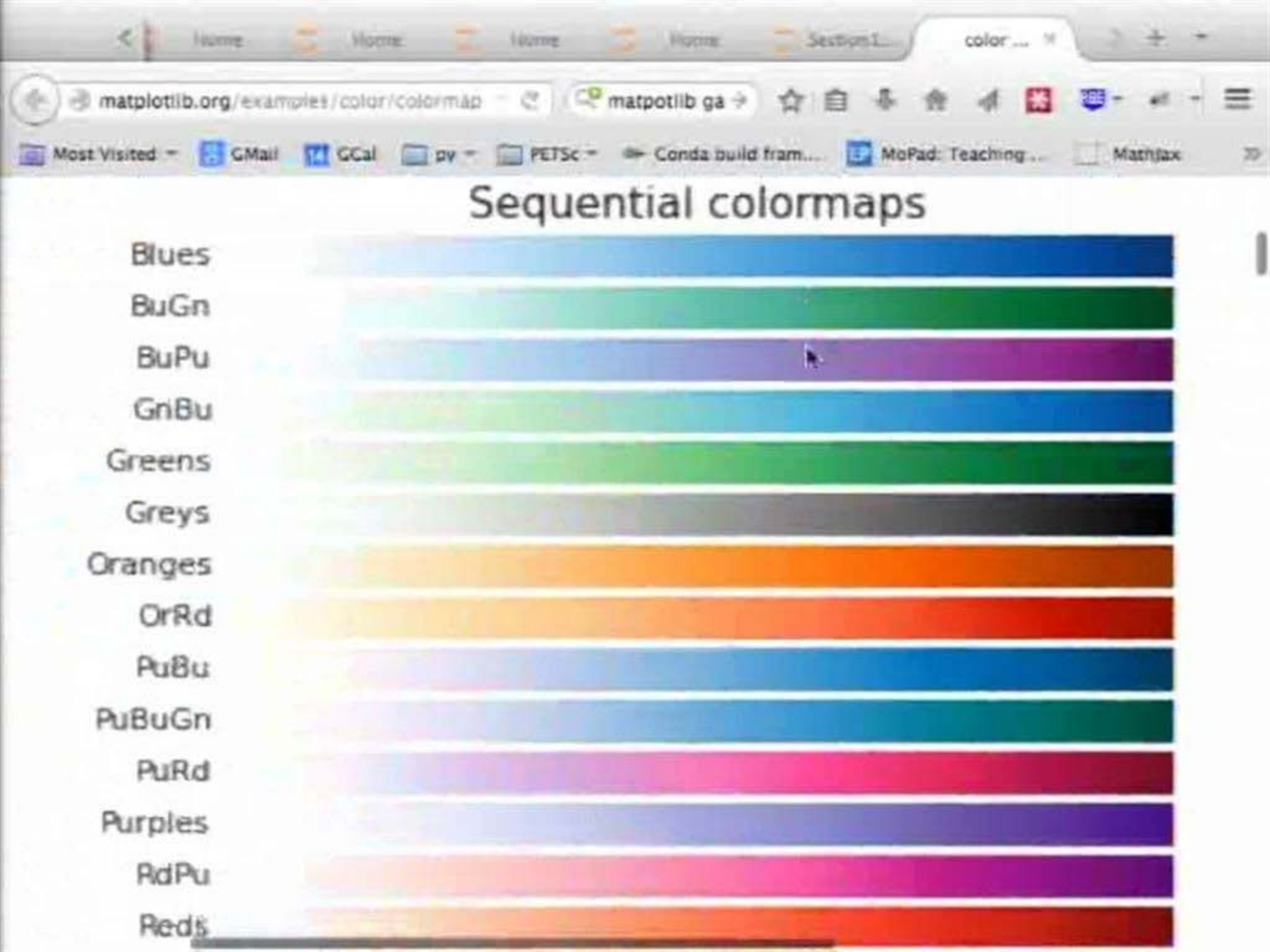
Markdown | Cell Toolbar: None

State is used to build up more complex plots, layering and adding legends, grids and other guides.

matplotlib colormaps map from inputs in 0-1 to a set of colors

In [ ]:

In [7]:
```
cmap = plt.get_cmap("Set1")
species = iris.Species.unique()
for (i, spec) in enumerate(species):
    subset = iris[iris.Species==spec]
    plt.scatter(subset.PetalLength, subset.PetalWidth, c=cmap(i
plt.legend(loc="lower right")
plt.grid()
```


matplotlib.org/examples/color/colormap
matpotlib ga
Most Visited
GMail
GCal
py
PETSc
Conda build fram...
MoPad: Teaching ...
MathJax
Sequential colormaps
Blues
BuGn
BuPu
GnBu
Greens
Greys
Oranges
OrRd
PuBu
PuBuGn
PuRd
Purples
RdPu
Reds

matplotlib.org/examples/color/colormap

matpotlib ga

Most Visited | GMail | GCal | py | PETSc | Conda build fram... | MoPad: Teaching ... | MathJax

YlGn

YlGnBu

YlOrBr

YlOrRd

ng, hires.png, pdf)

Sequential (2) colormaps

afmhot

autumn

bone

cool

copper

gist_heat

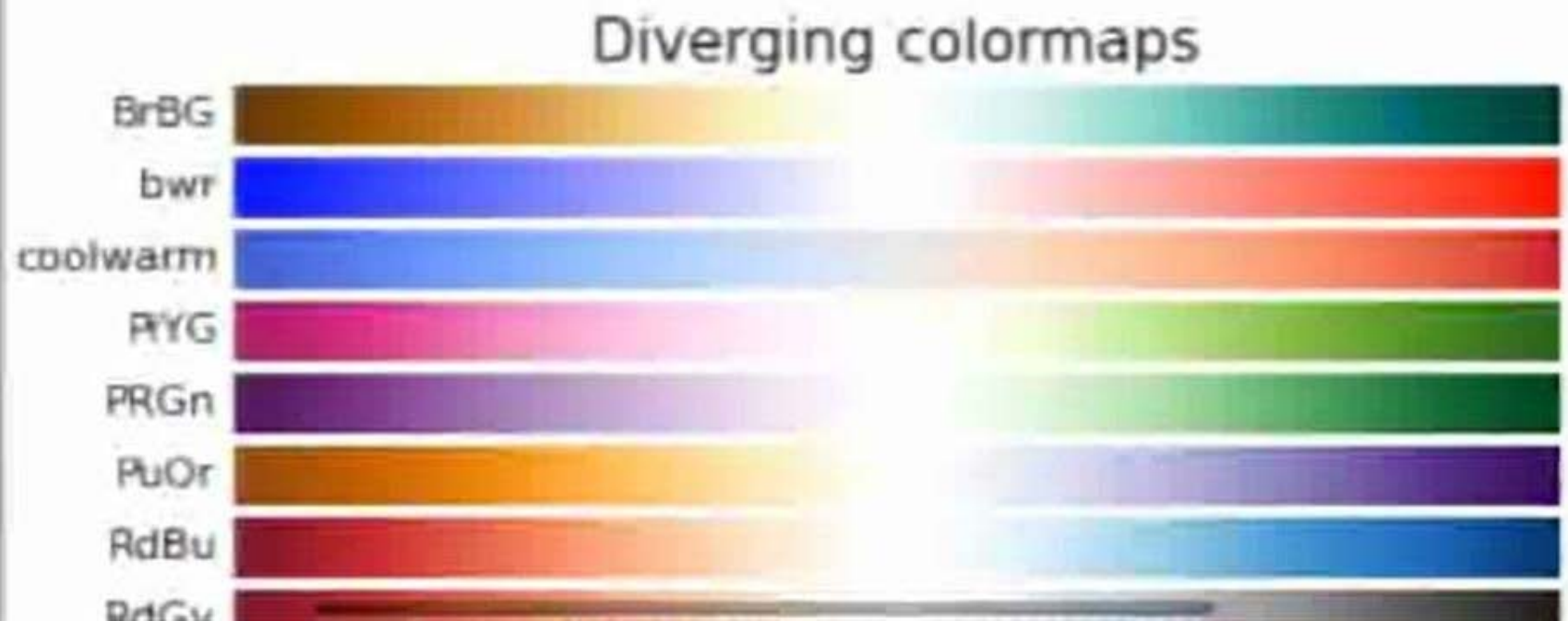
gray

hot

matplotlib.org

matplotlib ga

, hires.png, pdf)



Home | Home | Home | Home | Secti... | color ex...

localhost:8889/notebooks/Section1-2_m | matpotlib ga

Most Visited | GMail | GCal | py | PETSc | Conda build fram... | MoPad: Teaching ... | MathJax

jupyter Section1-2_matplotlib

File Edit View Insert Cell Kernel Help | Python 2

Markdown | Cell Toolbar: None

State is used to build up more complex plots, layering and adding legends, grids and other guides.

matplotlib colormaps map from inputs in 0-1 to a set of colors

In [ ]:

In [7]:
```
cmap = plt.get_cmap("Set1")
species = iris.Species.unique()
for (i, spec) in enumerate(species):
    subset = iris[iris.Species==spec]
    plt.scatter(subset.PetalLength, subset.PetalWidth, c=cmap(i
plt.legend(loc="lower right")
plt.grid()
```

Home | Home | Home | Home | Secti... | color ex...

localhost:8889/notebooks/Section1-2_m | matpotlib ga

Most Visited | GMail | GCal | py | PETSc | Conda build fram... | MoPad: Teaching ... | Mathjax

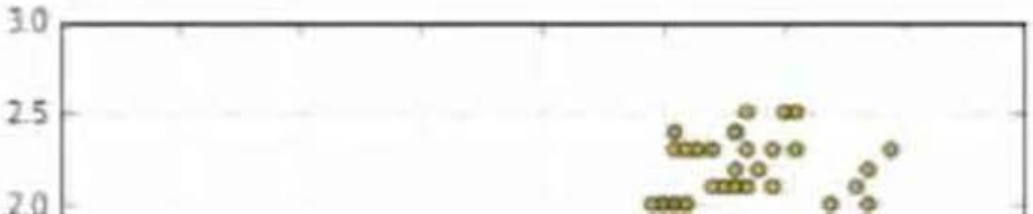
jupyter Section1-2_matplotlib

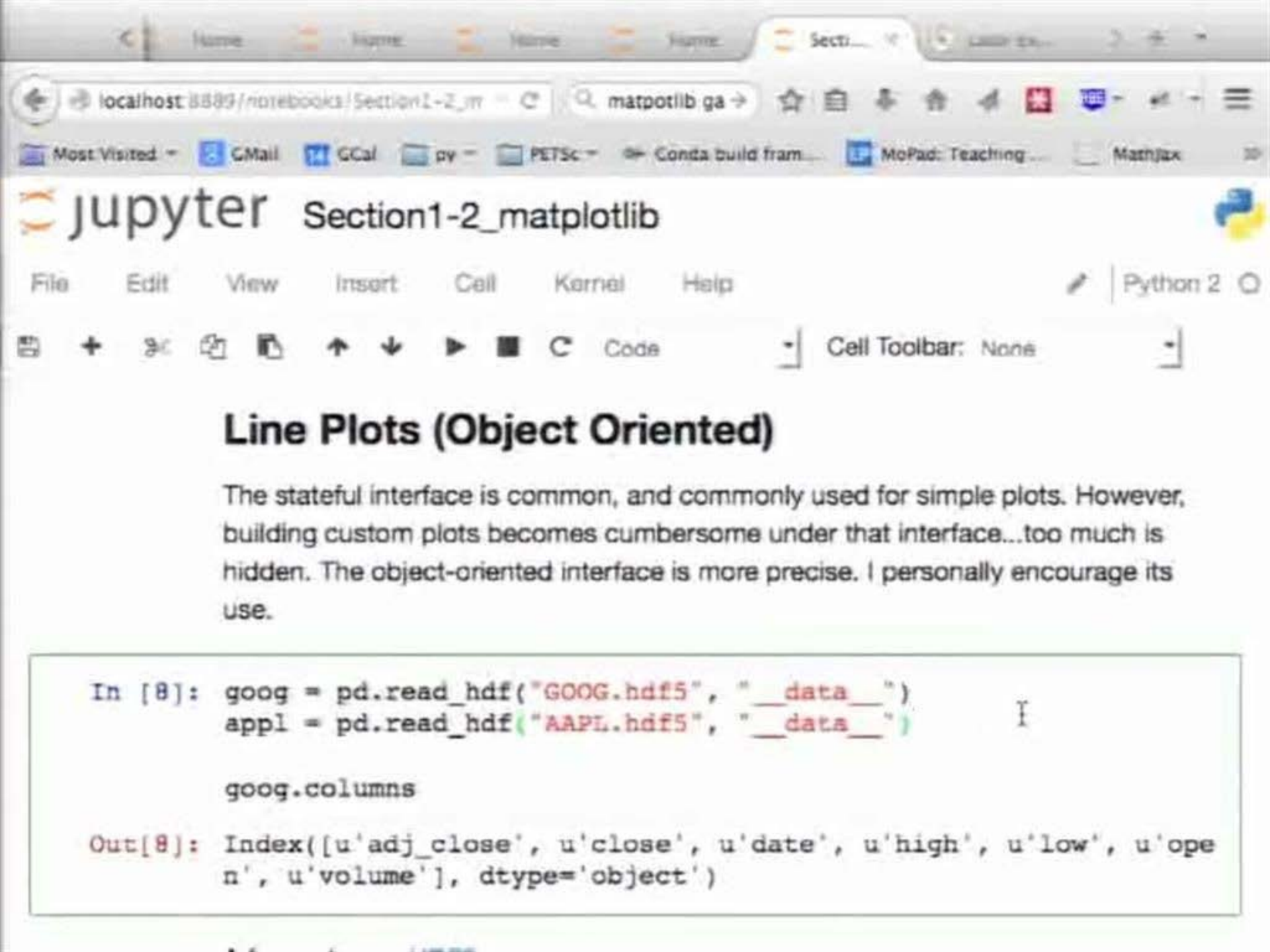
File Edit View Insert Cell Kernel Help | Python 2

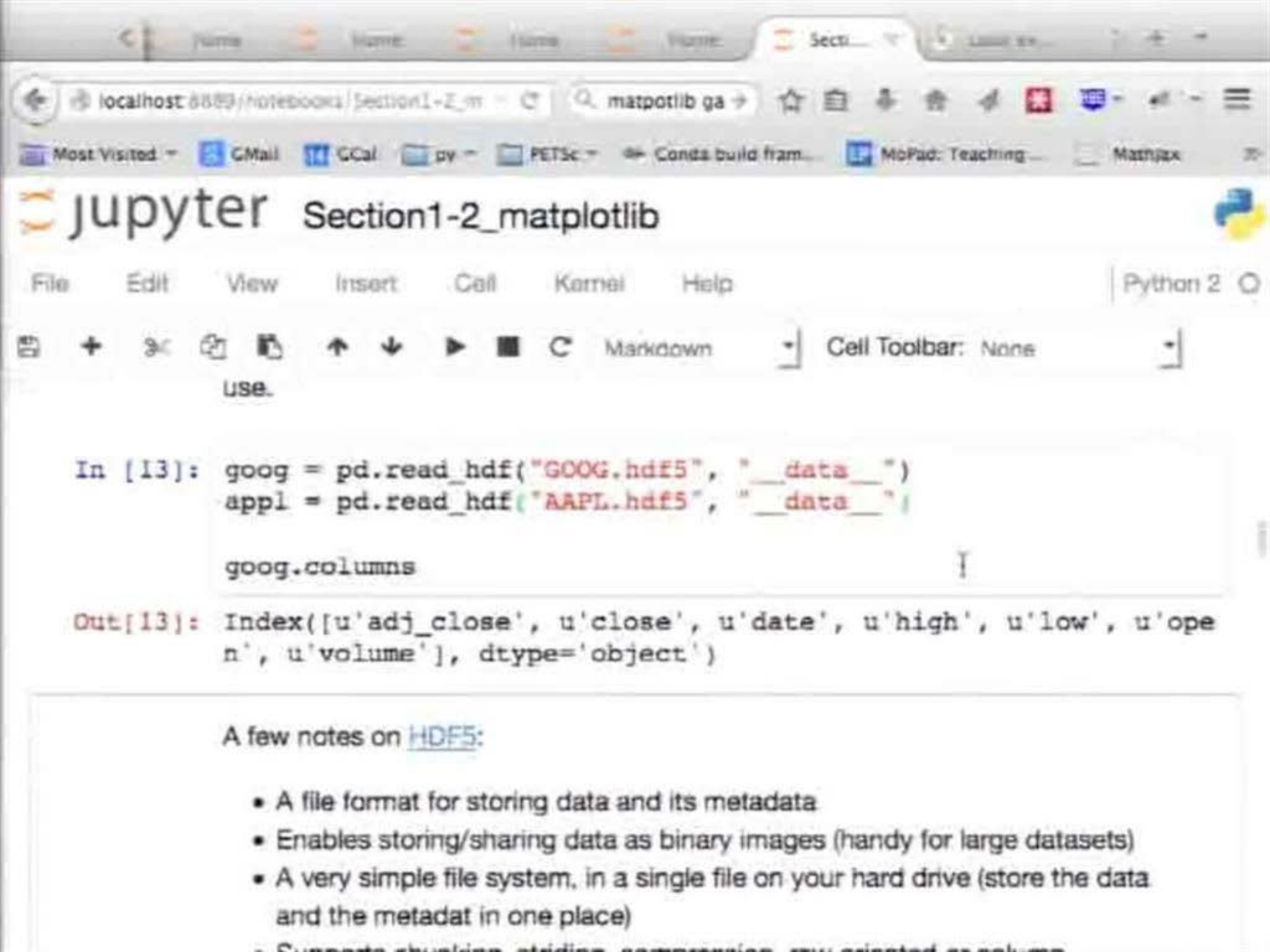
Code | Cell Toolbar: None

In [ ]:

In [7]:
```
cmap = plt.get_cmap("Set1")
species = iris.Species.unique()
for (i, spec) in enumerate(species):
    subset = iris[iris.Species==spec]
    plt.scatter(subset.PetalLength, subset.PetalWidth, c=cmap(i
plt.legend(loc="lower right")
plt.grid()
```




localhost:8889/notebooks/Section1-2_m
matplotlib ga
Most Visited
GMail
GCal
py
PETSc
Conda build fram...
MoPad: Teaching ...
MathJax
jupyter
Section1-2_matplotlib
File
Edit
View
Insert
Cell
Kernel
Help
Python 2
Code
Cell Toolbar: None
Line Plots (Object Oriented)
The stateful interface is common, and commonly used for simple plots. However, building custom plots becomes cumbersome under that interface...too much is hidden. The object-oriented interface is more precise. I personally encourage its use.
In [8]:
goog = pd.read_hdf("GOOG.hdf5", "__data__")
appl = pd.read_hdf("AAPL.hdf5", "__data__")
goog.columns
Out[8]:
Index([u'adj_close', u'close', u'date', u'high', u'low', u'open', u'volume'], dtype='object')


localhost:8889/notebooks/Section1-2_m
matpotlib ga
Most Visited
GMail
GCal
py
PETSc
Conda build fram...
MoPad: Teaching...
Mathjax
jupyter
Section1-2_matplotlib
File
Edit
View
Insert
Cell
Kernel
Help
Python 2
Markdown
Cell Toolbar: None
use.
In [13]:
goog = pd.read_hdf("GOOG.hdf5", "__data__")
appl = pd.read_hdf("AAPL.hdf5", "__data__")
goog.columns
Out[13]:
Index([u'adj_close', u'close', u'date', u'high', u'low', u'open', u'volume'], dtype='object')
A few notes on HDF5:
• A file format for storing data and its metadata
• Enables storing/sharing data as binary images (handy for large datasets)
• A very simple file system, in a single file on your hard drive (store the data and the metadat in one place)

jupyter Section1-2_matplotlib


```
goog.columns
```

```
Out[13]: Index([u'adj_close', u'close', u'date', u'high', u'low', u'ope
n', u'volume'], dtype='object')
```

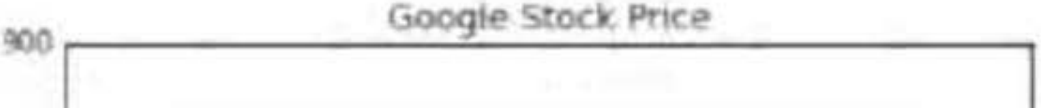
A few notes on HDF5:

- A file format for storing data and its metadata
- Enables storing/sharing data as binary images (handy for large datasets)
- A very simple file system, in a single file on your hard drive (store the data and the metadat in one place)
- Supports chunking, striding, compression, row oriented or column oriented...if you know how to ask
- Read/written by many analysis libraries

jupyter Section1-2_matplotlib

- Enables storing/sharing data as binary images (handy for large datasets)
- A very simple file system, in a single file on your hard drive (store the data and the metadat in one place)
- Supports chunking, striding, compression, row oriented or column oriented...if you know how to ask
- Read/written by many analysis libraries

In [9]:
```
fig,axes = plt.subplots()
axes.plot(goog.date, goog.close)
axes.set_xlabel("date")
axes.set_ylabel("price")
axes.set_title("Google Stock Price")
plt.show()
```



jupyter Section1-2_matplotlib

- Read/written by many analysis libraries

In [9]:
```
fig,axes = plt.subplots()
axes.plot(goog.date, goog.close)
axes.set_xlabel("date")
axes.set_ylabel("price")
axes.set_title("Google Stock Price")
plt.show()
```

Home | Home | Home | Home | Secti... | color ex...

localhost:8889/notebooks/Section1-2_m | matplotlib ga

Most Visited | GMail | GCal | py | PETSc | Conda build fram... | MoPad: Teaching ... | MathJax

jupyter Section1-2_matplotlib

File Edit View Insert Cell Kernel Help | Python 2

Code | Cell Toolbar: None

```
axes.set_xlabel("date")
axes.set_ylabel("price")
axes.set_title("Google Stock Price")
plt.show()
```

Home | Home | Home | Home | Secti... | color ex...

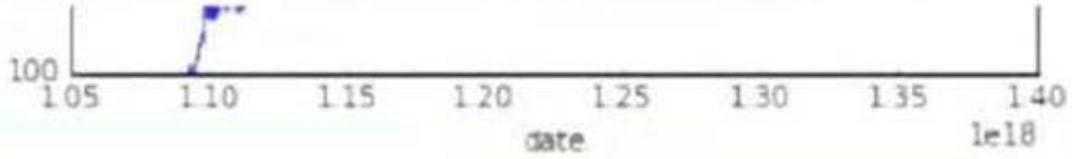
localhost:8889/notebooks/Section1-2_m | matplotlib ga

Most Visited | GMail | GCal | py | PETSc | Conda build fram... | MoPad: Teaching ... | MathJax

jupyter Section1-2_matplotlib

File Edit View Insert Cell Kernel Help Python 2

Code | Cell Toolbar: None



In [10]:
```
from datetime import datetime

fig,axes = plt.subplots(figsize=(12,3))  #figsize is in inches,
axes.plot(goog.date.astype(datetime), goog.close)
axes.set_xlabel("date")
axes.set_ylabel("price")
axes.set_title("Google Stock Price")
plt.show()
```

jupyter Section1-2_matplotlib


```
axes.hist(goog.close)
axes.set_title('Histogram of closing prices');
```

Home | Home | Home | Home | Secti... | color ex...

localhost:8889/notebooks/Section1-2_m | matpotlib ga

Most Visited | GMail | GCal | py | PETSc | Conda build fram... | MoPad: Teaching ... | MathJax

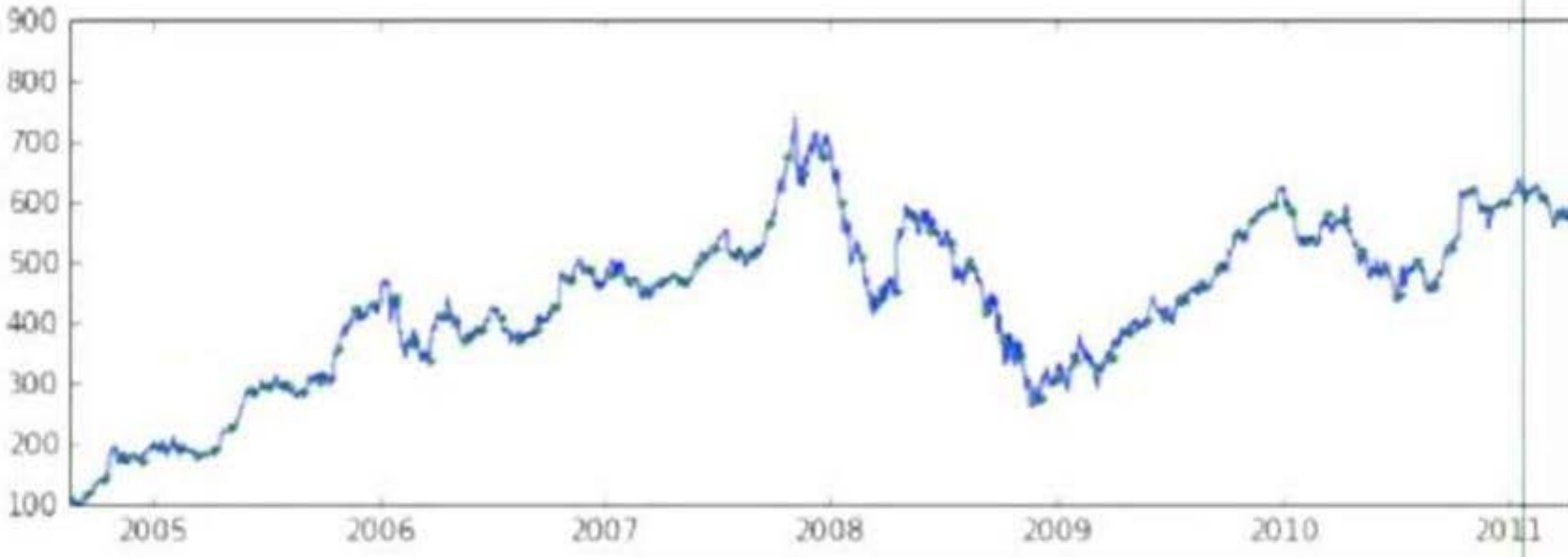
jupyter Section1-2_matplotlib

File Edit View Insert Cell Kernel Help | Python 2

Code | Cell Toolbar: None

```
axes.plot(goog.date.astype(datetime), goog.close)
axes.plot(goog.date[::20], goog.close[::20], ".g:")
```

Out[20]: [<matplotlib.lines.Line2D at 0x10de45590>]

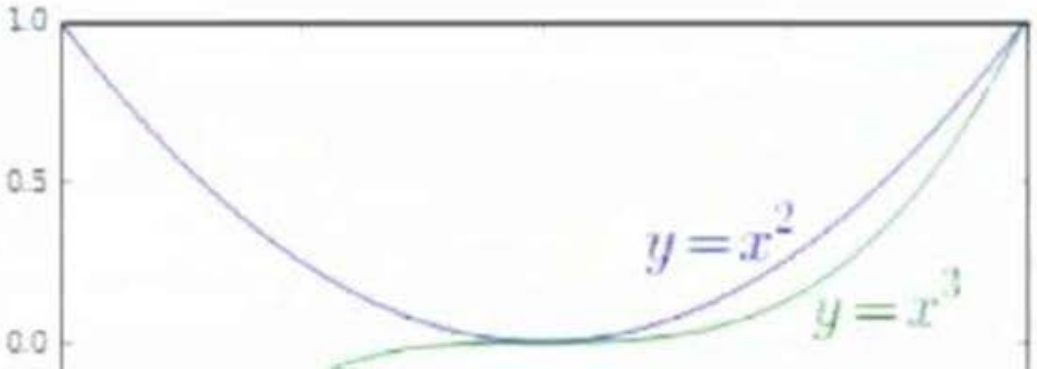


jupyter Section1-2_matplotlib

File Edit View Insert Cell Kernel Help Python 2

Code Cell Toolbar: None

In [22]:
```
fig, ax = plt.subplots()

xx = np.linspace(-1., 1., 100)
ax.plot(xx, xx**2)
ax.plot(xx, xx**3)

ax.text(0.2, 0.25, r"$y=x^2$", fontsize=22, color="b")   #x/y p
ax.text(0.55, 0.05, r"$y=x^3$", fontsize=22, color="g");
```



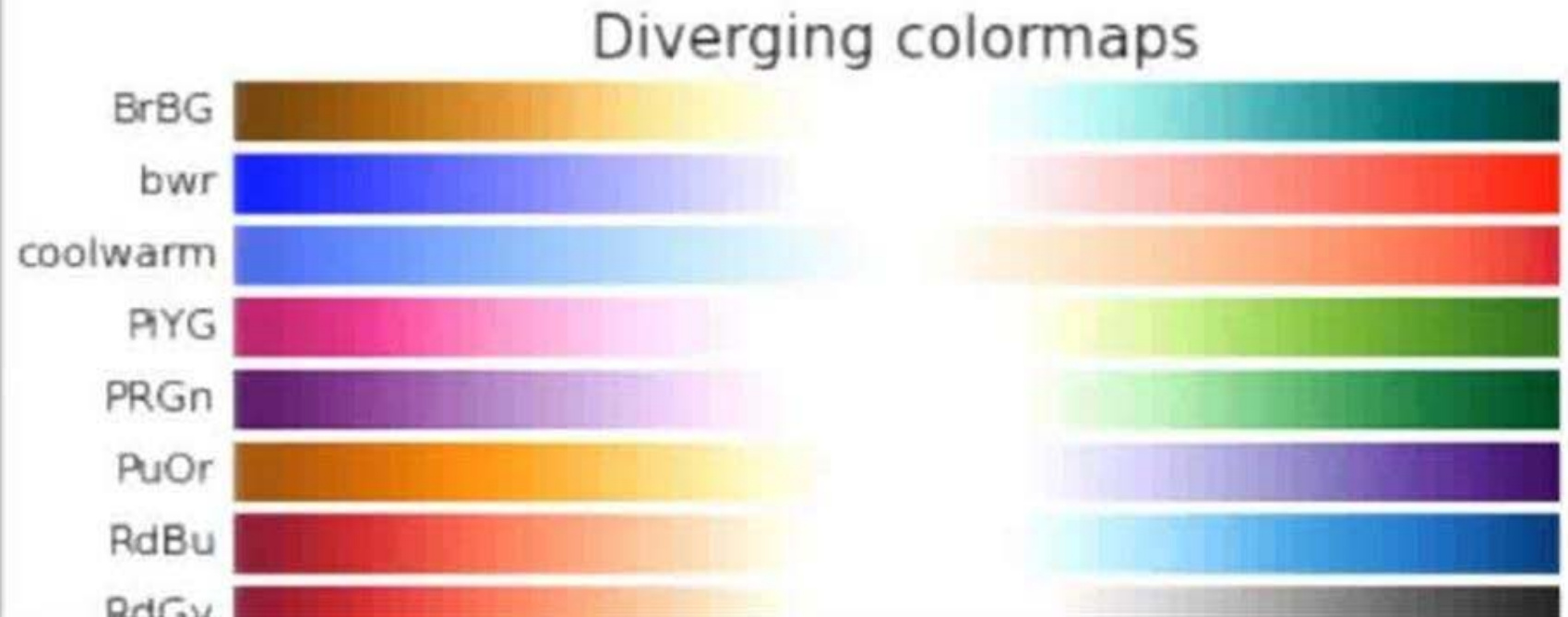
Home | Home | Home | Home | Section1... | color ...

matplotlib.org/examples/color/colormap

matpotlib ga

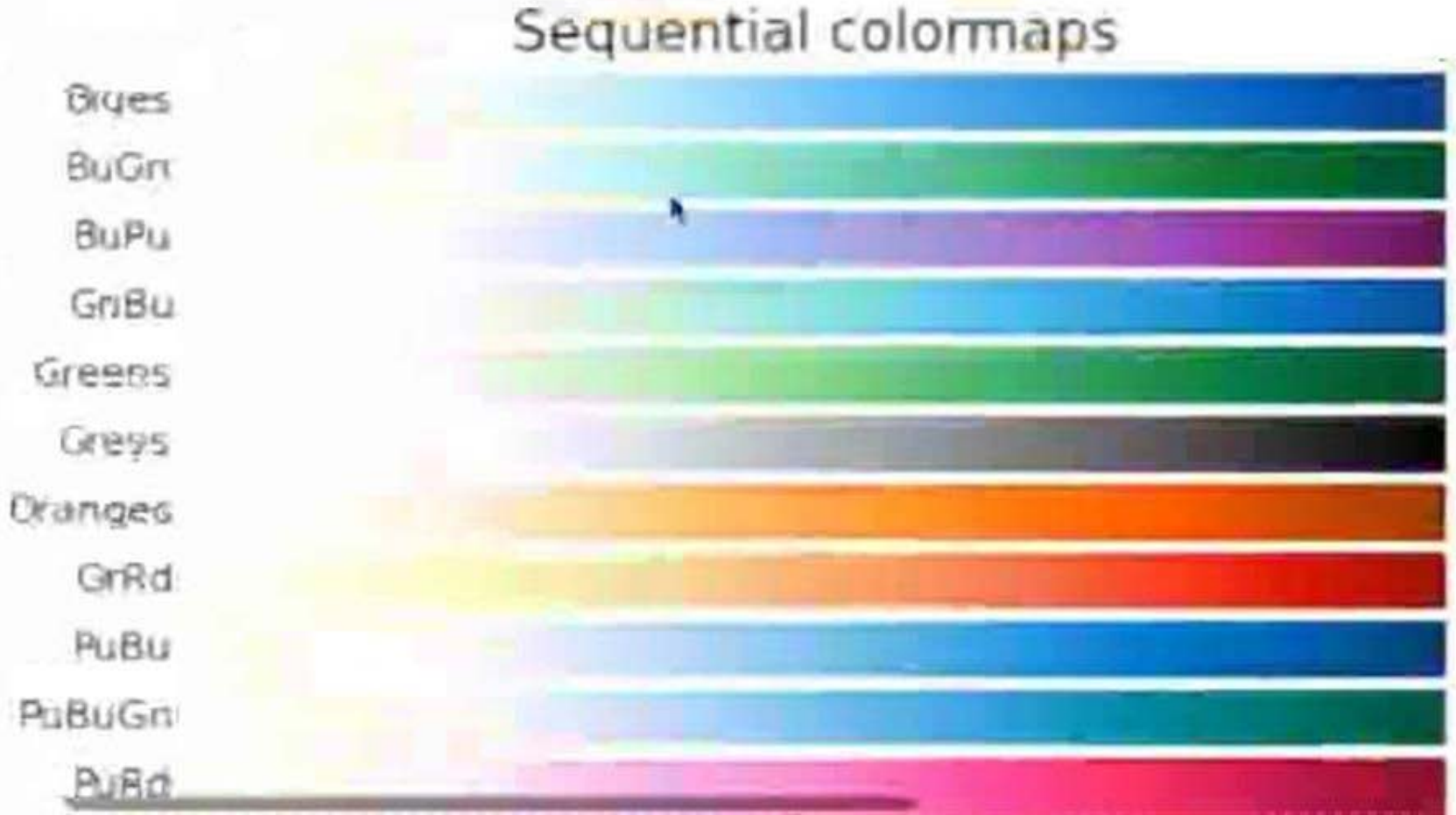
Most Visited | GMail | GCal | py | PETSc | Conda build fram... | MoPad: Teaching ... | MathJax

, hires.png, pdf)



colormaps_reference.py

(Source code)



Home | Home | Home | Home | Secti... | color ex...

localhost:8889/notebooks/Section1-2_m

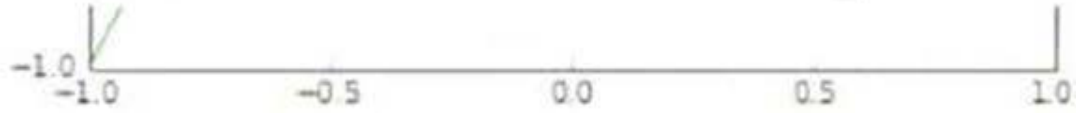
matpotlib ga

Most Visited | GMail | GCal | py | PETSc | Conda build fram... | MoPad: Teaching ... | Mathjax

jupyter Section1-2_matplotlib

File Edit View Insert Cell Kernel Help Python 2

Code | Cell Toolbar: None



In [18]:
```
#adapted from http://nbviewer.ipython.org/github/jrjohansson/sc
alpha = 0.7
phi_ext = 2 * np.pi * 0.5

def flux_qubit_potential(phi_m, phi_p):
    return 2 + alpha - 2 * np.cos(phi_p)*np.cos(phi_m) - alpha

phi_m = np.linspace(0, 2*np.pi, 100)
phi_p = np.linspace(0, 2*np.pi, 100)
X,Y = np.meshgrid(phi_p, phi_m)
Z = flux_qubit_potential(X, Y).T

fig, ax = plt.subplots(ncols=2, figsize=(10,4))
```

jupyter Section1-2_matplotlib

In [ ]:
```
# %load http://matplotlib.org/mpl_examples/color/colormaps_refe
"""
Reference for colormaps included with Matplotlib.

This reference example shows all colormaps included with Matplc
any colormap listed here can be reversed by appending "_r" (e.g
These colormaps are divided into the following categories:

Sequential:
    These colormaps are approximately monochromatic colormaps v
    between two color tones---usually from low saturation (e.g.
    saturation (e.g. a bright blue). Sequential colormaps are i
    representing most scientific data since they show a clear p
    low-to-high values.
```

Home | Home | Home | Home | (Busy... | color ex...

localhost:8889/notebooks/Section1-2_m | matpotlib ga

Most Visited | GMail | GCal | py | PETSc | Conda build fram... | MoPad: Teaching ... | Mathjax

jupyter Section1-2_matplotlib

File Edit View Insert Cell Kernel Help | Python 2

Code | Cell Toolbar: None

```
    for cmap_category, cmap_list in cmaps:
        plot_color_gradients(cmap_category, cmap_list)

    plt.show()
```

In [18]:
```
#adapted from http://nbviewer.ipython.org/github/jrjohansson/sc
alpha = 0.7
phi_ext = 2 * np.pi * 0.5

def flux_qubit_potential(phi_m, phi_p):
    return 2 + alpha - 2 * np.cos(phi_p)*np.cos(phi_m) - alpha

phi_m = np.linspace(0, 2*np.pi, 100)
phi_p = np.linspace(0, 2*np.pi, 100)
X,Y = np.meshgrid(phi_p, phi_m)
Z = flux_qubit_potential(X, Y).T
```