A very short intro to the IPython Notebook

By Lynn Cherny (lynn@ghostweather.com) for Python Data Science Afternoon 12/2/12

Startup

To start up the notebook server, you want to be IN a directory with notebooks you want to edit. The file list that's visible is the directory you start the server from.

Start it with this command:

ipython notebook --pylab inline

You will see the server info as it starts, and a browser tab will open showing you a notebook homepage.

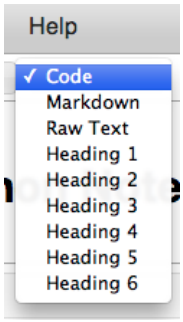
What They Contain

IPython notebooks can contain text, images (see below-soon this will be less codelike and more html-like), and executable code + output. The notebook also features a full interactive ipython environment, which is super helpful during dev.

In [33]:
```
#  This is excutable code.  That's the default.  I'm using some code to load an image here:

from IPython.core.display import Image
Image(filename='screencaps/code_menu.png')
# up above is a little menu that lets you change your cell to text or code
```

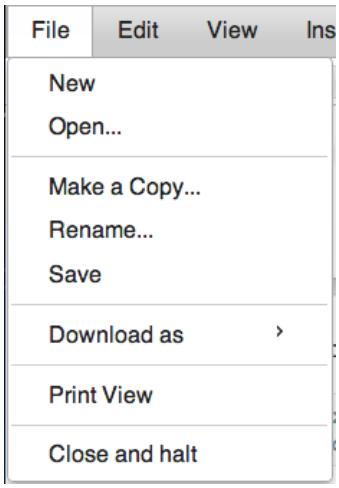
Out[33]:



This is a markdown cell with plain text. (Double click it to get edit access.) Here's a link to a file with markdown help:
http://daringfireball.net/projects/markdown/

In [34]:
```
from IPython.core.display import Image
Image(filename='screencaps/file_menu.png')
```

Out[34]:



"Rename" is very useful. And don't forget to save frequently.

Make sure to look at the other menus and hover over the icons on the top.

In [12]:
```
ls   # many useful commands work in the ipython shell, even in the notebook.
```

```
IPythonNotebooks.ipynb  ipython_magic.png
MovielensClean2.ipynb   ml-1m/
```

In [35]:
```
from IPython.core.display import Image
Image(filename='screencaps/ipython_magic.png')
```

Out[35]:

Table 3-2. Frequently-used IPython Magic Commands

Command	Description
%quickref	Display the IPython Quick Reference Card
%magic	Display detailed documentation for all of the available magic commands
%debug	Enter the interactive debugger at the bottom of the last exception traceback
%hist	Print command input (and optionally output) history
%pdb	Automatically enter debugger after any exception
%paste	Execute pre-formatted Python code from clipboard
%cpaste	Open a special prompt for manually pasting Python code to be executed
%reset	Delete all variables / names defined in interactive namespace
%page *OBJECT*	Pretty print the object and display it through a pager
%run *script.py*	Run a Python script inside IPython
%prun *statement*	Execute *statement* with cProfile and report the profiler output
%time *statement*	Report the execution time of single statement
%timeit *statement*	Run a statement multiple times to compute an emsemble average execution time. Useful for timing code with very short execution time
%who, %who_ls, %whos	Display variables defined in interactive namespace, with varying levels of information / verbosity
%xdel *variable*	Delete a variable and attempt to clear any references to the object in the IPython internals

In [9]:
```
%magic   # execute this call by using "shift-enter"
# Note: you can close the window that opens at the bottom by clicking on the separator bar.
```

In [26]:
```
# Be sure to try tab-completion. Define a string variable first:
mystring = "hi there"
```

In [27]:
```
# Now try adding a . after mystring and hitting tab after you do --
mystring.
```

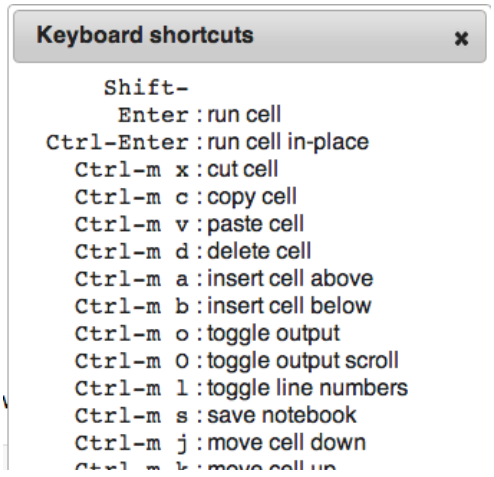
```
  File "<ipython-input-27-1d40cd9b3212>", line 2
    mystring.
             ^
SyntaxError: invalid syntax
```

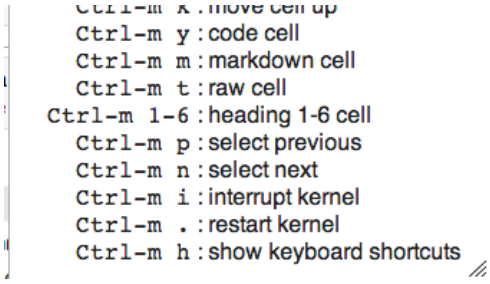
In [29]:
```
# Try the ? help too.
mystring.capitalize?
# You can close the window that opens with a click on the separator bar!
```

In [37]:
```
from IPython.core.display import Image
Image(filename='screencaps/keyboard.png')

# These are the keyboard shortcuts that you can use inside the notebook:
```

Out[37]:





Some Cool (and Important) Sharing Stuff

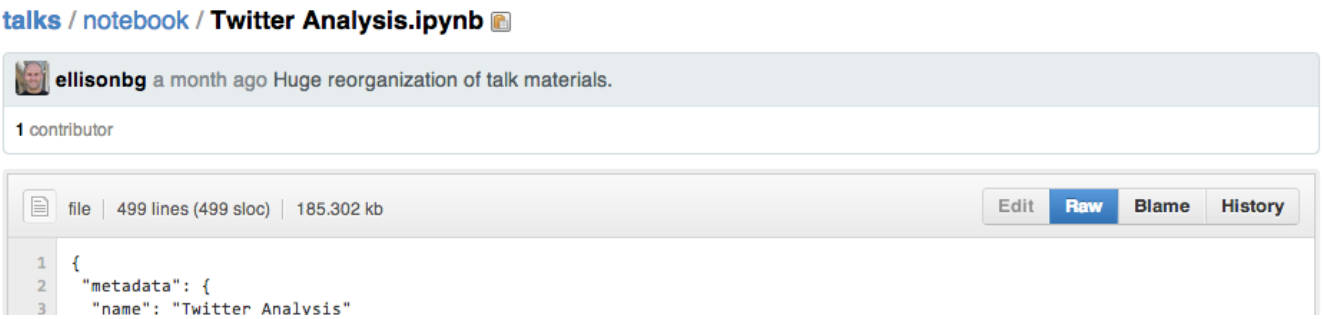
The File menu offers a "Print View." If you choose that, you can save your notebook as PDF to send to someone! You can also save your notebook as plain python code, for future use.

If you want to look at notebooks you find online, there is a useful online Notebook Viewer: http://nbviewer.ipython.org/

To use it, you must enter a path to a "raw" notebook; on github, this means clicking on the "raw" link before copying the path:

In [36]:
```
from IPython.core.display import Image
Image(filename='screencaps/raw_nb.png')
```

Out[36]:



After you go to the "raw" view and copy the url, you can load it in the NB viewer and get a url view like this:
http://nbviewer.ipython.org/urls/raw.github.com/ipython/talks/master/notebook/Twitter%2520Analysis.ipynb

Here's a few links to some more materials:

- A notebook tutorial deck: http://archive.ipython.org/media/PyCon2012-IPythonTutorial-Notebook.pdf
- The IPython website with presentations: http://ipython.org/presentation.html