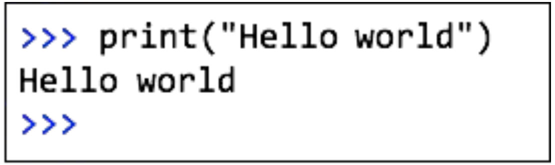
Python Environment To Works!

Alexis Salas Burgos

V 1.0

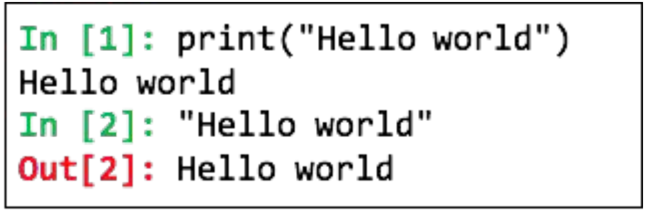
Python

- Interactive
  Default prompt >>>



```
>>> print("Hello world")
Hello world
>>>
```

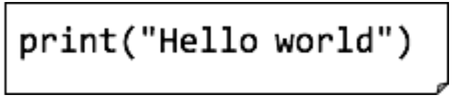
- IPython shell



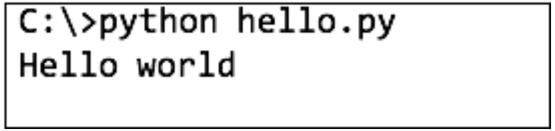
```
In [1]: print("Hello world")
Hello world
In [2]: "Hello world"
Out[2]: Hello world
```

- From a file

File hello.py

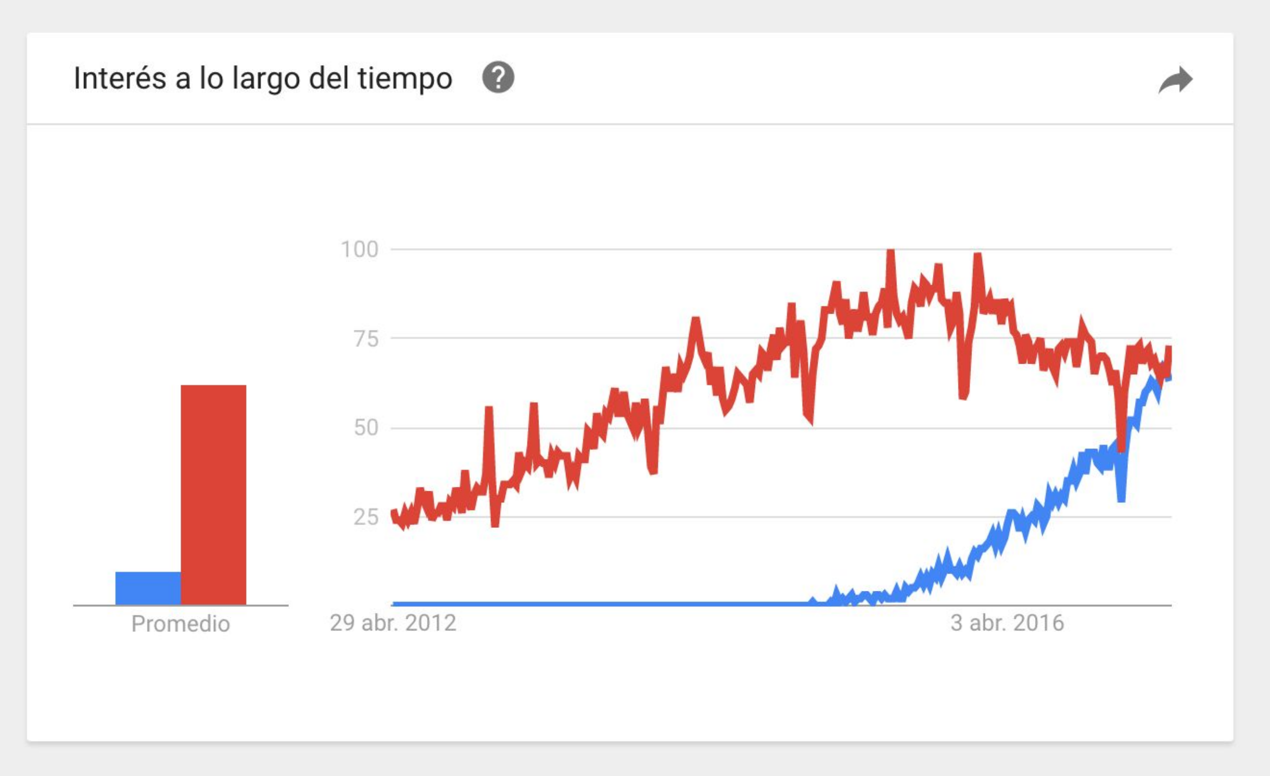


```
print("Hello world")
```



```
C:\>python hello.py
Hello world
```

Ipython (rojo) v/s Jupyter Notebook (Abril 2017)



Tendencias

Baja de uso

Matlab

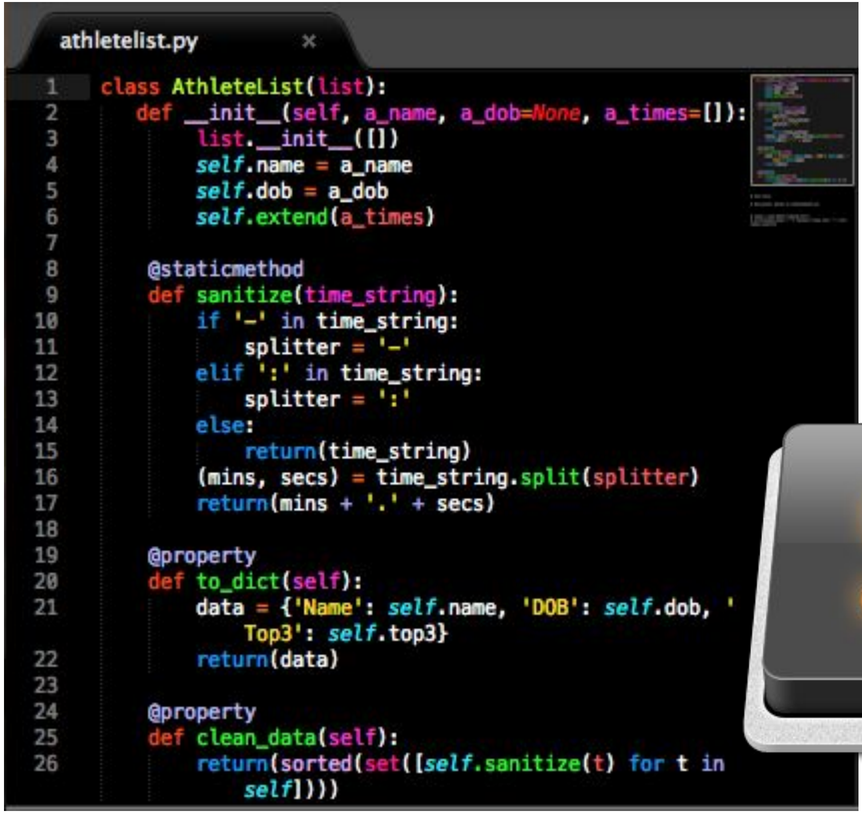
Tendencias

Python

R

Editores, Medioambientes e IDEs

- Editores
  - Sublime-text
  - Atom
- IDES
  - PyCharm
  - Spyder
  - Visual Code Studio
- Ipython Shell
- Jupyter



```python
class AthleteList(list):
    def __init__(self, a_name, a_dob=None, a_times=[]):
        list.__init__([])
        self.name = a_name
        self.dob = a_dob
        self.extend(a_times)

    @staticmethod
    def sanitize(time_string):
        if '-' in time_string:
            splitter = '-'
        elif ':' in time_string:
            splitter = ':'
        else:
            return(time_string)
        (mins, secs) = time_string.split(splitter)
        return(mins + '.' + secs)

    @property
    def to_dict(self):
        data = {'Name': self.name, 'DOB': self.dob, '
            Top3': self.top3}
        return(data)

    @property
    def clean_data(self):
        return(sorted(set([self.sanitize(t) for t in
            self])))
```

Medioambientes

Deprecated

Virtual env

Easy_install

Setup Tools

Otros lenguajes

Rvm https://rvm.io

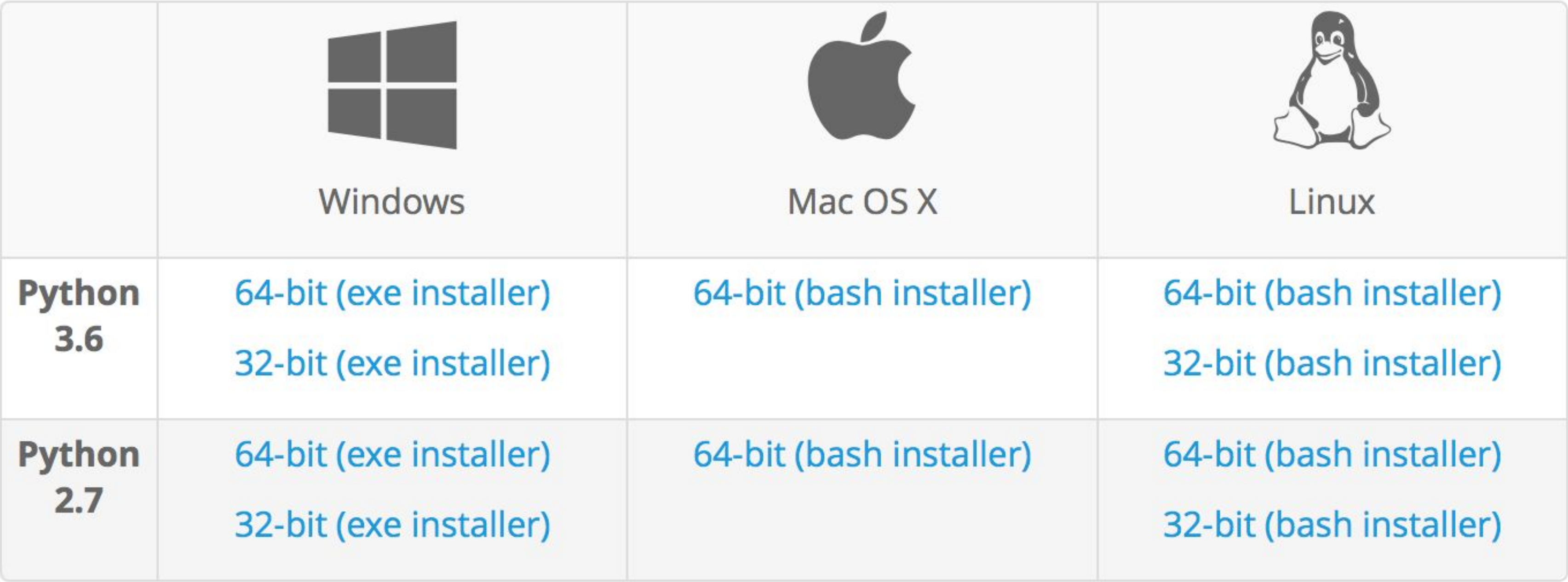
Jenv http://www.jenv.be

Python

https://www.continuum.io

https://pypi.python.org/pypi/pip

Miniconda



	Windows	Mac OS X	Linux
Python 3.6	64-bit (exe installer) 32-bit (exe installer)	64-bit (bash installer)	64-bit (bash installer) 32-bit (bash installer)
Python 2.7	64-bit (exe installer) 32-bit (exe installer)	64-bit (bash installer)	64-bit (bash installer) 32-bit (bash installer)

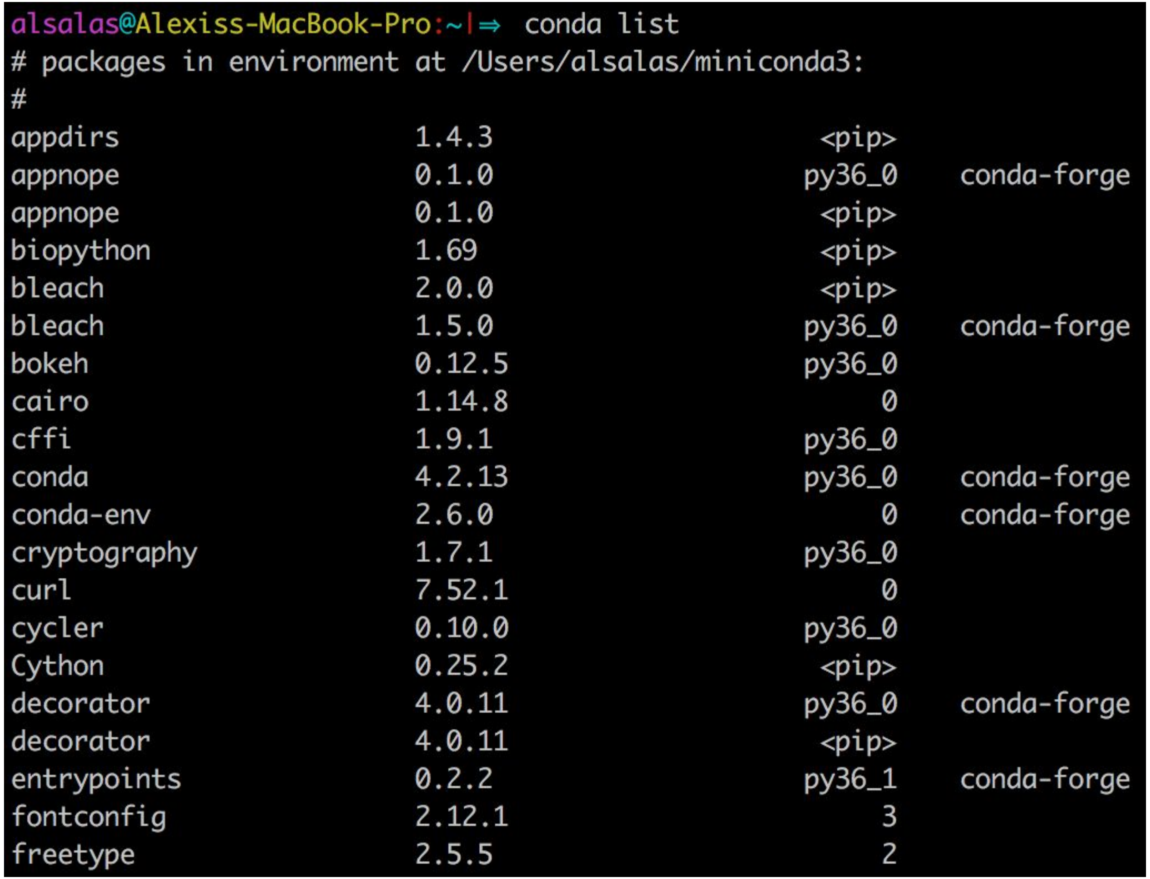
Instalado Jupyter

>conda install notebook

#Install ipython 6.0

#Install jupyter 5.0

#todas las dependencias

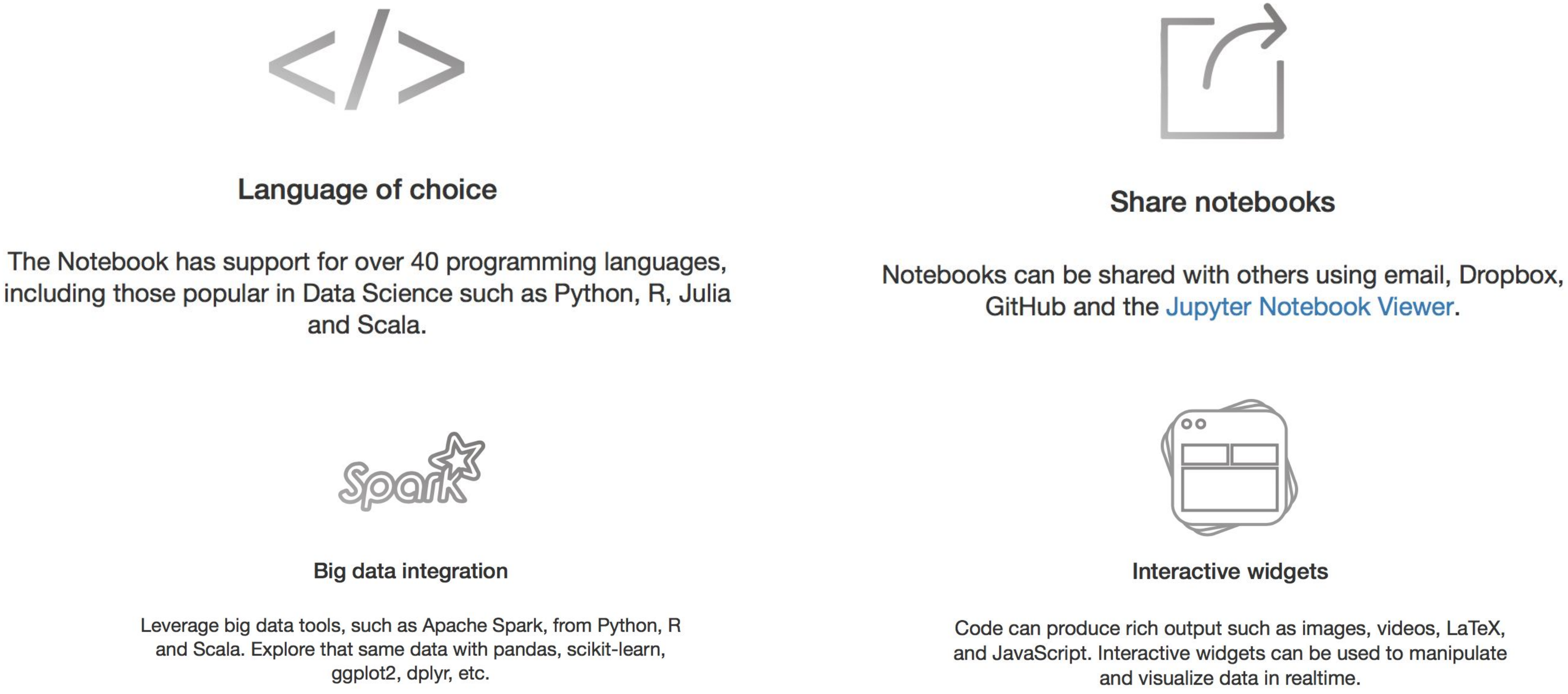


```
alsalas@Alexiss-MacBook-Pro:~|⇒  conda list
# packages in environment at /Users/alsalas/miniconda3:
#
appdirs                   1.4.3                     <pip>
appnope                   0.1.0                    py36_0    conda-forge
appnope                   0.1.0                     <pip>
biopython                 1.69                      <pip>
bleach                    2.0.0                     <pip>
bleach                    1.5.0                    py36_0    conda-forge
bokeh                     0.12.5                   py36_0
cairo                     1.14.8                        0
cffi                      1.9.1                    py36_0
conda                     4.2.13                   py36_0    conda-forge
conda-env                 2.6.0                         0    conda-forge
cryptography              1.7.1                    py36_0
curl                      7.52.1                        0
cycler                    0.10.0                   py36_0
Cython                    0.25.2                    <pip>
decorator                 4.0.11                   py36_0    conda-forge
decorator                 4.0.11                    <pip>
entrypoints               0.2.2                    py36_1    conda-forge
fontconfig                2.12.1                        3
freetype                  2.5.5                         2
```

Características de Jupyter

- Es una forma de combinar texto y fórmulas en conjunto con código (ejecutable) en un documento que es accesible desde un navegador.
- Los cuadernos son almacenados en un archivo de texto JSON con la extensión .ipynb
- Jupyter puede ejecutar sobre 40 lenguajes diferentes, originalmente fue concebido para Julia, Python y R (Ju-Pyt-eR).

Características de Jupyter



Language of choice

The Notebook has support for over 40 programming languages, including those popular in Data Science such as Python, R, Julia and Scala.

Share notebooks

Notebooks can be shared with others using email, Dropbox, GitHub and the Jupyter Notebook Viewer.

Big data integration

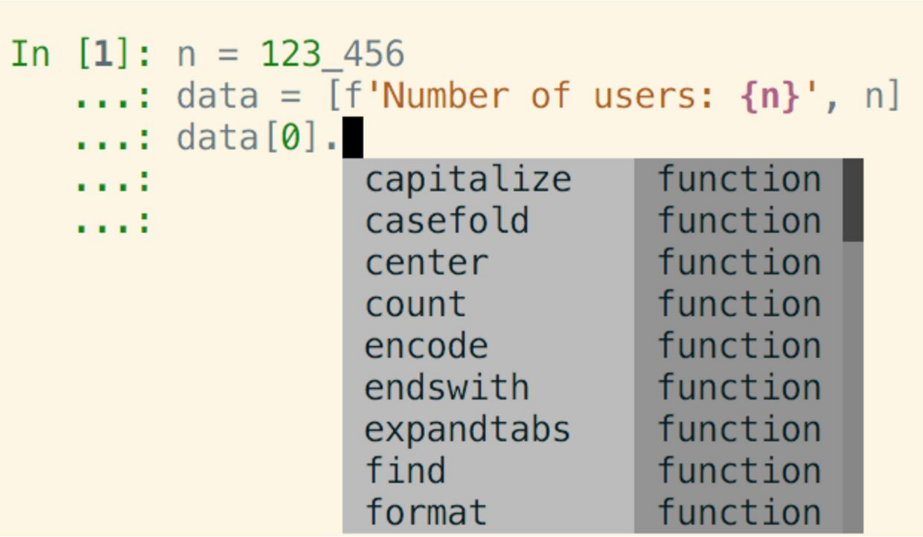
Leverage big data tools, such as Apache Spark, from Python, R and Scala. Explore that same data with pandas, scikit-learn, ggplot2, dplyr, etc.

Interactive widgets

Code can produce rich output such as images, videos, LaTeX, and JavaScript. Interactive widgets can be used to manipulate and visualize data in realtime.

Completado ipython 6 con Jedi

JEDI



```
In [1]: n = 123_456
   ...: data = [f'Number of users: {n}', n]
   ...: data[0].
   ...:         capitalize   function
   ...:         casefold     function
                center       function
                count        function
                encode       function
                endswith     function
                expandtabs   function
                find         function
                format       function
```

https://jedi.readthedocs.io/en/stable/

- Autocompletado con el tab
- Ayuda con el comando+?

Ventajas de Jupyter

Proveer tu código en un cuaderno favorece la reproducibilidad de los resultados.

Para responder todas las preguntas no es suficiente sólo el código.

Discute tus resultados obtenidos.

Comparte tus cuadernos en varios formatos.

Jupyter subcomandos

console

kernelspec

migrate

nbconvert

nbextension

notebook

qtconsole

run

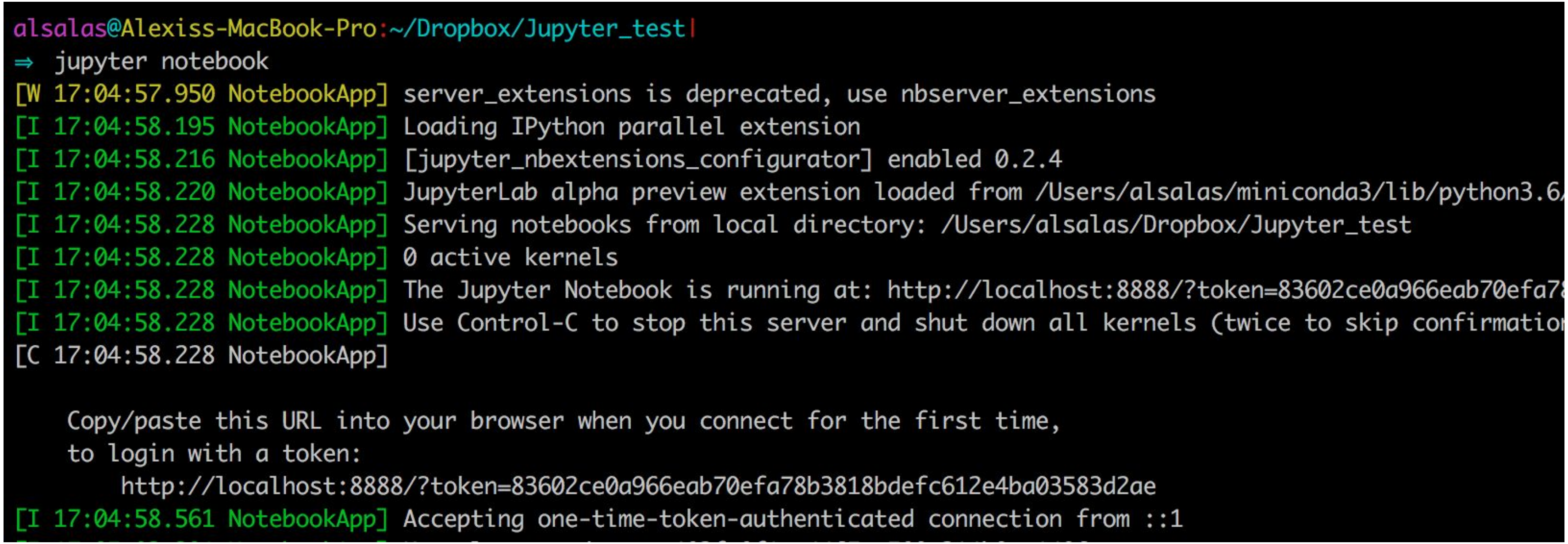
serverextension

troubleshoot

trust

Lanzamiento de un cuaderno

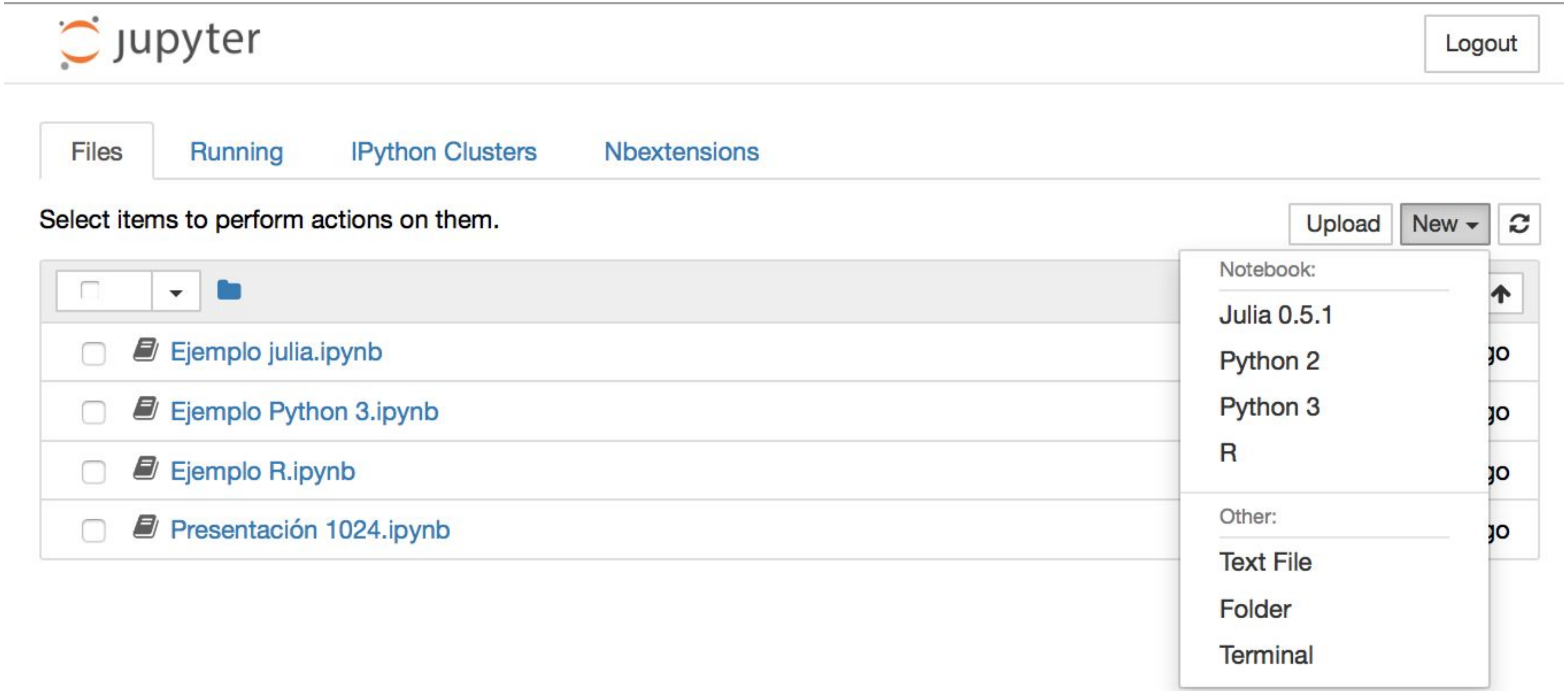
>jupyter notebook



```
alsalas@Alexiss-MacBook-Pro:~/Dropbox/Jupyter_test|
⇒  jupyter notebook
[W 17:04:57.950 NotebookApp] server_extensions is deprecated, use nbserver_extensions
[I 17:04:58.195 NotebookApp] Loading IPython parallel extension
[I 17:04:58.216 NotebookApp] [jupyter_nbextensions_configurator] enabled 0.2.4
[I 17:04:58.220 NotebookApp] JupyterLab alpha preview extension loaded from /Users/alsalas/miniconda3/lib/python3.6
[I 17:04:58.228 NotebookApp] Serving notebooks from local directory: /Users/alsalas/Dropbox/Jupyter_test
[I 17:04:58.228 NotebookApp] 0 active kernels
[I 17:04:58.228 NotebookApp] The Jupyter Notebook is running at: http://localhost:8888/?token=83602ce0a966eab70efa7
[I 17:04:58.228 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmatio
[C 17:04:58.228 NotebookApp]

    Copy/paste this URL into your browser when you connect for the first time,
    to login with a token:
        http://localhost:8888/?token=83602ce0a966eab70efa78b3818bdefc612e4ba03583d2ae
[I 17:04:58.561 NotebookApp] Accepting one-time-token-authenticated connection from ::1
```

Cuaderno en un navegador

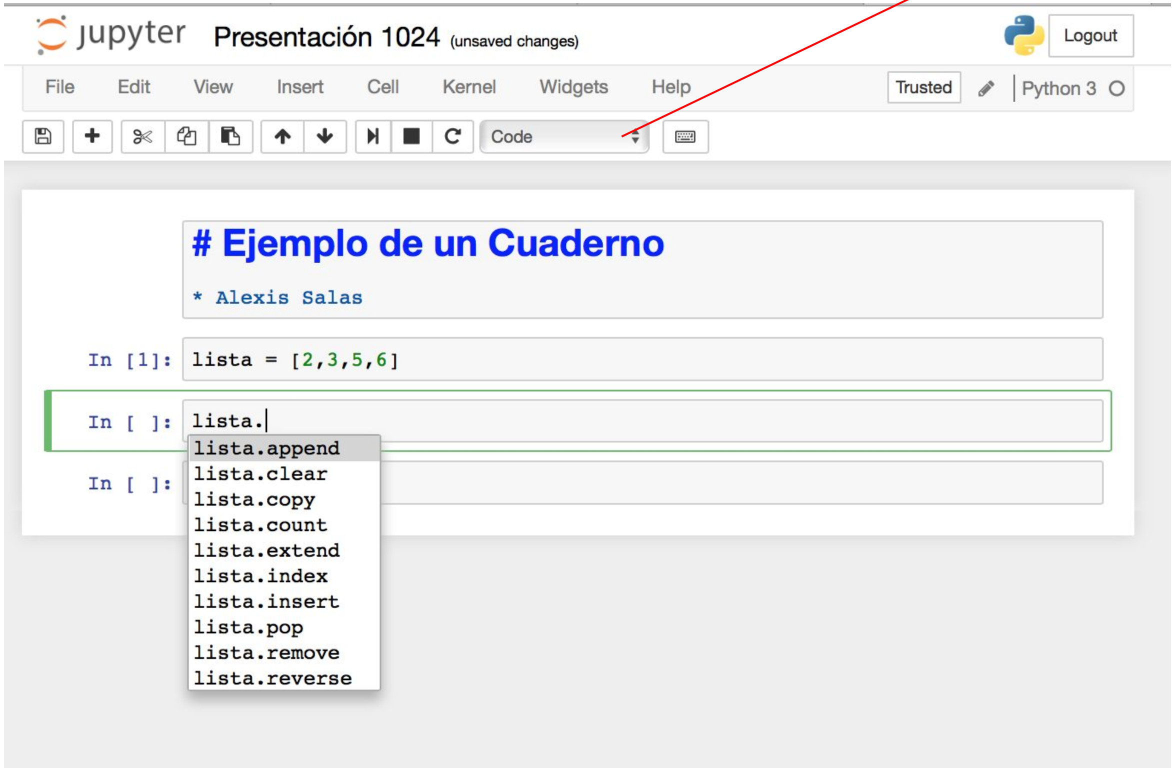
http://localhost:8888/tree#



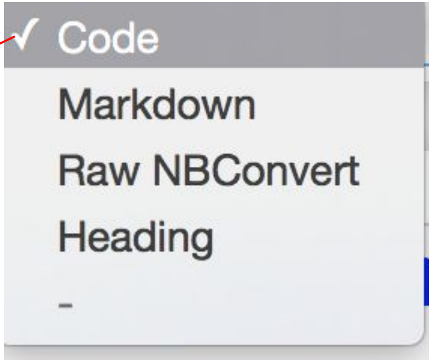
Select items to perform actions on them.

- Ejemplo julia.ipynb
- Ejemplo Python 3.ipynb
- Ejemplo R.ipynb
- Presentación 1024.ipynb

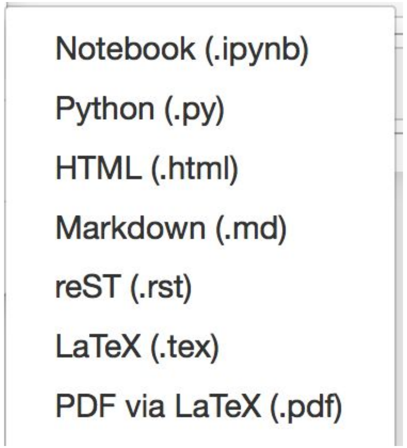
Cuaderno de Jupyter



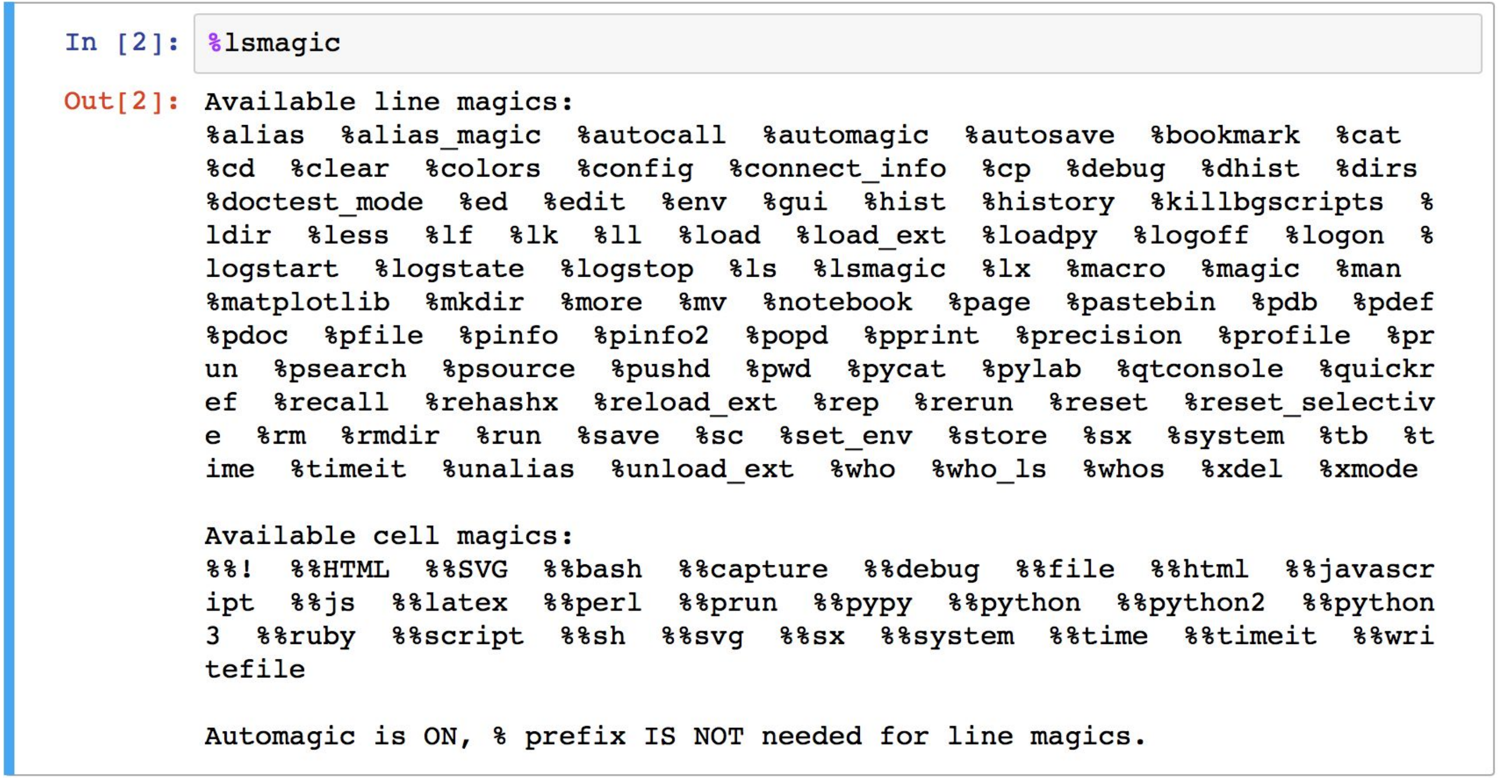
```
# Ejemplo de un Cuaderno

* Alexis Salas

In [1]: lista = [2,3,5,6]

In [ ]: lista.
        lista.append
        lista.clear
        lista.copy
        lista.count
        lista.extend
        lista.index
        lista.insert
        lista.pop
        lista.remove
        lista.reverse

In [ ]:
```



File>>Export>>

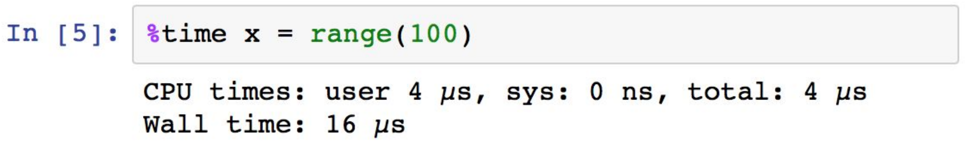


Los comandos mágicos %

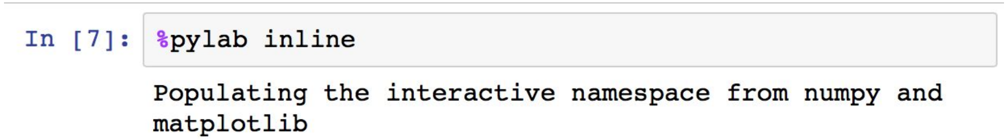


```
In [2]: %lsmagic

Out[2]: Available line magics:
        %alias  %alias_magic  %autocall  %automagic  %autosave  %bookmark  %cat
        %cd  %clear  %colors  %config  %connect_info  %cp  %debug  %dhist  %dirs
        %doctest_mode  %ed  %edit  %env  %gui  %hist  %history  %killbgscripts  %
        ldir  %less  %lf  %lk  %ll  %load  %load_ext  %loadpy  %logoff  %logon  %
        logstart  %logstate  %logstop  %ls  %lsmagic  %lx  %macro  %magic  %man
        %matplotlib  %mkdir  %more  %mv  %notebook  %page  %pastebin  %pdb  %pdef
        %pdoc  %pfile  %pinfo  %pinfo2  %popd  %pprint  %precision  %profile  %pr
        un  %psearch  %psource  %pushd  %pwd  %pycat  %pylab  %qtconsole  %quickr
        ef  %recall  %rehashx  %reload_ext  %rep  %rerun  %reset  %reset_selectiv
        e  %rm  %rmdir  %run  %save  %sc  %set_env  %store  %sx  %system  %tb  %t
        ime  %timeit  %unalias  %unload_ext  %who  %who_ls  %whos  %xdel  %xmode

        Available cell magics:
        %%!  %%HTML  %%SVG  %%bash  %%capture  %%debug  %%file  %%html  %%javascr
        ipt  %%js  %%latex  %%perl  %%prun  %%pypy  %%python  %%python2  %%python
        3  %%ruby  %%script  %%sh  %%svg  %%sx  %%system  %%time  %%timeit  %%wri
        tefile

        Automagic is ON, % prefix IS NOT needed for line magics.
```

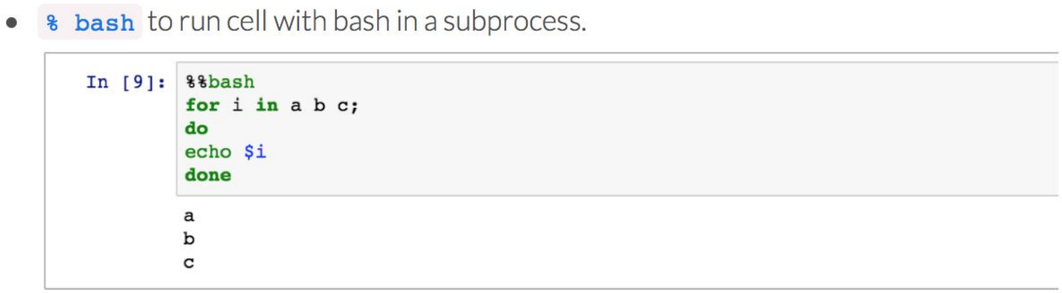
Los comandos mágicos %



```
In [5]: %time x = range(100)

CPU times: user 4 µs, sys: 0 ns, total: 4 µs
Wall time: 16 µs
```

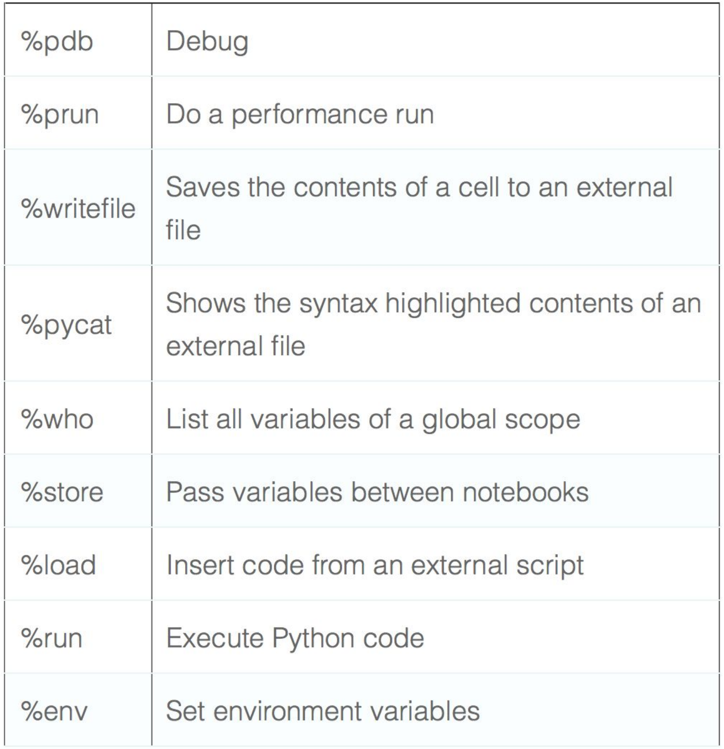


```
In [7]: %pylab inline

Populating the interactive namespace from numpy and
matplotlib
```



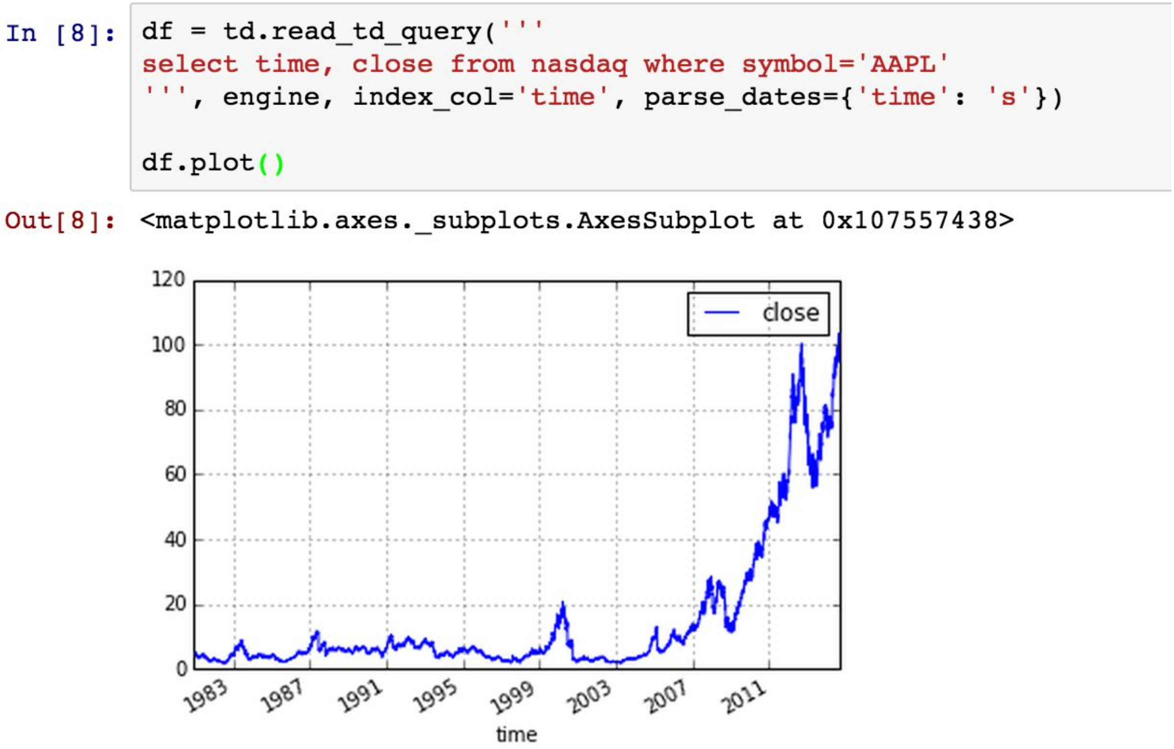
- `% bash` to run cell with bash in a subprocess.

```
In [9]: %%bash
        for i in a b c;
        do
        echo $i
        done

        a
        b
        c
```



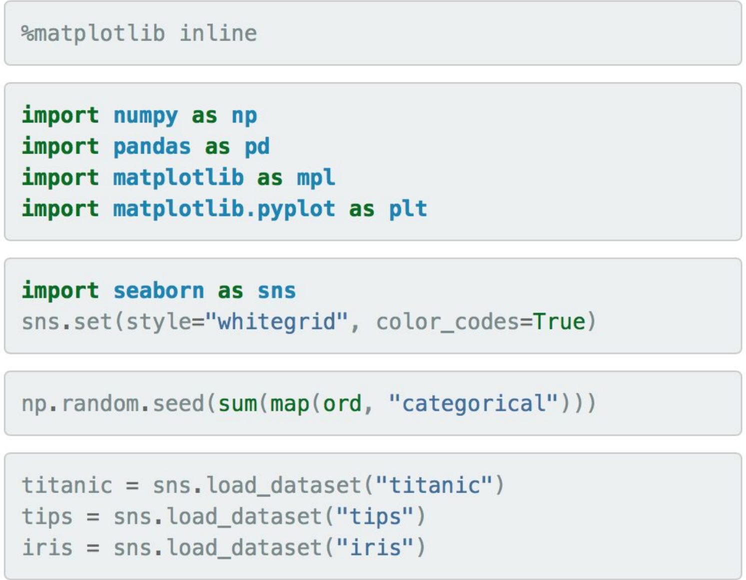
%pdb	Debug
%prun	Do a performance run
%writefile	Saves the contents of a cell to an external file
%pycat	Shows the syntax highlighted contents of an external file
%who	List all variables of a global scope
%store	Pass variables between notebooks
%load	Insert code from an external script
%run	Execute Python code
%env	Set environment variables

Matplotlib http://matplotlib.org



```
In [8]: df = td.read_td_query('''
        select time, close from nasdaq where symbol='AAPL'
        ''', engine, index_col='time', parse_dates={'time': 's'})

        df.plot()

Out[8]: <matplotlib.axes._subplots.AxesSubplot at 0x107557438>
```

Seaborn http://seaborn.pydata.org



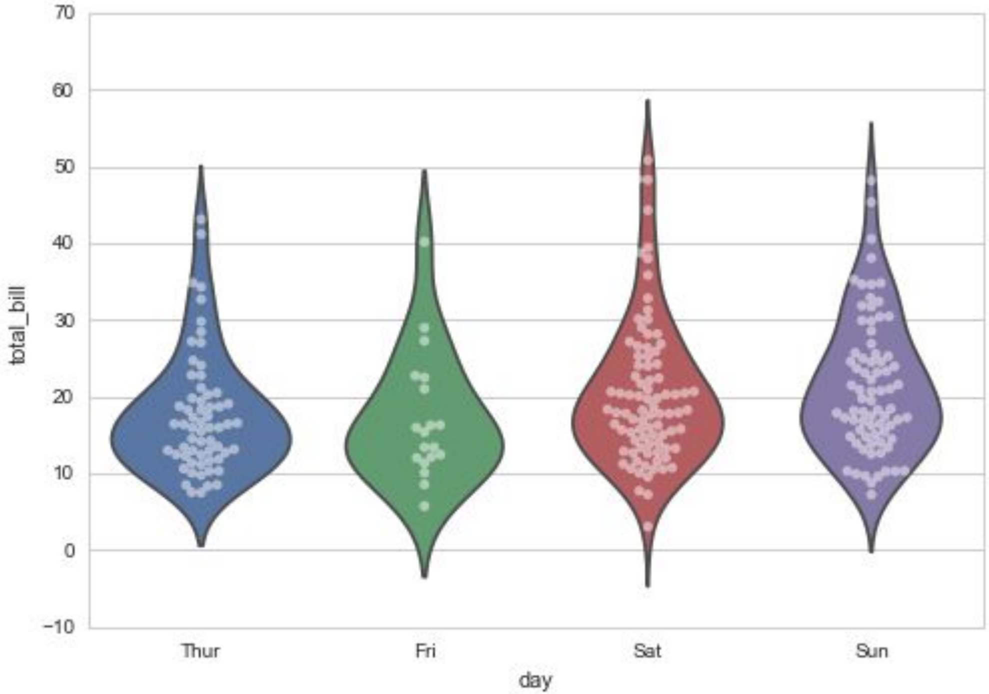
```
%matplotlib inline
```

```
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
```

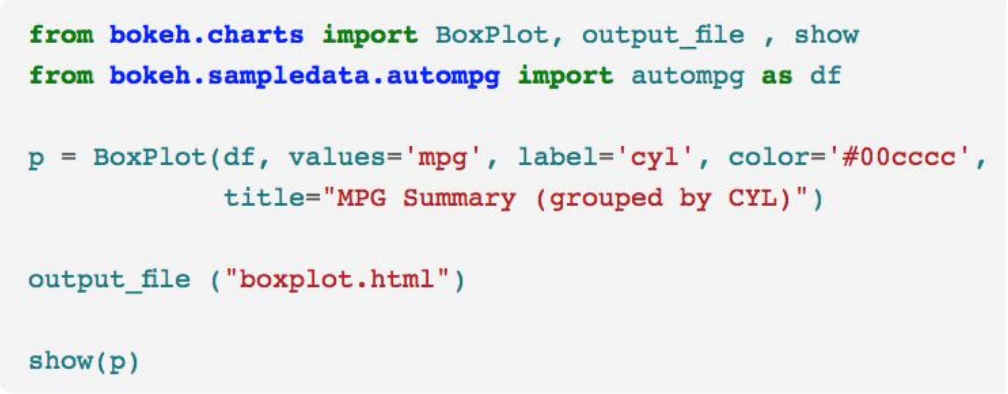
```
import seaborn as sns
sns.set(style="whitegrid", color_codes=True)
```

```
np.random.seed(sum(map(ord, "categorical")))
```

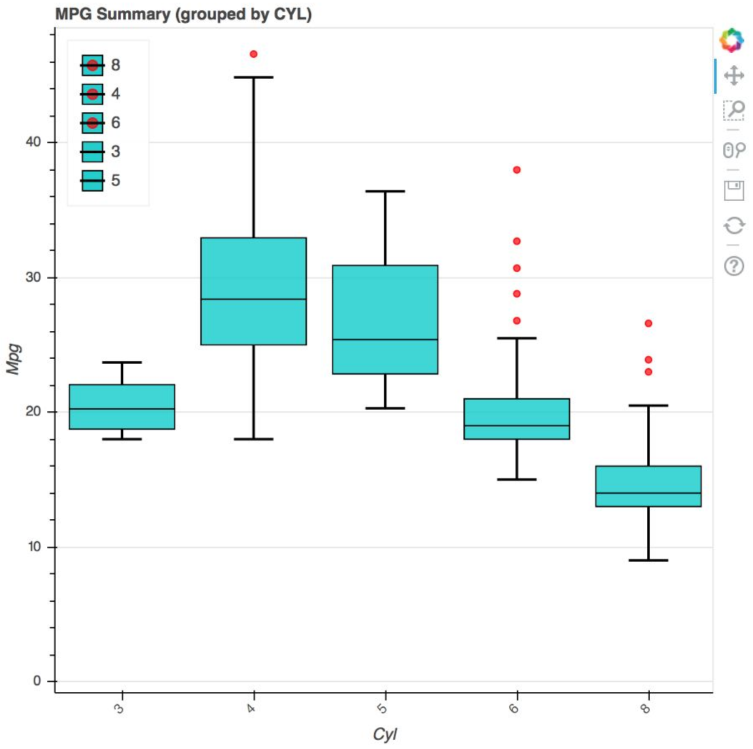
```
titanic = sns.load_dataset("titanic")
tips = sns.load_dataset("tips")
iris = sns.load_dataset("iris")
```



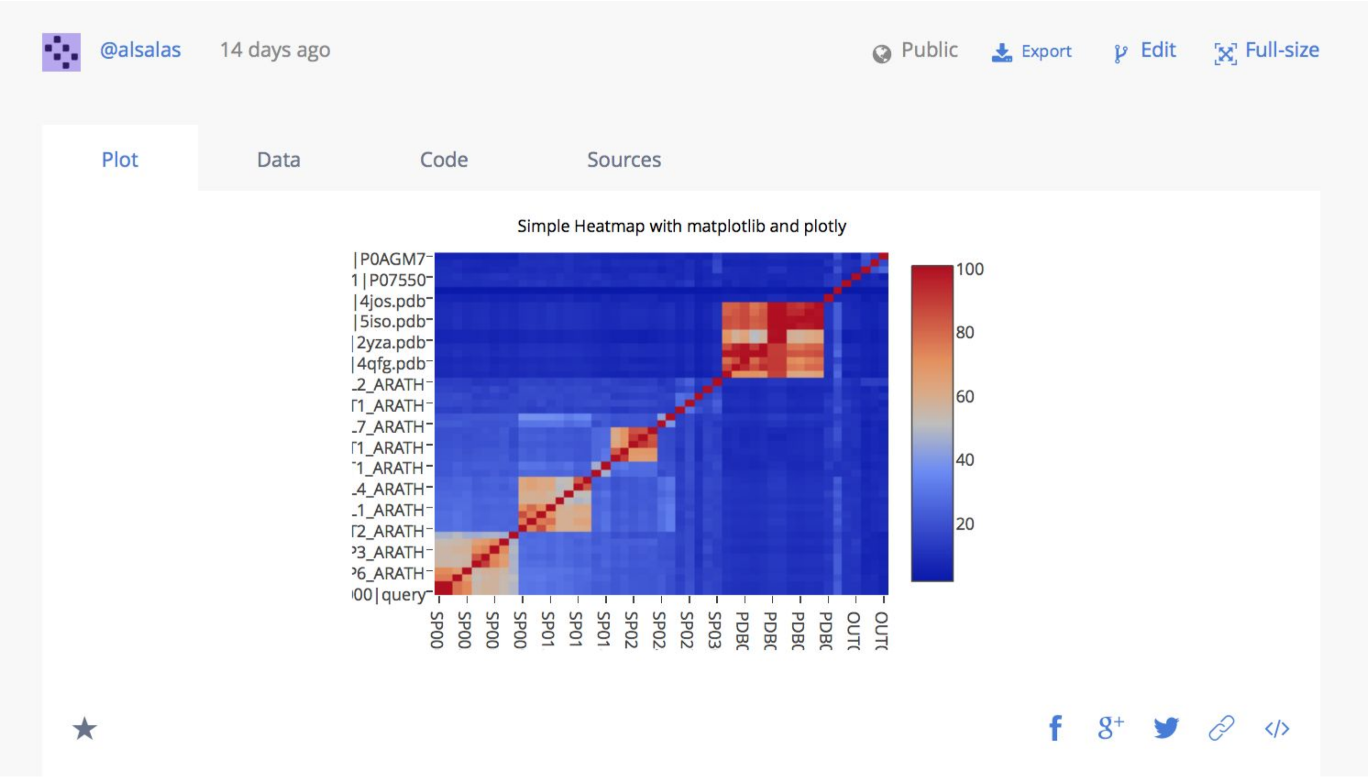
Bokeh http://bokeh.pydata.org



```
from bokeh.charts import BoxPlot, output_file , show
from bokeh.sampledata.autompg import autompg as df

p = BoxPlot(df, values='mpg', label='cyl', color='#00cccc',
            title="MPG Summary (grouped by CYL)")

output_file ("boxplot.html")

show(p)
```



Plot.ly https://plot.ly



Jupyter Widgets

Notebook Widgets

http://jupyter.org/widgets

Instalación:

```
pip install ipywidgets
jupyter nbextension enable --py
--sys-prefix widgetsnbextension
```

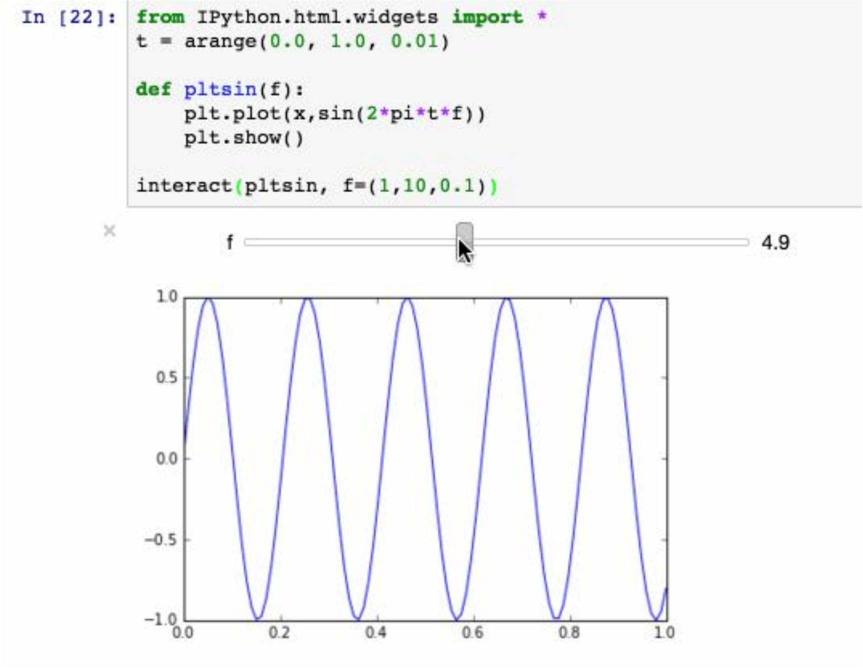
```
pip install ipyleaflet
jupyter nbextension enable --py --sys-prefix ipyleaflet
```

```
pip install bqplot
jupyter nbextension enable --py --sys-prefix bqplot
```

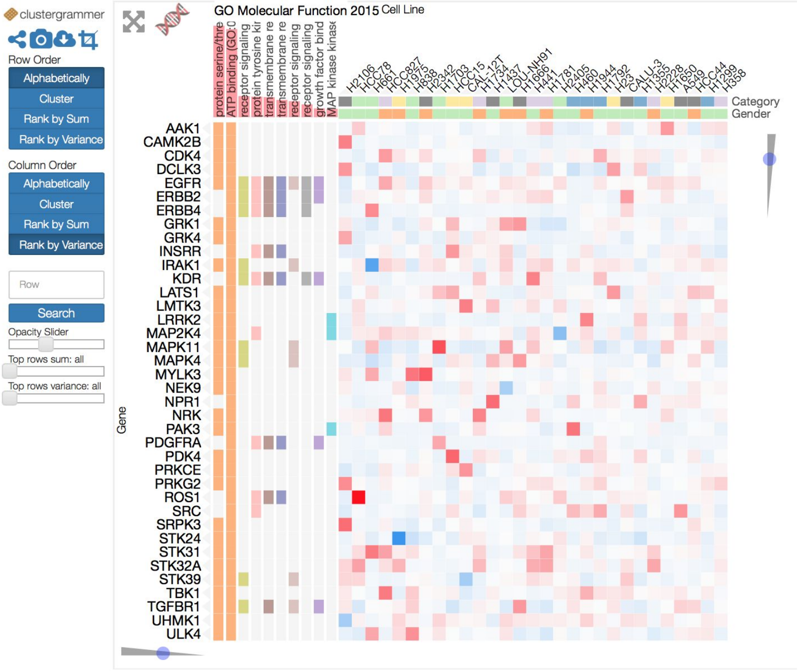
```
pip install pythreejs
jupyter nbextension enable --py --sys-prefix pythreejs
```

```
pip install ipyvolume
jupyter nbextension enable --py --sys-prefix ipyvolume
```

Jupyter Widgets

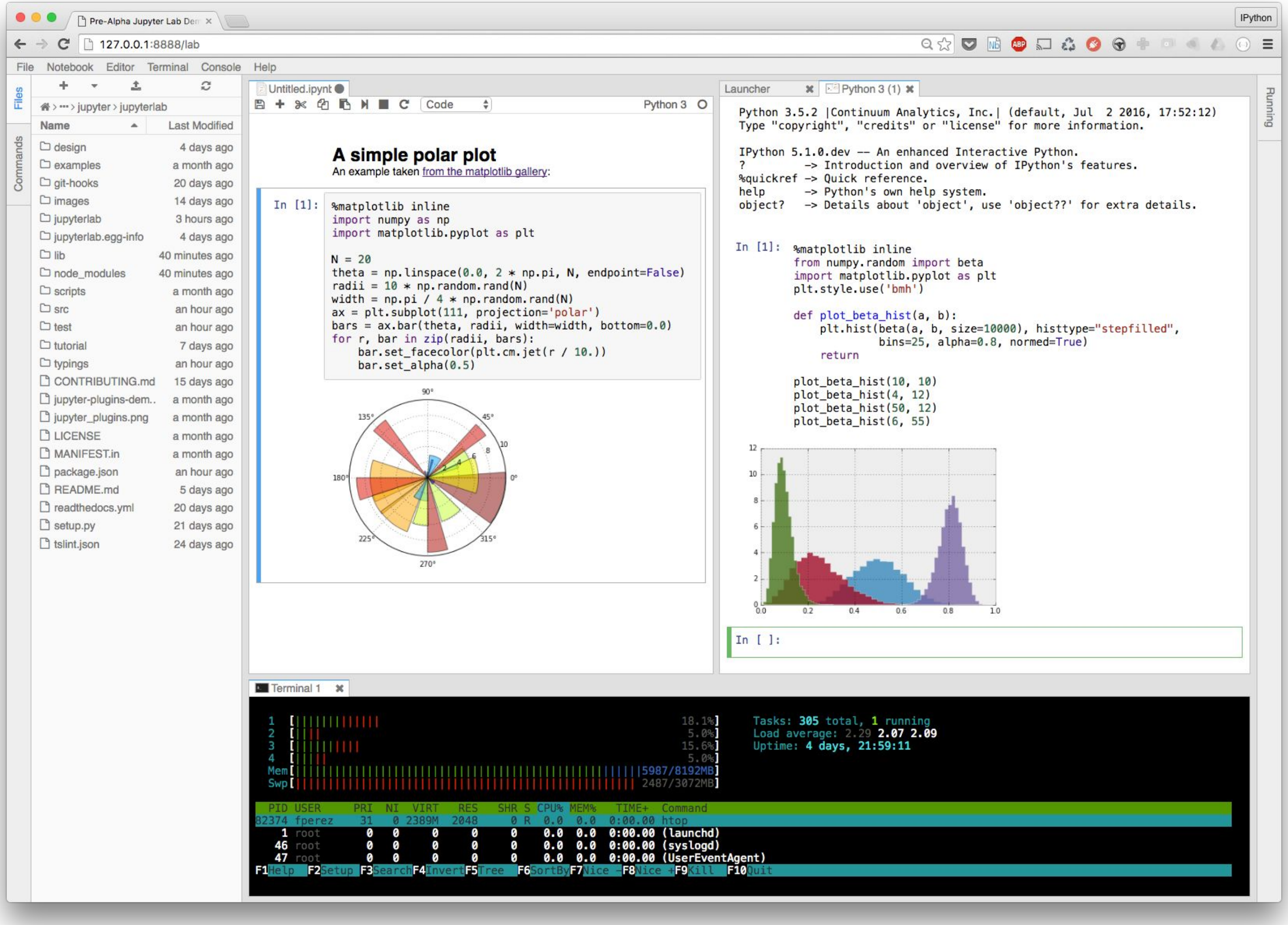


```
In [22]: from IPython.html.widgets import *
         t = arange(0.0, 1.0, 0.01)

         def pltsin(f):
             plt.plot(x,sin(2*pi*t*f))
             plt.show()

         interact(pltsin, f=(1,10,0.1))
```



http://nbviewer.jupyter.org/github/MaayanLab/clustergrammer-widget/blob/master/Running_clustergrammer_widget.ipynb

JupyterLab: La nueva generación de Jupyter notebook



Name	Last Modified
design	4 days ago
examples	a month ago
git-hooks	20 days ago
images	14 days ago
jupyterlab	3 hours ago
jupyterlab.egg-info	4 days ago
lib	40 minutes ago
node_modules	40 minutes ago
scripts	a month ago
src	an hour ago
test	an hour ago
tutorial	7 days ago
typings	an hour ago
CONTRIBUTING.md	15 days ago
jupyter-plugins-dem..	a month ago
jupyter_plugins.png	a month ago
LICENSE	a month ago
MANIFEST.in	a month ago
package.json	an hour ago
README.md	5 days ago
readthedocs.yml	20 days ago
setup.py	21 days ago
tslint.json	24 days ago

A simple polar plot

An example taken from the matplotlib gallery:

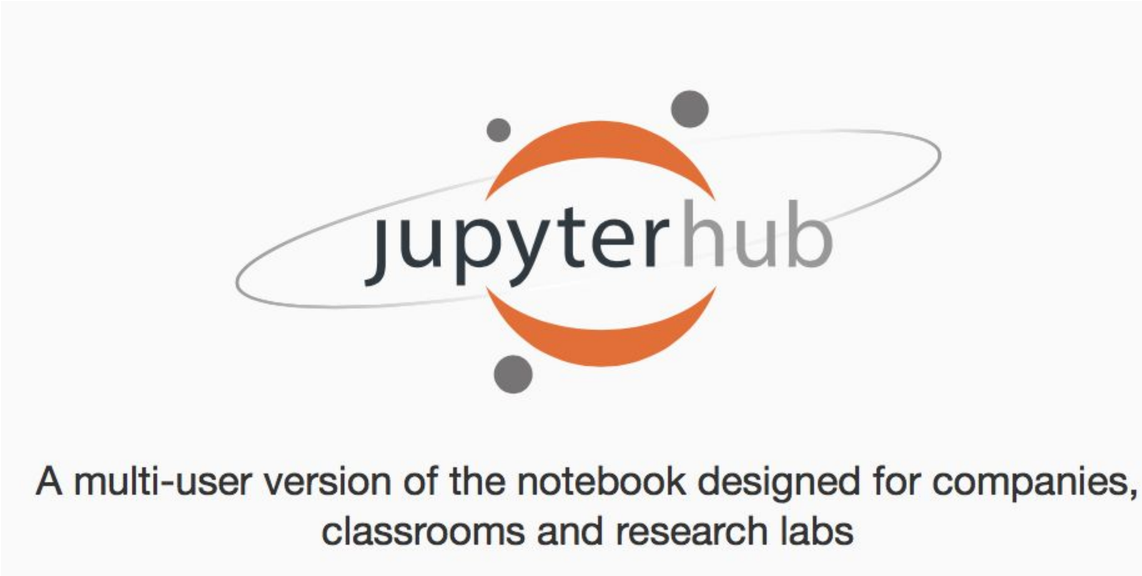
```
In [1]: %matplotlib inline
        import numpy as np
        import matplotlib.pyplot as plt

        N = 20
        theta = np.linspace(0.0, 2 * np.pi, N, endpoint=False)
        radii = 10 * np.random.rand(N)
        width = np.pi / 4 * np.random.rand(N)
        ax = plt.subplot(111, projection='polar')
        bars = ax.bar(theta, radii, width=width, bottom=0.0)
        for r, bar in zip(radii, bars):
            bar.set_facecolor(plt.cm.jet(r / 10.))
            bar.set_alpha(0.5)
```

```
Python 3.5.2 |Continuum Analytics, Inc.| (default, Jul  2 2016, 17:52:12)
Type "copyright", "credits" or "license" for more information.

IPython 5.1.0.dev -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.


In [1]: %matplotlib inline
        from numpy.random import beta
        import matplotlib.pyplot as plt
        plt.style.use('bmh')

        def plot_beta_hist(a, b):
            plt.hist(beta(a, b, size=10000), histtype="stepfilled",
                     bins=25, alpha=0.8, normed=True)
            return

        plot_beta_hist(10, 10)
        plot_beta_hist(4, 12)
        plot_beta_hist(50, 12)
        plot_beta_hist(6, 55)
```

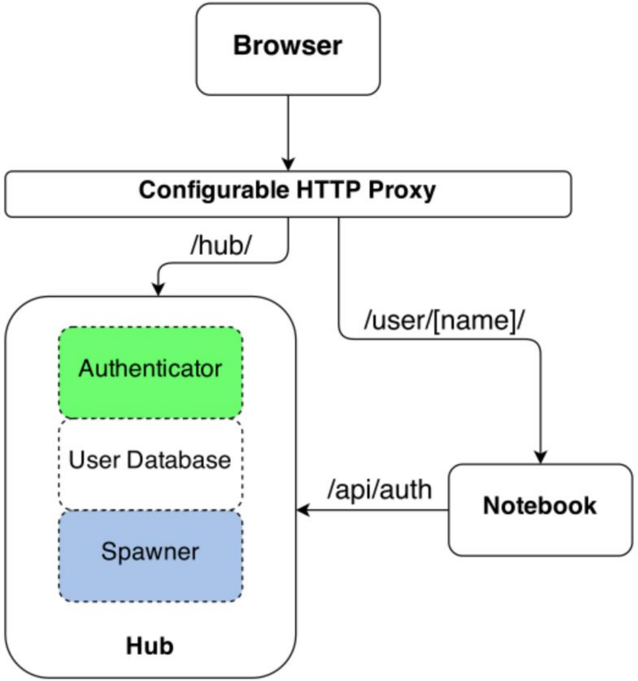
```
In [ ]:
```

```
  1  [|||||||||||||                                                18.1%]   Tasks: 305 total, 1 running
  2  [||||                                                          5.0%]   Load average: 2.29 2.07 2.09
  3  [||||||||||                                                   15.6%]   Uptime: 4 days, 21:59:11
  4  [|||||                                                         5.0%]
  Mem[||||||||||||||||||||||||||||||||||||||||||||||||||||5987/8192MB]
  Swp[|||||||||||||||||||||||||||||||||||||||||||||||||   2487/3072MB]

  PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
82374 fperez     31   0 2389M  2048     0 R  0.0  0.0  0:00.00 htop
    1 root        0   0     0     0     0    0.0  0.0  0:00.00 (launchd)
   46 root        0   0     0     0     0    0.0  0.0  0:00.00 (syslogd)
   47 root        0   0     0     0     0    0.0  0.0  0:00.00 (UserEventAgent)
F1Help  F2Setup  F3SearchF4InvertF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```

https://github.com/jupyterlab/jupyterlab



A multi-user version of the notebook designed for companies, classrooms and research labs



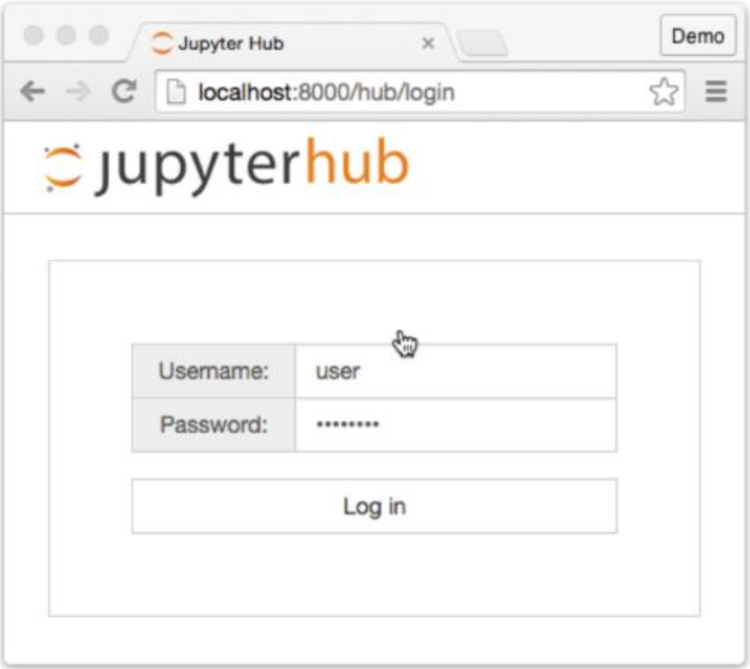
```
docker run -d --name jupyterhub jupyterhub/jupyterhub jupyterhub
```

https://github.com/jupyterhub/jupyterhub

jupyterhub

- El ingreso de usuarios es administrado por el Hub
- Los usuarios se autentifican vía un formulario
- Hub soporta proxies
- Redirige al usuario a /user/[name]
- Configuración centralizada
- Seguimiento de cuadernos

r



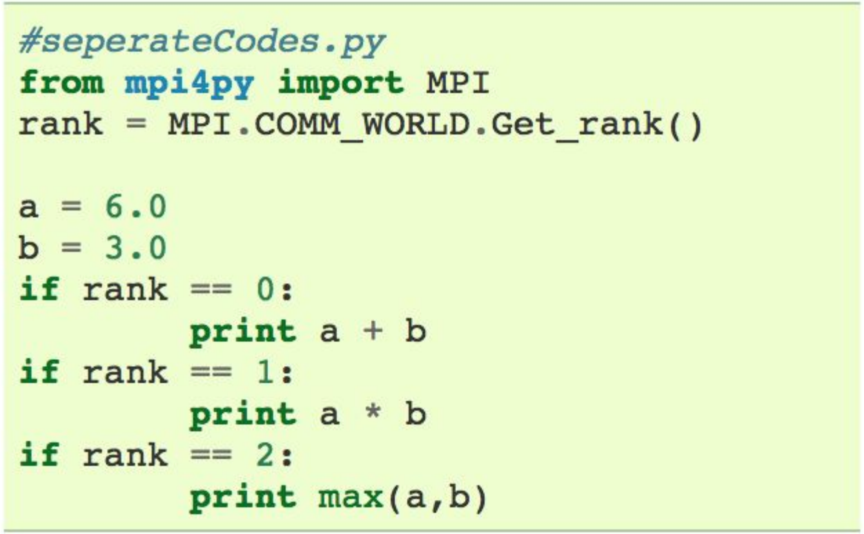
Username: user

Password: •••••••••

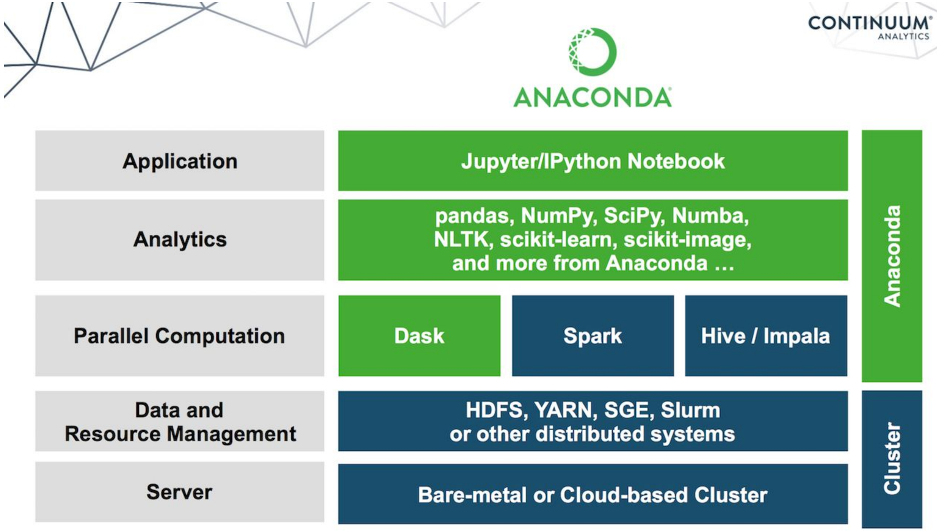
Jupyter y Paralelismo

Mpi4py http://pythonhosted.org/mpi4py

Dask https://github.com/dask

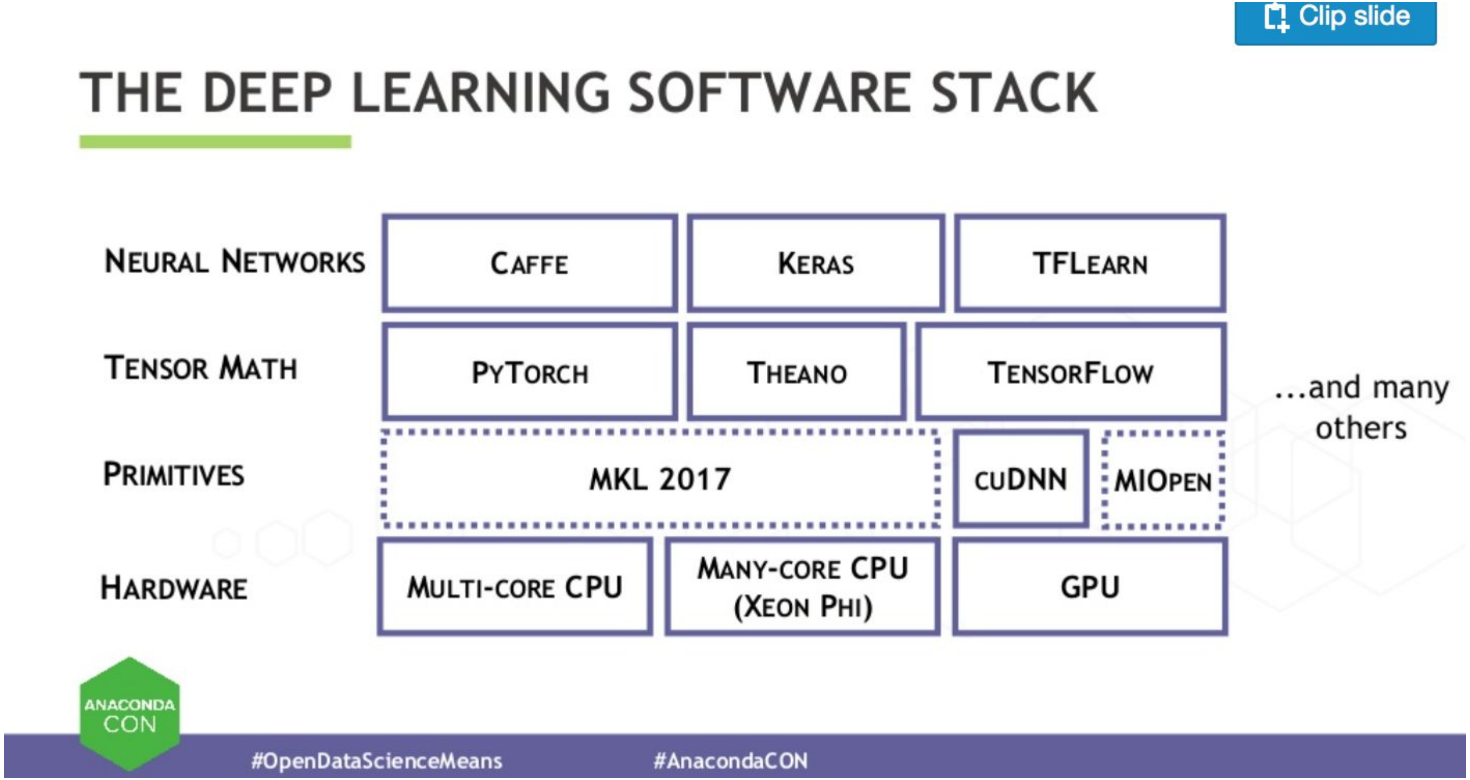


```
#seperateCodes.py
from mpi4py import MPI
rank = MPI.COMM_WORLD.Get_rank()

a = 6.0
b = 3.0
if rank == 0:
        print a + b
if rank == 1:
        print a * b
if rank == 2:
        print max(a,b)
```



CONTINUUM ANALYTICS
ANACONDA
Application
Jupyter/IPython Notebook
Analytics
pandas, NumPy, SciPy, Numba, NLTK, scikit-learn, scikit-image, and more from Anaconda …
Parallel Computation
Dask
Spark
Hive / Impala
Data and Resource Management
HDFS, YARN, SGE, Slurm or other distributed systems
Server
Bare-metal or Cloud-based Cluster
Anaconda
Cluster

Jupyter y Aprendizaje Profundo IA



THE DEEP LEARNING SOFTWARE STACK

NEURAL NETWORKS	CAFFE	KERAS	TFLEARN
TENSOR MATH	PYTORCH	THEANO	TENSORFLOW
PRIMITIVES	MKL 2017	CUDNN	MIOPEN
HARDWARE	MULTI-CORE CPU	MANY-CORE CPU (XEON PHI)	GPU

...and many others

#OpenDataScienceMeans #AnacondaCON

Alexis Salas Burgos
alsalas@udec.cl

V 1.0

¿Consultas?