

Jupyter & it's Moons

**Shreyas Cholia, Matt Henderson, Oliver Evans
(and in spirit, Fernando Pérez)**

Lawrence Berkeley Lab

Gateways 2017 Tutorial, Ann Arbor MI

Getting Started

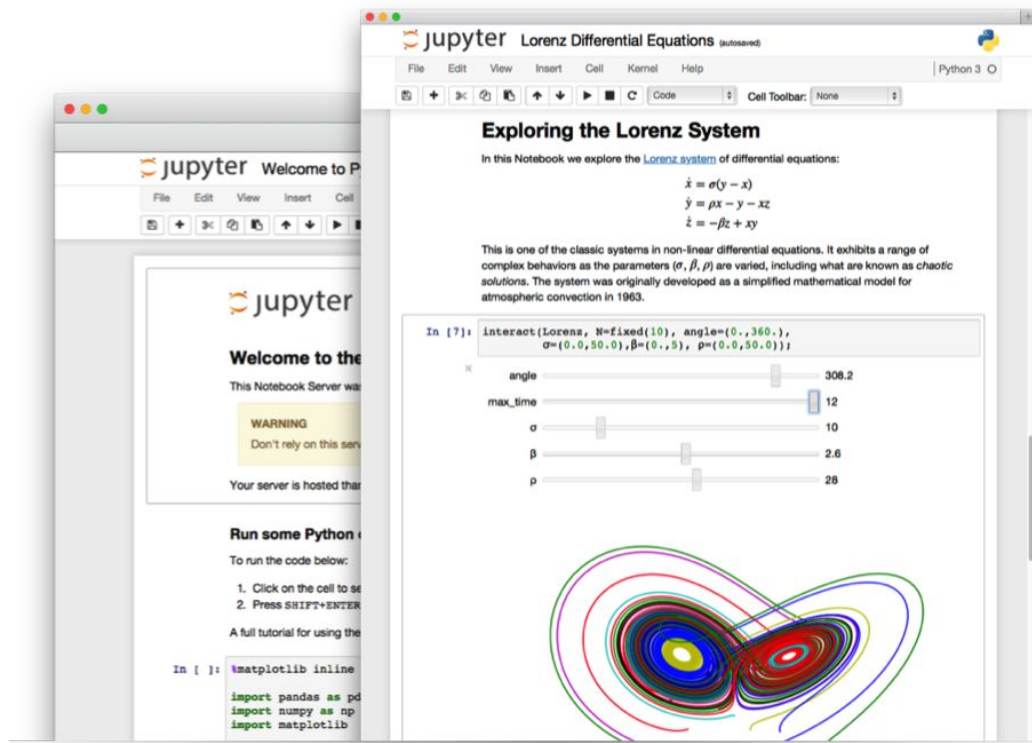
<http://tinyurl.com/jupyter-moons>

<https://github.com/Jupyter-Kale/Jupyter-and-its-moons>

Make sure Anaconda is installed and running

Jupyter

Interactive notebook environment that can combine code with visualizations, documentation and programmable widgets



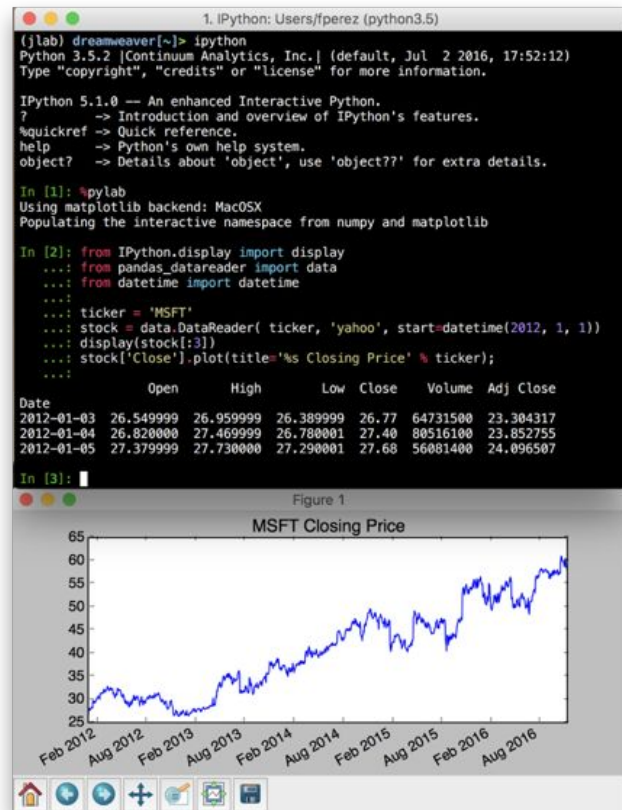
Aside

A brief history of scientific computing

- Brought to you by Jupyter

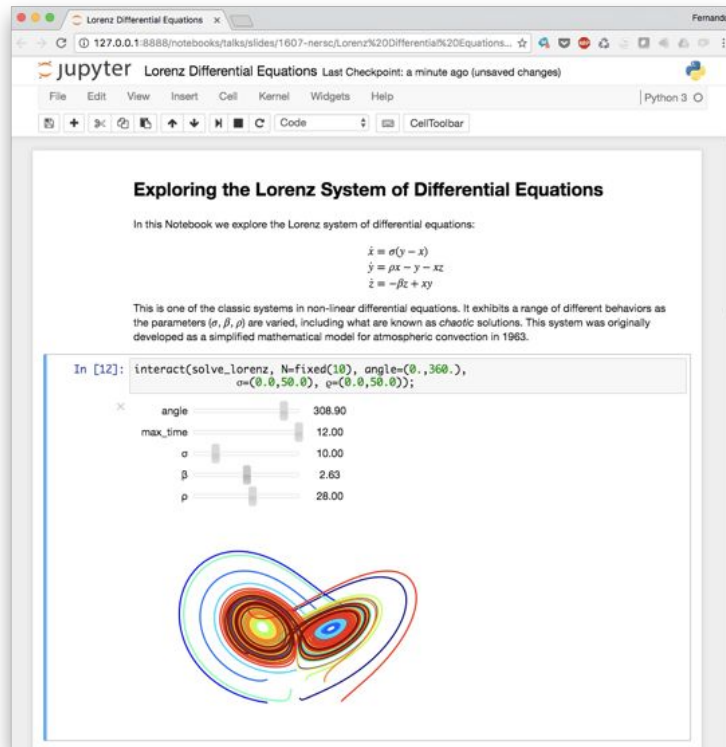
IPython: Interactive Python, 2001

- Object Introspection (TAB!)
- OS Integration
- Rich terminal client
- GUI support (plots, ...)
- %magic commands
- Embeddable



The IPython/Jupyter Notebook

- Rich web client
- Text & math
- Code
- Results
- Share, reproduce.



Why Jupyter?

- Interactivity
- Reproducibility and Collaboration
- Education

Interactive Computing

- Code + Viz + Widgets
- Paradigm shift for exploratory computing
- Break up steps into cells

5 lines (214 sloc) | 4.72 KB

Raw

Blame

History



Factoring Polynomials with SymPy

Here is an example that uses [SymPy](#) to factor polynomials.

```
In [1]: from ipywidgets import interact
```

```
In [2]: from sympy import Symbol, Eq, factor
```

```
In [3]: x = Symbol('x')
```

```
In [4]: def factorit(n):  
        return Eq(x**n-1, factor(x**n-1))
```

```
In [5]: factorit(12)
```

```
Out[5]: Eq(x**12 - 1, (x - 1)*(x + 1)*(x**2 + 1)*(x**2 - x + 1)*(x**2 + x + 1)*(x**4 - x**2 + 1))
```

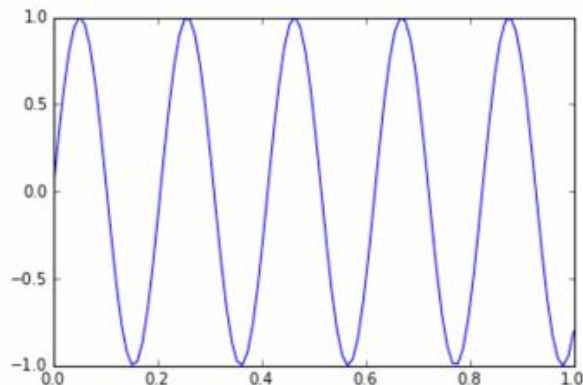
```
In [6]: interact(factorit, n=(2,40));
```

A Jupyter Widget

```
In [22]: from IPython.html.widgets import *  
t = arange(0.0, 1.0, 0.01)  
  
def plttsin(f):  
    plt.plot(x, sin(2*pi*t*f))  
    plt.show()  
  
interact(plttsin, f=(1,10,0.1))
```



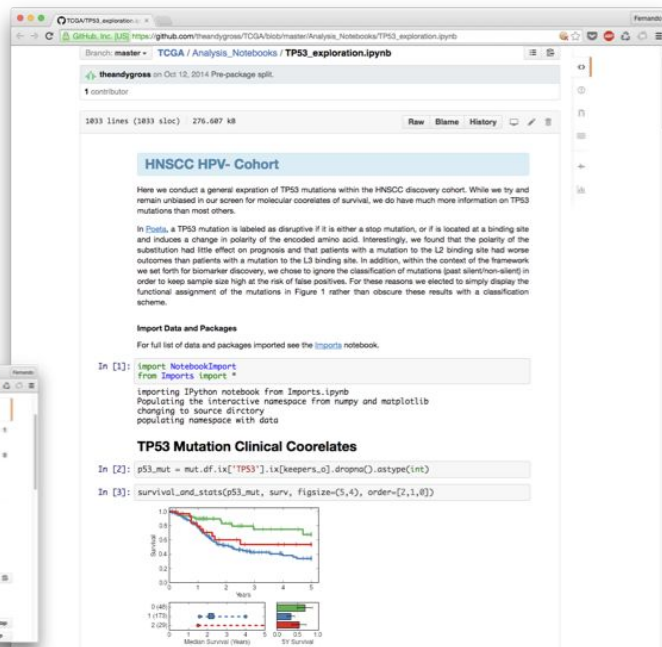
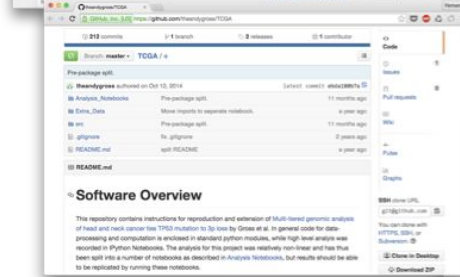
f



Reproducible Research: Papers + Notebooks on Github

“An article about computational science in a scientific publication is not the scholarship itself, it is merely advertising of the scholarship. The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures.”

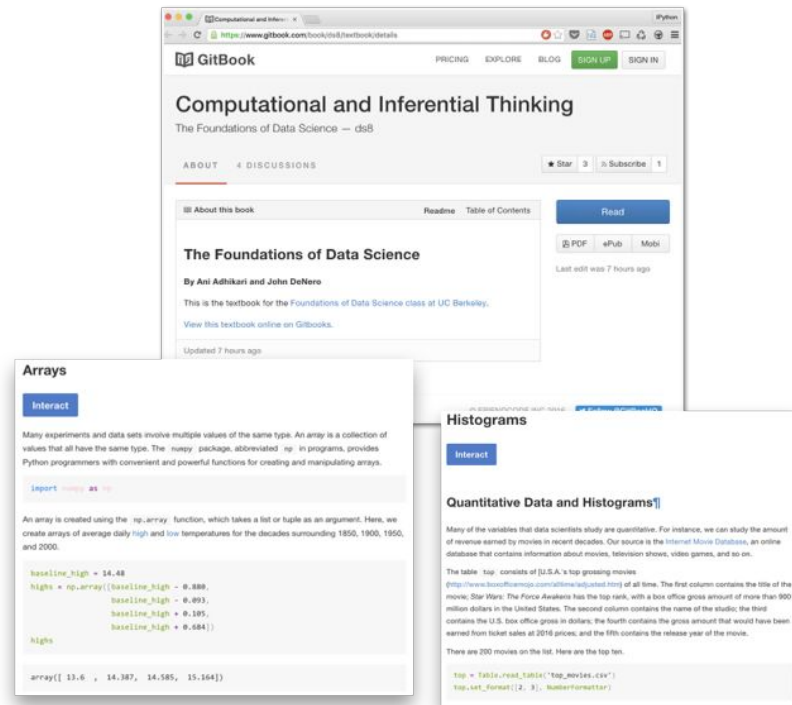
Buckheit and Donoho, WaveLab and Reproducible Research, 1995



Education

- New curriculum aimed at all freshmen at UC Berkeley
- Interactive textbook is Jupyter Notebooks
- Course deployment is JupyterHub

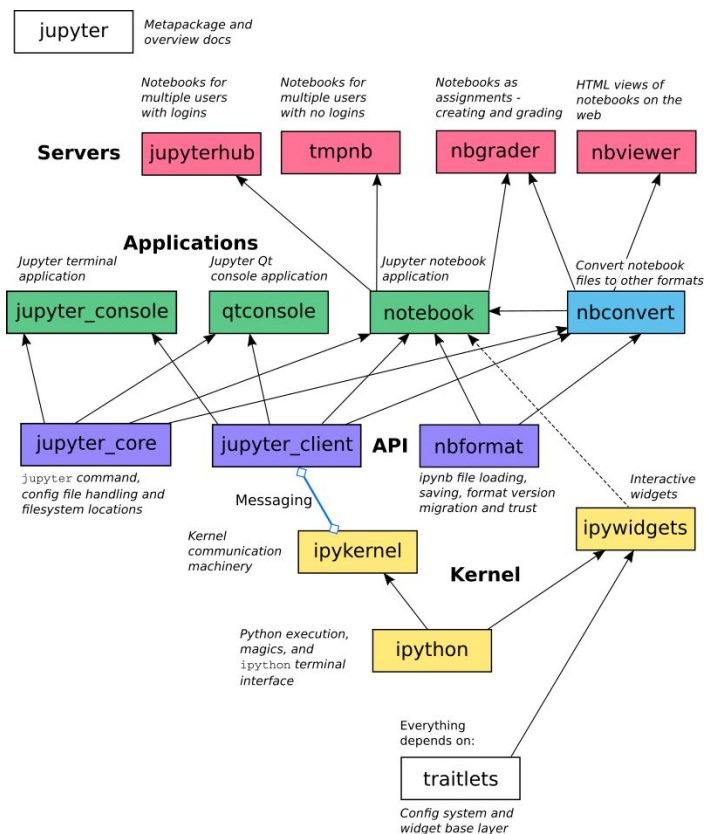
<http://data8.org>



Lots of things have emerged in the ecosystem

- Multi-user environments (Jupyterhub)
- Interactive Widgets
- Jupyter Platform (Jupyterlab)
- Sharing
- Classroom platform (nbgrader)
- And more ...

Visual overview of the ecosystem

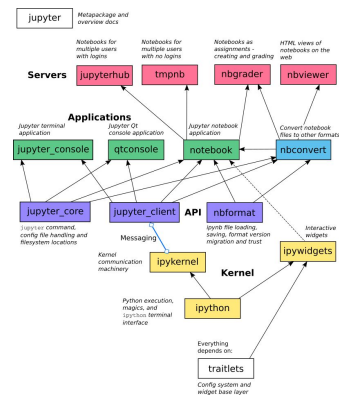


We're not going to cover all of this!

This is really an introductory survey tutorial. We won't cover writing your own kernel or Jupyter Protocol level communications

But we hope to help you get started

- Google, Github etc. are your friends



Outline

- Installation and Running
- Introduction and History
- Jupyter Browser View
- Notebook Tour
- Super Short Intro to Python (break @ ~1 hr)
- Jupyter Architecture
- Scientific Computing - Pandas, Numpy, MPL, Sympy etc.
- Widgets (break @ ~1hr)
- Sharing notebooks
- JupyterLab
- JupyterHub
- Kernels and Customizing Your Environment
- Other moons (wrap up ~1 hr)