

IPython

A multi-language architecture
for interactive computing

[ipython.org \(<http://ipython.org>\)](http://ipython.org)

Fernando Pérez

[fperez.org \(<http://fperez.org>\)](http://fperez.org), [@fperez](http://twitter.com/fperez) [org \(<http://twitter.com/fperez> org\)](http://twitter.com/fperez)

U.C. Berkeley

In [1]: `%pylab inline`
`%run talktools`

Populating the interactive namespace from numpy and matplotlib

Why IPython?

"The purpose of computing is insight, not numbers"

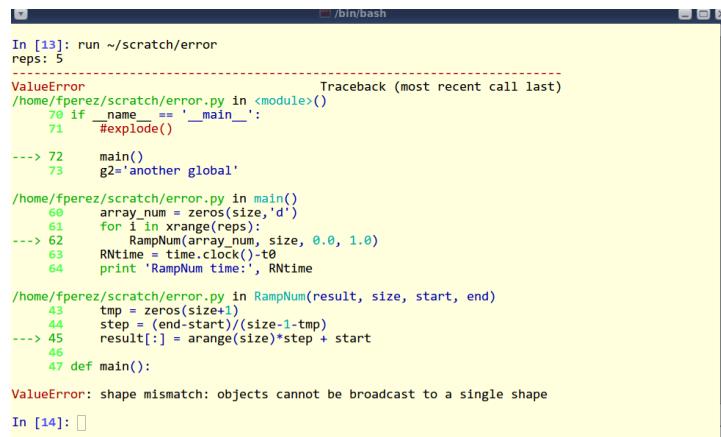
Hamming '62

The Lifecycle of a Scientific Idea (schematically)

1. **Individual** exploratory work
2. **Collaborative** development
3. **Parallel** production runs (HPC, cloud, ...)
4. **Publication** (with reproducible results!)
5. **Education**
6. Goto 1.

From a better shell...

2001, instead of a Physics dissertation

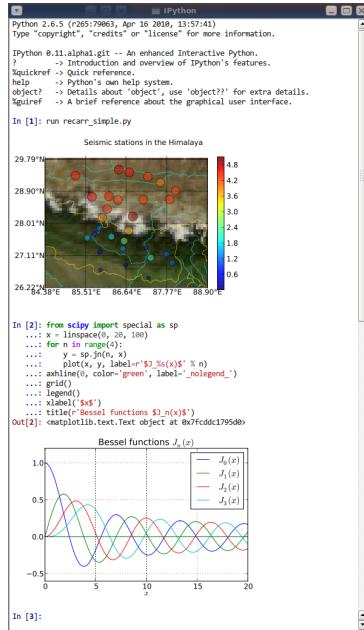


A screenshot of a terminal window titled '/bin/bash'. The window shows a Python script being run. The code is as follows:

```
In [13]: run ~/scratch/error
reps: 5
-----
ValueError: Traceback (most recent call last)
/home/fperez/scratch/error.py in <module>()
 70 if __name__ == '__main__':
 71     #Explode()
 72     main()
 73     g2='another global'
/home/fperez/scratch/error.py in main()
 60     array_num = zeros(size,'d')
 61     for i in xrange(reps):
 62         RampNum(array_num, size, 0.0, 1.0)
 63         RNtime = time.clock()-t0
 64         print 'RampNum time:', RNtime
/home/fperez/scratch/error.py in RampNum(result, size, start, end)
 43     tmp = zeros(size+1)
 44     step = (end-start)/(size-1-tmp)
 45     result[:]= arange(size)*step + start
 46
 47 def main():
ValueError: shape mismatch: objects cannot be broadcast to a single shape
In [14]:
```

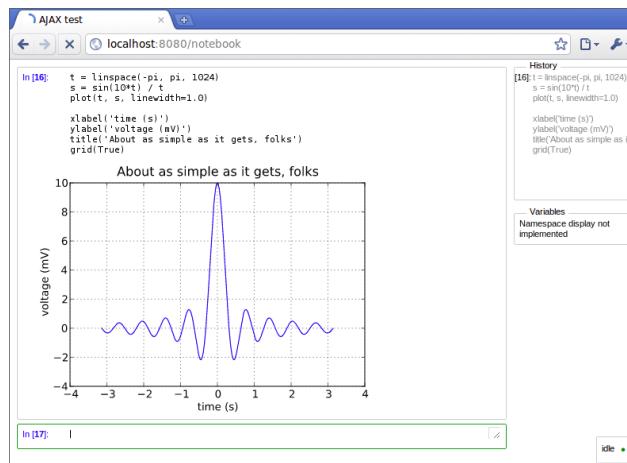
... to a fancier shell (and a protocol!)...

2010, E Patterson, Enthought (key support!)



... 6 notebook attempts over 6 years...

T. Matev, T. Alatalo, R. Kern, Min RK, J. Gao, B. Granger

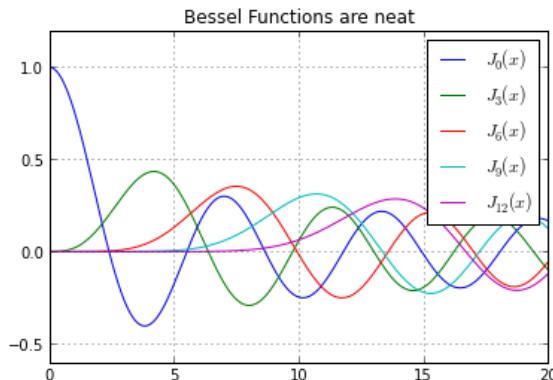


... up to today's IPython Notebook

Everyday exploratory computing, with annotations

Let's plot the Bessel functions J_i for $i = 0, 3, 6, 9$:

```
In [2]: import scipy.special as spec
x = linspace(0, 20, 200)
for n in range(0,13,3):
    plot(x, spec.jn(n, x), label=r'$J_{%i}(x)$' % n)
grid()
legend()
title('Bessel Functions are neat');
```



Explore parameters interactively

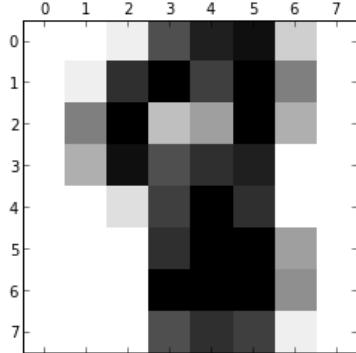
```
In [4]: @interactive(x=(1, 10))
def f(x):
    print 'X is:', x
```



```
X is: 5
```

```
In [5]: from sklearn import datasets  
  
digits = datasets.load_digits()  
n = len(digits.images)  
  
def view_image(i):  
    @interactive(i=(0,n-1))  
    print 'Classif. Label:', digits.target[i]  
    plt.matshow(digits.images[i], cmap=cm.gray_r)  
    plt.show()
```

Classif. Label: 8



Displaying results: a rich protocol

```
In [6]: Image('fig/logo.png')
```

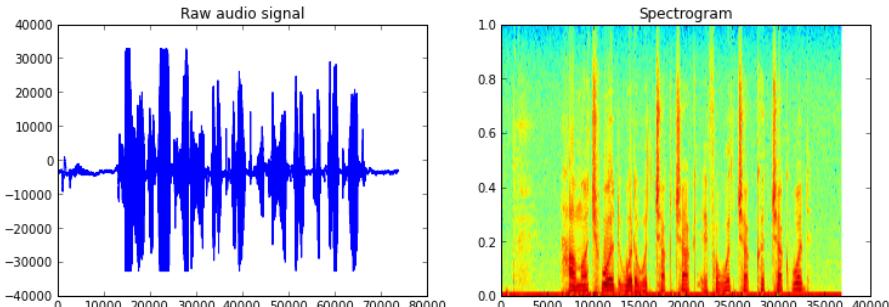
Out[6]: 

```
In [7]: Video('fig/animation.m4v') # Credit: Chris Kees, Army ERDC; created with Proteus.
```

Out[7]:

Browser multimedia + scientific visualization

```
In [8]: plot_audio('voice.wav')
Audio('voice.wav')
```



```
Out[8]:
voice.wav:
```



LaTeX and symbolic mathematics support

```
In [9]: Math(r'F(k) = \int_{-\infty}^{\infty} f(x) e^{2\pi i k} dx')
```

$$F(k) = \int_{-\infty}^{\infty} f(x) e^{2\pi i k} dx$$

```
In [10]: from sympy import symbols, Eq, factor, init_printing, expand
init_printing(use_latex=True)
x, y = symbols("x y")
```

```
In [11]: eq = ((x+y)**3 * (x+1))
display(Eq(eq, expand(eq)))
```

$$(x + 1)(x + y)^3 = x^4 + 3x^3y + x^3 + 3x^2y^2 + 3x^2y + xy^3 + 3xy^2 + y^3$$

```
In [12]: @interactive(n=(1,10))
def _(n):
    eq = ((x+y)**n * (x+1))
    display(Eq(eq, expand(eq)))
```

$$(x + 1)(x + y)^5 = x^6 + 5x^5y + x^5 + 10x^4y^2 + 5x^4y + 10x^3y^3 + 10x^3y^2 + 5x^2y^4 + 10x^2y^3 + xy^5$$

Dynamic graph visualizations with D3

Talk to other languages: R (or Octave)

```
In [13]: X = np.array([0,1,2,3,4])
Y = np.array([3,5,4,6,7])
# Now, load R support
%load_ext rmagic
```

```
In [14]: %%R -i X,Y -o XYcoef
XYlm = lm(Y~X)
XYcoef = coef(XYlm)
print(summary(XYlm))
par(mfrow=c(2,2))
plot(XYlm)
```

Call:

```
lm(formula = Y ~ X)
```

Residuals:

| 1 | 2 | 3 | 4 | 5 |
|------|-----|------|-----|-----|
| -0.2 | 0.9 | -1.0 | 0.1 | 0.2 |

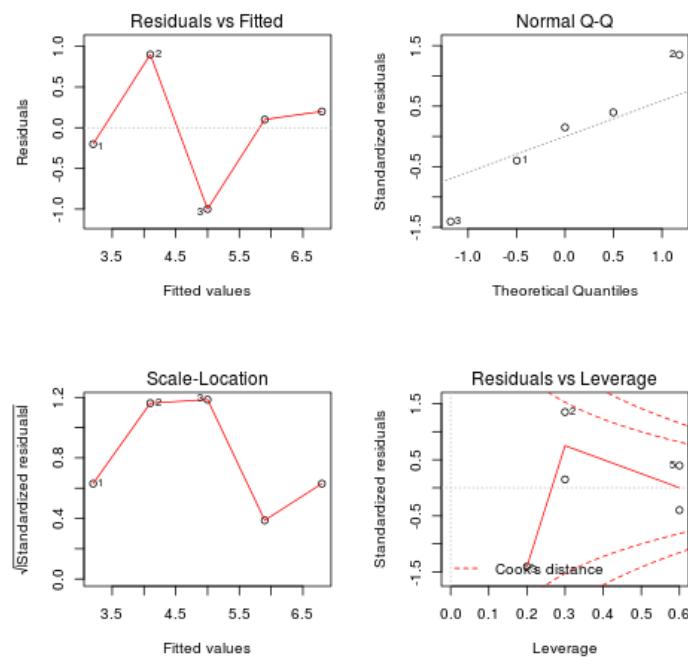
Coefficients:

| | Estimate | Std. Error | t value | Pr(> t) |
|----------------|----------|------------|----------|----------|
| (Intercept) | 3.2000 | 0.6164 | 5.191 | 0.0139 * |
| X | 0.9000 | 0.2517 | 3.576 | 0.0374 * |
| --- | | | | |
| Signif. codes: | 0 '***' | 0.001 '**' | 0.01 '*' | 0.05 '.' |
| ' 1 | | | | |

Residual standard error: 0.7958 on 3 degrees of freedom

Multiple R-squared: 0.81, Adjusted R-squared: 0.7467

F-statistic: 12.79 on 1 and 3 DF, p-value: 0.03739



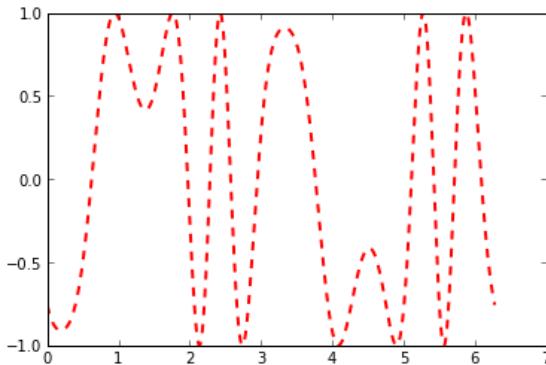
Julia: seamless 2-way communication

```
In [15]: %load_ext julia.magic
%julia @pyimport matplotlib.pyplot as plt
%julia @pyimport numpy as np

Initializing Julia interpreter. This may take some time...
```

```
In [16]: %%julia
# Note how we mix numpy and julia:
x = linspace(0,2*pi,1000); # use the julia linspace
y = sin(3*x + 4*np.cos(2*x)); # use the numpy cosine and julia sine
plt.plot(x, y, color="red", linewidth=2.0, linestyle="--")
```

```
Out[16]: [Line2D(_line0)]
```



Julia and Python

The call stack as a layer cake

```
In [17]: jfib = %julia jfib(n, fib) = n < 2 ? n : fib(n-1, jfib) + fib(n-2, jfib)
)

def pyfib(n, fib):
    if n < 2:
        return n
    return fib(n-1, pyfib) + fib(n-2, pyfib)

pyfib(20, jfib) # Credit: Steven Johnson, MIT.
```

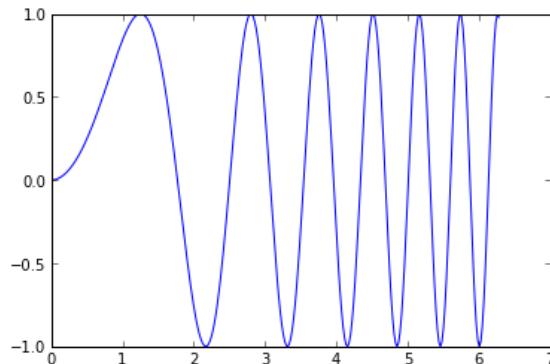
```
Out[17]: 6765
```

Fortran integration

```
In [18]: %load_ext fortranmagic
```

```
In [22]: %%fortran
subroutine f1(x, y, n)
    real, intent(in), dimension(n) :: x
    real, intent(out), dimension(n) :: y
    !intent(hide) :: n
    y = sin(x**2)
end subroutine f1
```

```
In [23]: x = np.linspace(0, 2*pi, 300)
plot(x, f1(x));
```



nbconvert: notebook manipulation tools

- Library for format conversions
- HTML, LaTeX, PDF output.
- Basic command-line usage.
- Rich customization possibilities.

NBViewer: Share notebooks online

Matthias Bussonnier, 2012

```
In [24]: website('nbviewer.ipython.org')
```

```
Out[24]: nbviewer.ipython.org  
(http://nbviewer.ipython.org)
```

IPython Notebook Viewer

A simple way to share IPython Notebooks

Programming Languages

Reproducible research...

**Min RK, Brian, Greg Caporaso, Rob
Knight (group), Justin Riley**

```
In [25]: website('http://www.nature.com/ismej/journal/v7/n3/full/ismej2012123a.html', 'Paper:')
```

Out[25]:

Paper: (<http://www.nature.com/ismej/journal/v7/n3/full/ismej2012123a.html>)

[nature.com homepage](#)

- [Jump to main content](#)

nature.com

'The world's best science and medicine on your desktop'

- [Publications A-Z index](#)
- [Browse by subject](#)

Search [Advanced search](#)

Server Error

**Notebooks and AMI for
reproducibility:**

```
In [26]: website('http://qiime.org/home_static/nih-cloud-apr2012', 'Companion We  
bsite:')
```

```
Out[26]: Companion Website: (http://qiime.org  
/home\_static/nih-cloud-apr2012)
```

Instructions and supporting data for the QIIME/IPython /StarCluster demo at the 2012 NIH Cloud Computing the Microbiome workshop and our corresponding paper in the ISMI Journal.

The analysis made use of the [IPython Notebook](#), [QIIME](#), [StarCluster](#), [PyCogent](#) and [PrimerProspector](#). All of these tools are pre-installed in the `ami-9f69c1f6` public Amazon EC2 instance, which was used in this study.

Supporting Files

The IPython notebooks supporting this study can be viewed [here](#) and are available here in PDF format:

- [NIH Cloud Demo \(Complete\)](#)
-

Notebook-based technical blogs

Jake VanderPlas (astronomy/ML at UW)

```
In [27]: website('http://jakevdp.github.io/blog/2013/12/05/static-interactive-wi  
dgets',  
           'Pythonic Perambulations')
```

Out[27]:

[Pythonic Perambulations](http://jakevdp.github.io/blog/2013/12/05/static-interactive-widgets)
([http://jakevdp.github.io/blog/2013/12/05](http://jakevdp.github.io/blog/2013/12/05/static-interactive-widgets)
/static-interactive-widgets)

Pythonic Perambulations **(<http://jakevdp.github.io/>)**

Musings and ramblings through the world of Python and beyond

- Atom (/atom.xml)

- Archives (/archives.html)
- Home Page (<http://www.astro.washington.edu/users/vanderplas>)

Static Interactive Widgets for IPython Notebooks

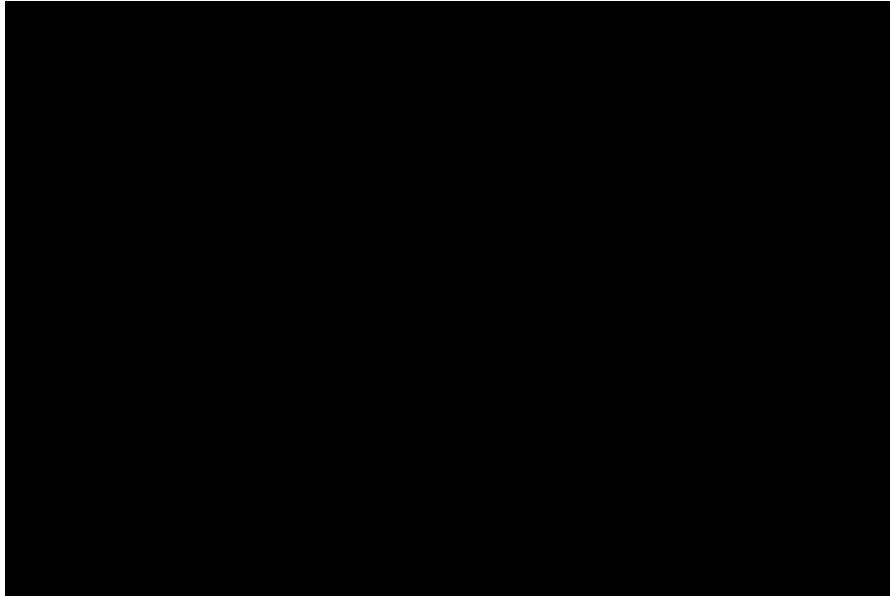
Training IBM Watson

Using Notebook and `IPython.parallel`

- **From:** 8000 lines of Java, JS and HTML5, 2min/query
- **To:** 200 lines of Python, 2sec/query

In [28]: YouTubeVideo('tlontoyWX70', start='800', width=600, height=400)

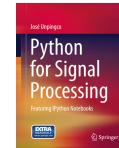
Out[28]:



Books: "Literate Computing"

Python for Signal Processing, By [Jose Unpingco](https://github.com/unpingco/Python-for-Signal-Processing) (<https://github.com/unpingco/Python-for-Signal-Processing>)

- Springer hardcover (<http://link.springer.com/book/10.1007/978-3-319-01342-8/page/1>) book
- Each chapter is an IPython Notebook
- Posted as a [blog entry](http://python-for-signal-processing.blogspot.com/) (<http://python-for-signal-processing.blogspot.com/>)
- And all available as a [Github repo](https://github.com/unpingco/Python-for-Signal-Processing) (<https://github.com/unpingco/Python-for-Signal-Processing>)



... "Probabilistic Programming and Bayesian Methods for Hackers"

By [Cam Davidson Pilon](http://www.camdp.com) (<http://www.camdp.com>).

```
In [29]: website('camdavidsonpilon.github.io/Probabilistic-Programming-and-Bayesian-Methods-for-Hackers',  
           'Probabilistic programming...')
```

```
Out[29]: Probabilistic programming...  
(http://camdavidsonpilon.github.io/Probabilistic-Programming-and-Bayesian-Methods-for-Hackers)
```

[View on GitHub](#)

Probabilistic Programming & Bayesian Methods for Hackers

[Download this project as a .zip file](#) [Download this project as a tar.gz file](#)

An intro to Bayesian methods and probabilistic programming from a computation/understanding first, mathematics-second point of view.

1. [Prologue](#)
2. [Contents](#)

... "Mining the Social Web"

By [Matthew Russell \(<https://github.com/ptwobrussell>\)](#).

- | | |
|--|---|
| <ul style="list-style-type: none">• O'Reilly (http://shop.oreilly.com/product/0636920010203.do) book• Companion Notebook collection• All available as a Github repo (https://github.com/ptwobrussell/Mining-the-Social-Web-2nd-Edition) |  |
|--|---|



CS 109: Data Science at Harvard

```
In [30]: website('http://cs109.org/homework/homework.php', 'CS 109 Homeworks')
```

```
Out[30]: CS 109 Homeworks (http://cs109.org/homework/homework.php)
```



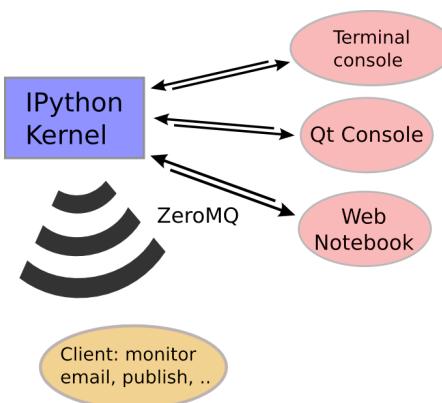
CS109 Data Science

- [Home](#)
- [Piazza](#)
- [Syllabus](#)
- [Schedule](#)
- [Homework](#)
- [Readings](#)
- [Projects](#)
- [Resources](#)

A growing (and amazing) body of work

For much more: (<http://ipython.org/gallery>)
<http://ipython.org/gallery>
(<http://ipython.org/gallery>)

A simple and generic architecture



IPython.parallel

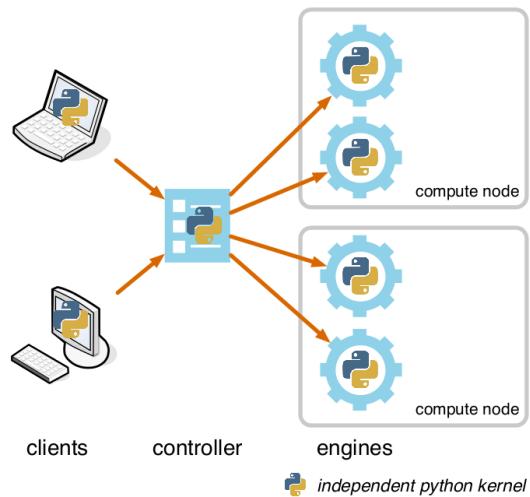


Image credit: Continuum Analytics.

A protocol that enables kernels in other languages

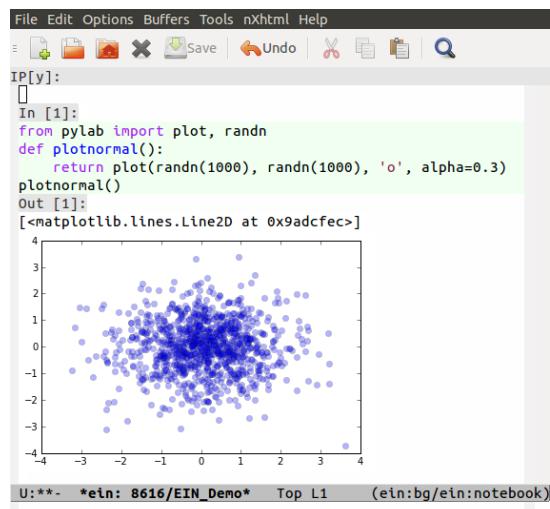
- [IJulia \(<http://nbviewer.ipython.org/url/dj.mit.edu/~stevenj/IJulia%20Preview.ipynb>\)](http://nbviewer.ipython.org/url/dj.mit.edu/~stevenj/IJulia%20Preview.ipynb)
- [IHaskell \(<http://nbviewer.ipython.org/github/gibiansky/IHaskell/blob/master/demo/IHaskell.ipynb>\)](http://nbviewer.ipython.org/github/gibiansky/IHaskell/blob/master/demo/IHaskell.ipynb)
- [IFSharp \(<http://nbviewer.ipython.org/github/BayardRock/IfSharp/blob/master/Feature%20Notebook.ipynb>\)](http://nbviewer.ipython.org/github/BayardRock/IfSharp/blob/master/Feature%20Notebook.ipynb)
- [IRuby \(<http://nbviewer.ipython.org/github/minad/iruby/blob/master/IRuby-Example.ipynb>\)](http://nbviewer.ipython.org/github/minad/iruby/blob/master/IRuby-Example.ipynb)
- [IGo \(<https://github.com/takluyver/igo>\)](https://github.com/takluyver/igo)
- [IScala \(<https://github.com/mattppap/IScala>\)](https://github.com/mattppap/IScala)
- In planning stages: R and Matlab.

The interactive widget architecture is also 100% language-agnostic

A healthy ecosystem, OSS and commercial

Emacs IPython Notebook Client

Takafumi Arakaki



The screenshot shows a window titled "File Edit Options Buffers Tools nXhtml Help". Below the menu is a toolbar with icons for file operations like Open, Save, Undo, and Search. The main area is labeled "IP[y]:". It contains an "In [1]" cell with Python code to generate a scatter plot:

```
In [1]:
from pylab import plot, randn
def plotnormal():
    return plot(randn(1000), randn(1000), 'o', alpha=0.3)
plotnormal()
Out [1]:
[<matplotlib.lines.Line2D at 0x9adcfc>]
```

Below the code is a scatter plot of blue points centered around (0,0) with axes ranging from -4 to 4.

At the bottom, the status bar shows "U:**- *ein: 8616/EIN_Demo* Top L1 (ein:bg/ein:notebook)".

Vim client

Paul Ivanov @Berkeley

```
# Visual Mode
# It also works blockwise in Visual Mode.
# Select the next block and send it to IPython

import this,math # secret decoder ring
a,b,c,d,e,f,g,h,i = range(1,10)
msg = "I'm a big frag schm I've \nknown you\n "+this.s.split()[1][g]
decode=lambda x: '\n'.join([(this.d.get(c),c) for c in x])+''
format=lambda X: X[0]+X[1].ljust(9)+'\n'+X[2].ljust(9)+'\n'+X[3].ljust(9)+'\n'+X[4].ljust(9)+'\n'+X[5].ljust(9)+'\n'+X[6].ljust(9)+'\n'+X[7].ljust(9)+'\n'+X[8].ljust(9)+'\n'+X[9].ljust(9)+'\n'

# Then, go to the qtconsole and run this line
print secret_decoder('x,_')

##  
*** 4 lines: Running whole file  
##  
*** 22 lines: IPython's object? Functionality  
*** 33 lines: IPython's tab-completion Functionality
```

test.py()

35 0x23 23,1(master) Bot

Out[4]: <��ුඩා 0s 11ms /usr/lib/python2.7/zipfile.py>

In [3]: from vim

NameError: name 'from_vim' is not defined

recent call last)

/home/pi/code/vim-ipython/<ipython-input-3-72ba696b990>: in

<module>

>>> 3 from vim

NameError: name 'from_vim' is not defined

In [5]: from vim

Out[5]: 1

In [7]: from vim

Out[7]: 14

In [8]:

18:45:pm:python maxwell \$ ipython qtconsole

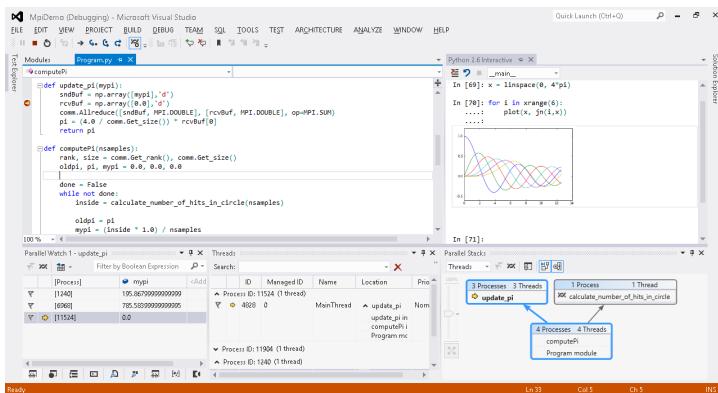
[IPKernelApp] To connect another client to this kernel, use:

[IPKernelApp] --existing -shell=5125 -ip=0.0.0.0 -stdin=5000 -hb=43400

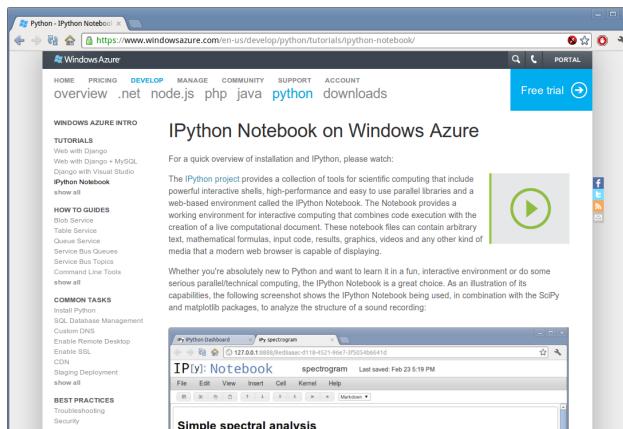
Enthought Canopy

Microsoft Python Tools for Visual Studio and Azure

Shahrokh Mortazavi, Dino Viehland, Wenming Ye, Dennis Gannon. Thanks!!

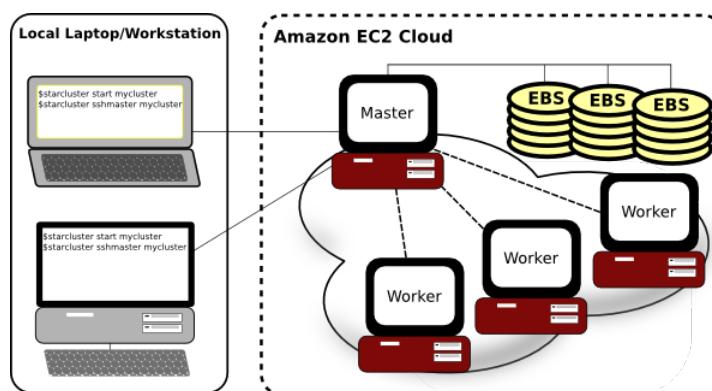


Microsoft Azure (S. Mortazavi, W. Ye)



StarCluster

Justin Riley @ MIT



Continuum Analytics: Wakari

The screenshot shows a Wakari notebook interface. The title bar says "Lecture 3 Scipy". The main area has a green button "Run/Edit this Notebook". Below it, a section titled "Example: double pendulum" contains text and code. The code in "In [15]" is:

```
In [15]: Image(url='http://upload.wikimedia.org/wikipedia/commons/c/c9/Double-compound-pendulum-dimensioned.svg')
```

The output "Out[15]" shows a diagram of a double pendulum. The first link is of length ℓ and mass m , pivoted at the origin $(0,0)$. It makes an angle θ_1 with the vertical. A second link of length ℓ and mass m hangs from the end of the first, making an angle θ_2 with the vertical. The center of mass of the second link is at (x_2, y_2) . The diagram includes coordinate axes and labels for angles θ_1 and θ_2 .

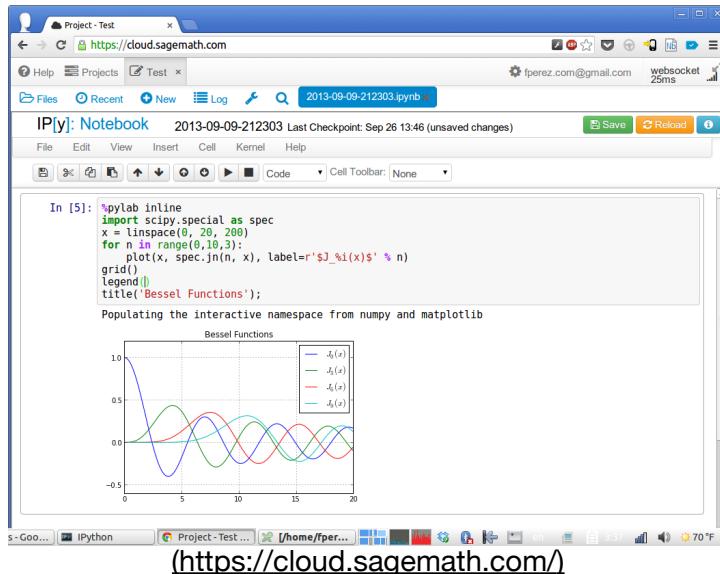
The equations of motion of the pendulum are given on the wiki page:

$$\dot{\theta}_1 = \frac{6}{m\ell^2} \frac{2p_{\theta_1} - 3\cos(\theta_1 - \theta_2)p_{\theta_2}}{16 - 9\cos^2(\theta_1 - \theta_2)}$$
$$\dot{\theta}_2 = \frac{6}{m\ell^2} \frac{8p_{\theta_2} - 3\cos(\theta_1 - \theta_2)p_{\theta_1}}{16 - 9\cos^2(\theta_1 - \theta_2)}.$$
$$\dot{p}_{\theta_1} = -\frac{1}{2}m\ell^2[\dot{\theta}_1\dot{\theta}_2\sin(\theta_1 - \theta_2) + 3\frac{g}{\ell}\sin\theta_1]$$
$$\dot{p}_{\theta_2} = -\frac{1}{2}m\ell^2[-\dot{\theta}_1\dot{\theta}_2\sin(\theta_1 - \theta_2) + \frac{g}{\ell}\sin\theta_2]$$

To make the Python code simpler to follow, let's introduce new variable names and the vector notation: $x = [\theta_1, \theta_2, p_{\theta_1}, p_{\theta_2}]$

$$\dot{x}_1 = \frac{6}{m\ell^2} \frac{2x_3 - 3\cos(x_1 - x_2)x_4}{16 - 9\cos^2(x_1 - x_2)}$$

Cloud.sagemath.com: UW startup, W. Stein



Plot.ly: interactive JS plots in Python

```
In [31]: website('http://nbviewer.ipython.org/gist/jackparmer/7729584', 'Rosling Countries', width=900, height=600)
```

```
Out[31]: Rosling Countries \(http://nbviewer.ipython.org/gist/jackparmer/7729584\)
```

Bubble Charts & Hover Text with Plotly

I'm Jack Parmer

Plotly is like graphing crack. It standardizes the graphing interface across scientific computing languages (Python, R, MATLAB, ect) while giving rich interactivity and web shareability that has been possible before with matplotlib, ggplot, MATLAB, ect. On the Plotly website, you can publish your graphs with a GUI, so you don't have to spend hours writing code that simply changes the legend opacity.

Plotly does this all while backing up your graphs on the cloud, so that years later, you can find that may have otherwise been on a harddrive in a landfill. If you make your data public, *other people* can also find your graphs and data. The best practice that we have today for saving and sharing research data is to entomb it as a thesis in the engineering library basement. All that is changing.

Like d3.js? Like interactive, NYT graphics? So do we. Now, with the [Plotly APIs](https://plot.ly) (<https://plot.ly>), you can make them yourself without being an expert web programmer. If you are an expert programmer, now you have scientific languages and tools like R, Python, Pandas, and MATLAB instead of javascript to wrangle your data and create beautiful data vis. Science meets the wide-web. Engineering meets design. Let's do this.

I'm going to show you this brave new world below, starting with bubble charts. Bubble charts are sweet because they take advantage of the innate interactivity of Plotly graphs. When you hover over a bubble chart point, you want to see what its size represents, you want to zoom-in to points that are clustered, and you want to pan around once you're zoomed-in. You become a Bubble Chart Explorer. Plotly lets you do all this, all while upping the game for scientific, publication-quality graphics.

The Core IPython team (2013)



Brian Granger



Min Ragan-Kelley



Thomas Kluyver



Matthias Bussonnier



Paul Ivanov



Brad Froehle



Jörgen Stenarson



Robert Kern



Evan Patterson



Jonathan March

Plus [Jonathan Frederic](https://github.com/jdfreder) (<https://github.com/jdfreder>) and [Zach Sailer](https://github.com/Zsailer) (<https://github.com/Zsailer>)!

An incomplete cast of characters

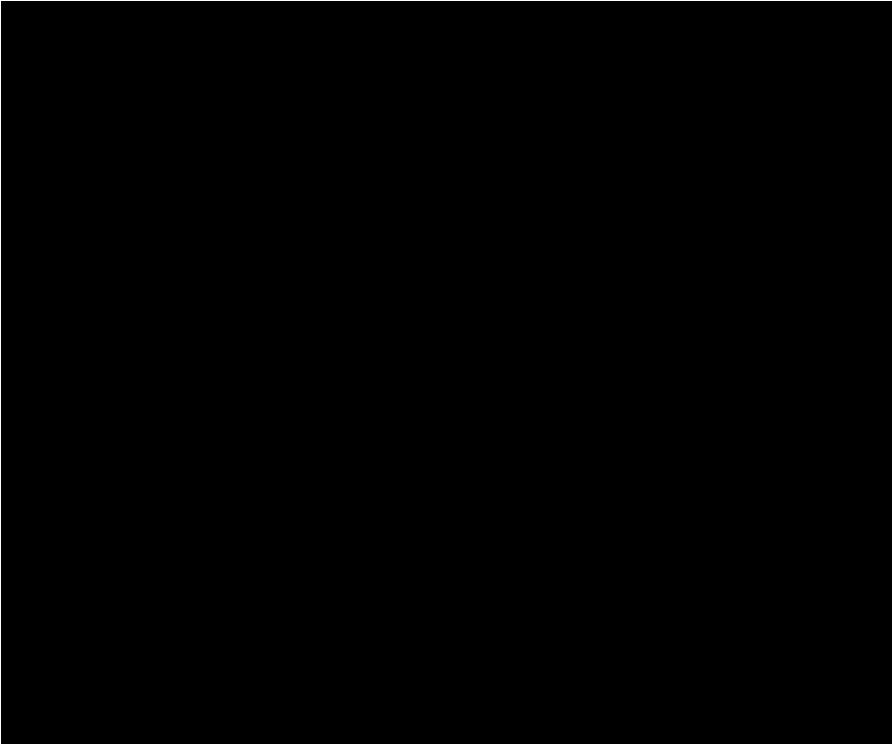
- *Brian Granger* - Physics, Cal State San Luis Obispo
- *Min Ragan-Kelley* - Nuclear Engineering, UC Berkeley
- *Matthias Bussonnier* - Physics, Institut Curie, Paris
- *Jonathan March*- Enthought
- *Thomas Kluyver* - Biology, U. Sheffield
- *Jörgen Stenarson* - Elect. Engineering, Sweden.
- *Paul Ivanov* - Neuroscience, UC Berkeley.
- *Robert Kern* - Enthought
- *Evan Patterson* - Physics, Caltech/Enthought
- *Brad Froehle* - Mathematics, UC Berkeley
- *Stefan van der Walt* - UC Berkeley
- *John Hunter* - TradeLink Securities, Chicago.
- *Prabhu Ramachandran* - Aerospace Engineering, IIT Bombay.
- *Satra Ghosh*- MIT Neuroscience
- *Gaël Varoquaux* - Neurospin (Orsay, France)
- *Ville Vainio* - CS, Tampere University of Technology, Finland
- *Barry Wark* - Neuroscience, U. Washington.
- *Ondrej Certik* - Physics, LANL
- *Darren Dale* - Cornell
- *Justin Riley* - MIT
- *Mark Voorhies* - UC San Francisco
- *Nicholas Rougier* - INRIA Nancy Grand Est
- *Thomas Spura* - Fedora project

Many more! (~220 commit authors)

Public "Lab meetings on air"

In [32]: YouTubeVideo('UUjTAq8cCcs', width=600, height=500)

Out[32]:



Current IPython funding



SIMONS FOUNDATION



Note: We're hiring! Machine learning web pipelines for time-series analysis (astro, geo, neuro). With Josh Bloom (UCB Astro), NSF funding. Talk to me!

Prior IPython support, thanks!!

- **Enthought**, Austin, TX: Lots!
- **Microsoft**: WinHPC support, Visual Studio integration, Azure
- **DoD/DRC Inc**: 2011/12 (thanks to Jose Unpingco and Chris Keees).
- Indirect: NIH via NiPy grant, NSF via Sage grant.
- Google: summer of code 2005, 2010.
- Tech-X Corp., Boulder, CO: Parallel/notebook (previous versions)

A new \$37.8M initiative in Data Science

- Moore/Sloan Foundations, 5 year support for *UC Berkeley, U. Washington, NYU*.
- Open source scientific computing tools will be central to this effort.
- **We're hiring:** Executive Director (now) (<http://vcresearch.berkeley.edu/datascience/career-opportunities>), Data Science Fellows (soon).



- My talk from today is a slightly updated version of this one from NIPS last December:

<http://nbviewer.ipython.org/urls/raw.githubusercontent.com/fperez/talk-1312-nips/master/IPython-Interactive-Computing.ipynb>

- The interactive widgets and other new features are documented in this notebook collection:

<http://nbviewer.ipython.org/urls/raw.githubusercontent.com/ipython/ipython/master/examples/Index.ipynb>

(that's the link I tried to open on nbviewer which failed, we just moved the URL yesterday).

- A gallery of interesting IPython Notebooks:

<https://github.com/ipython/ipython/wiki/A-gallery-of-interesting-IPython-Notebooks>

- Some videos in case folks are interested:

<http://ipython.org/videos.html>