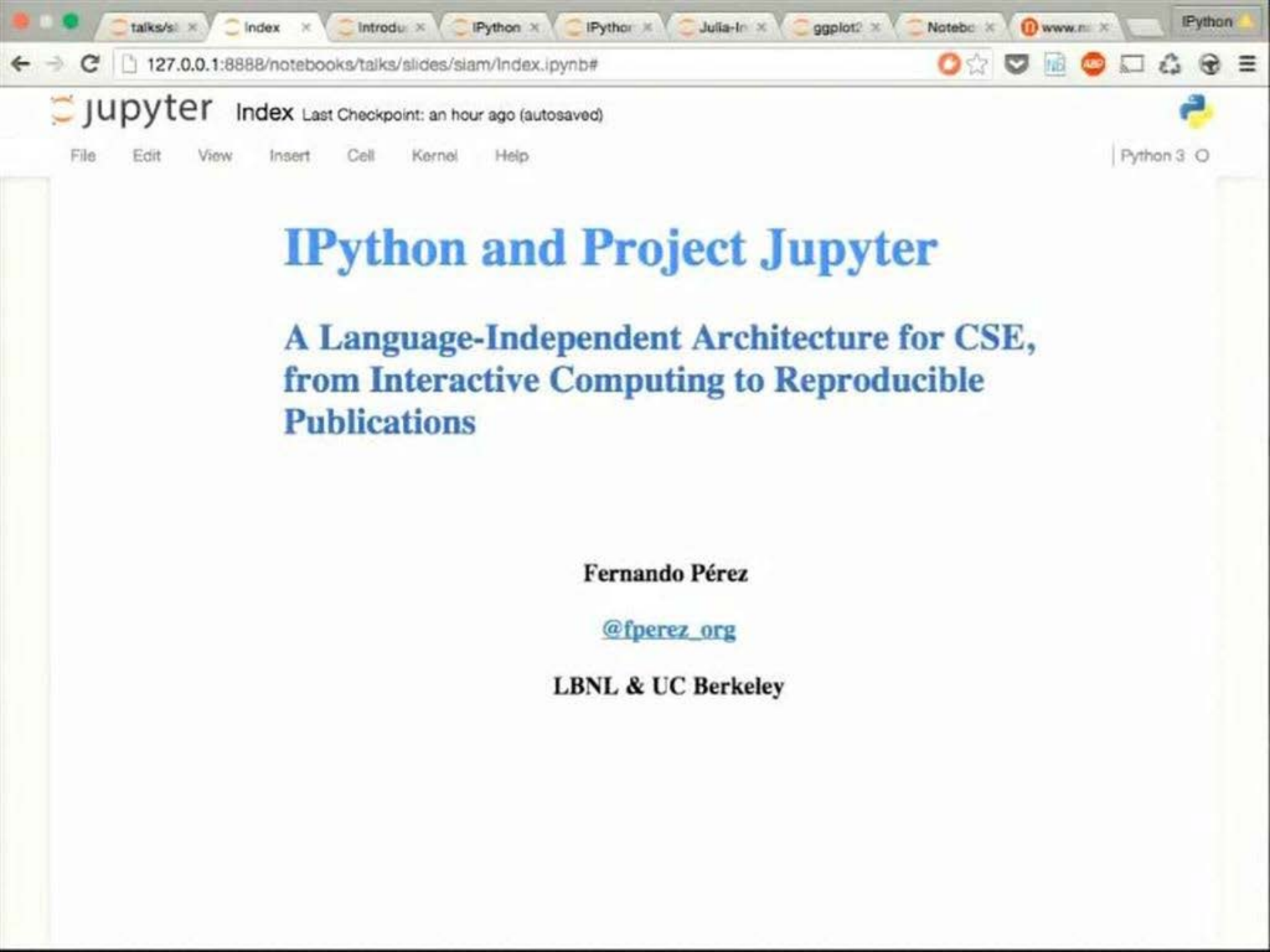

talks/s
Index
Introdu
IPython
IPythor
Julia-In
ggplot2
Notebc
www.n
IPython
127.0.0.1:8888/notebooks/talks/slides/siam/Index.ipynb#
jupyter
Index
Last Checkpoint: an hour ago (autosaved)
File
Edit
View
Insert
Cell
Kernel
Help
Python 3
IPython and Project Jupyter
A Language-Independent Architecture for CSE, from Interactive Computing to Reproducible Publications
Fernando Pérez
@fperez_org
LBNL & UC Berkeley

talks/s | index | introdu | IPython | IPytho | Julia-l | ggplot | Noteb | www.n | IPython

127.0.0.1:8888/notebooks/talks/slides/siam/IPython.ipynb#

jupyter IPython (autosaved)

File Edit View Insert Cell Kernel Help | Python 3

IPython Core Developers

click to expand output; double click to hide output

 Fernando Perez	 Brian Granger	 Min Ragan-Kelley	 Thomas Kluyver	 Matthias Bussonnier
 Jonathan Frederic	 Paul Ivanov	 Damian Avila	 Kyle Kelley	 Zach Sailer
 Jorgen Stenarson	 Jonathan March	 Brad Froehle	 Evan Patterson	 Robert Kern

Google

- Kayur Patel
- Kester Tong
- Mark Sandler
- Corinna Cortes

Bloomberg

- Jason Grout
- Sylvain Corlay

yt/NCSA

- Matthew Turk

Current IPython funding ¶

ALFRED P. SLOAN FOUNDATION

SIMONS FOUNDATION

POWERED BY rackspace the open cloud company

Microsoft

Google

```
from datetime import datetime
```

In [3]:
```
stock = web.DataReader( 'MSFT', 'yahoo', start=datetime(2012, 1, 1))
display(stock[:3])
stock['Close'].plot(title='Closing Price');
```

	Open	High	Low	Close	Volume	Adj Close
Date						
2012-01-03	26.55	26.96	26.39	26.77	64731500	24.42
2012-01-04	26.82	27.47	26.78	27.40	80516100	25.00
2012-01-05	27.38	27.73	27.29	27.68	56081400	25.25

jupyter IPython (unsaved changes)

File Edit View Insert Cell Kernel Help | Python 3

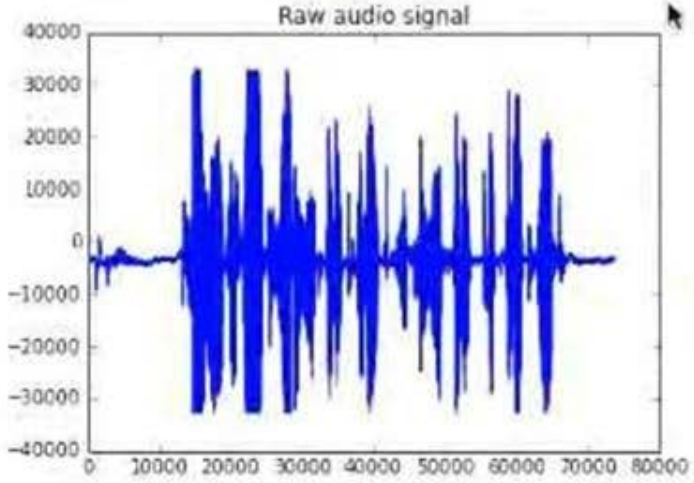
```
from datetime import datetime
```

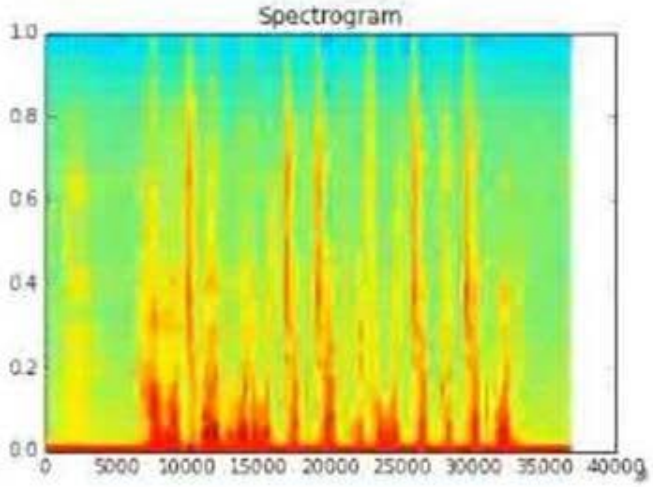
In [7]:
```
stock = web.DataReader( 'IBM', 'yahoo', start=datetime(2012, 1, 1))
display(stock[:3])
stock['Close'].plot(title='Closing Price');
```

	Open	High	Low	Close	Volume	Adj Close
Date						
2012-01-03	186.73	188.71	186.00	186.30	5646000	174.25
2012-01-04	185.57	186.33	184.94	185.54	4346700	173.53
2012-01-05	184.81	185.03	183.10	184.66	4463100	172.71

Browser multimedia + scientific visualization

```
In [4]: plot_audio('voice.wav')
        Audio('voice.wav')
```



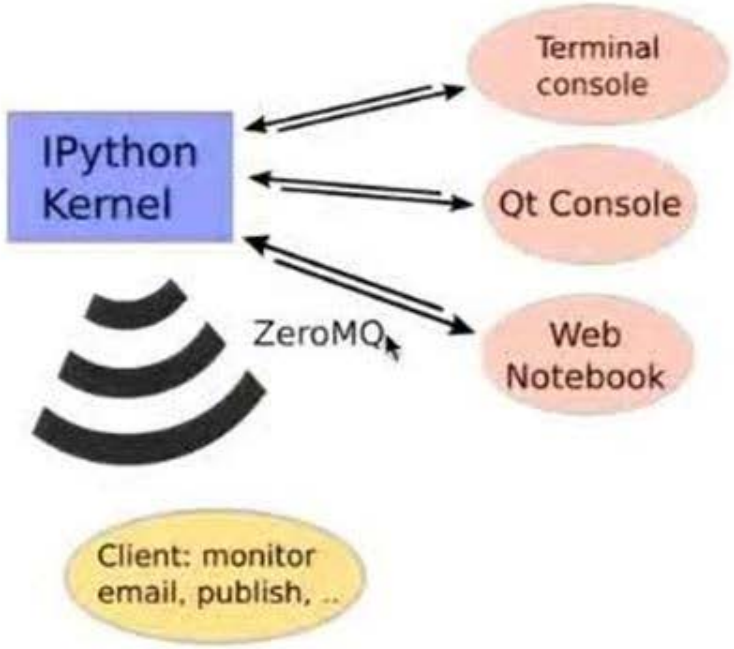


Out[4]:

A simple and generic architecture

0:04

A simple and generic architecture



Not just about Python: Kernels in any language

1. IPython "Official", we ship it

jupyter IPython-Multiple languages from inside IPython Last Checkpoint: an hour ago (autosaved)

The IPython kernel is a polyglot

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

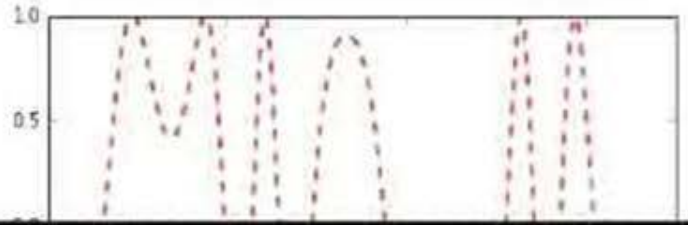
Julia: seamless 2-way integration

In [2]:
```
%load_ext julia.magic
%julia @pyimport matplotlib.pyplot as plt
%julia @pyimport numpy as np
```

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

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

Out[3]:
```
|<matplotlib.lines.Line2D at 0x11bb96b38>|
```



jupyter IPython-Multiple languages from inside IPython Last Checkpoint: an hour ago (autosaved)

File Edit View Insert Cell Kernel Help Python 3

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

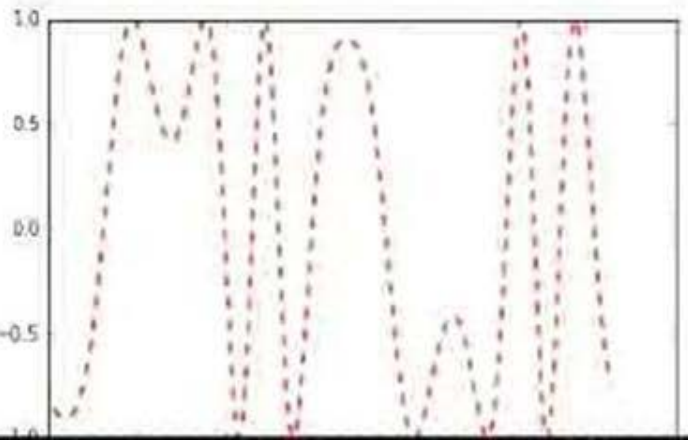
Julia: seamless 2-way integration

In [2]:
```
%load_ext julia.magic
%julia #pyimport matplotlib.pyplot as plt
%julia #pyimport numpy as np
```

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

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

Out[3]:
```
[<matplotlib.lines.Line2D at 0x11bb96b38>]
```



jupyter IPython-Multiple languages from inside IPython Last Checkpoint: an hour ago (autosaved)

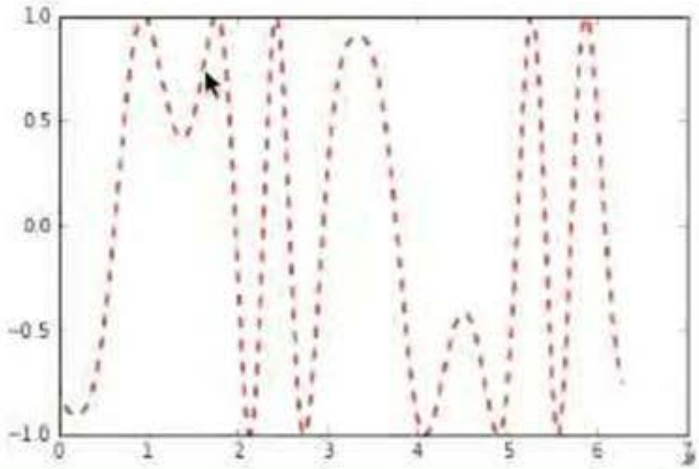
File Edit View Insert Cell Kernel Help | Python 3

```
%julia @pyimport matplotlib.pyplot as plt
%julia @pyimport numpy as np
```

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

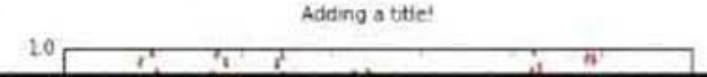
In [3]:
```
%%julia
# Note how we mix numpy and julia:
t = linspace(0,2*pi,1000); # use the julia linspace
s = sin(3*t + 4*np.cos(2*t)); # use the numpy cosine and julia sine
fig = plt.gcf()
plt.plot(t, s, color="red", linewidth=2.0, linestyle="--")
```

Out[3]:
```
[<matplotlib.lines.Line2D at 0x11bb96b38>]
```



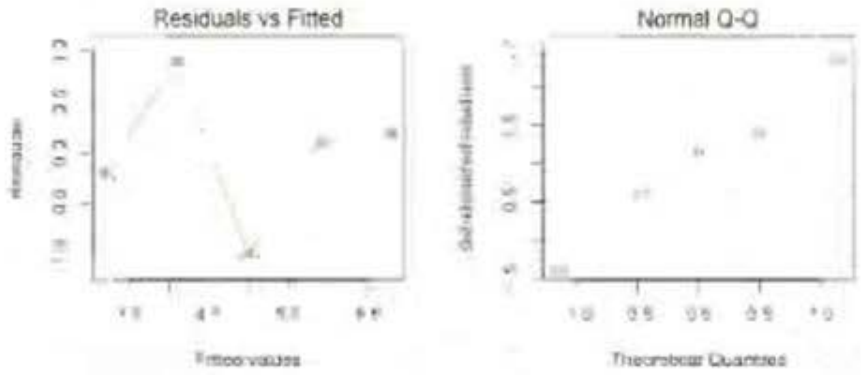
In [4]:
```
fig = %julia fig
fig.suptitle("Adding a title!")
fig
```

Out[4]:



jupyter IPython-Multiple languages from inside IPython Last Checkpoint: an hour ago (autosaved)

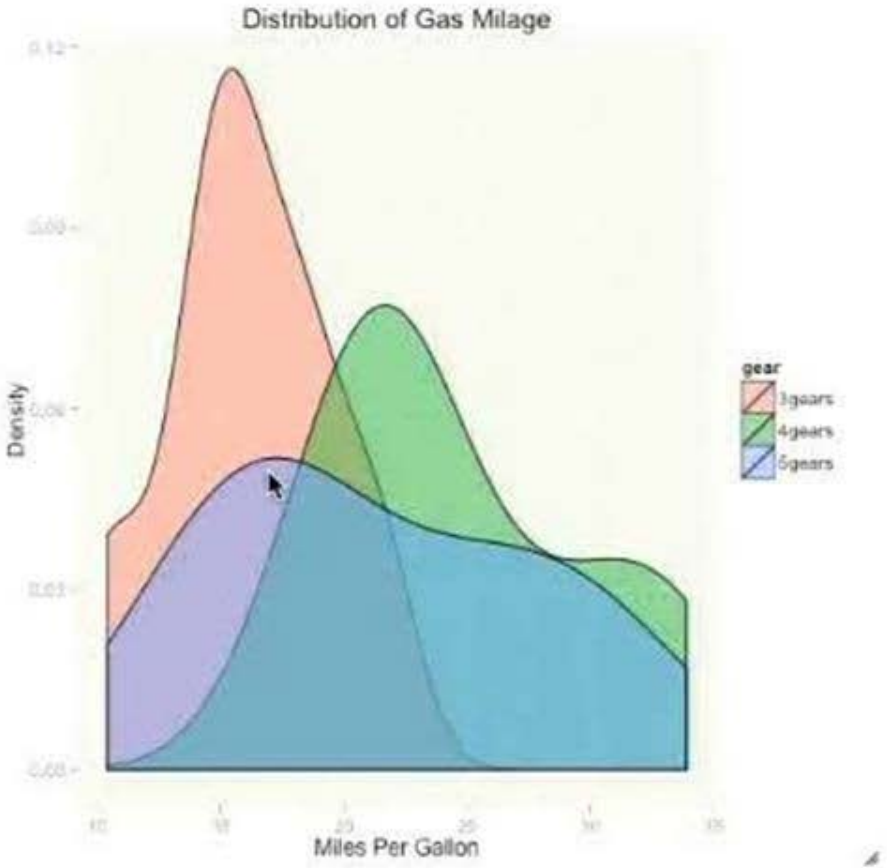

```
plot(XY.lm)


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)   1.2000     0.9165   1.309   0.2817
X             0.8000     0.2517   3.178   0.0503 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7958 on 3 degrees of freedom
Multiple R-squared:   0.77,     Adjusted R-squared:  0.6933
F-statistic: 10.1 on 1 and 3 DF,  p-value: 0.05035
```



jupyter IPython-Multiple languages from inside IPython Last Checkpoint: an hour ago (autosaved)

File Edit View Insert Cell Kernel Help Python 3

```
# Kernel density plots for mpg
# grouped by number of gears (indicated by color)
qplot(mpg, data=mtcars, geom="density", fill=gear, alpha=I(.5),
   main="Distribution of Gas Milage", xlab="Miles Per Gallon",
   ylab="Density")
```



jupyter IPython-Multiple languages from inside IPython Last Checkpoint: an hour ago (autosaved)

Cython: Speed up Python, C integration

In [8]:
```
def f(x):
    return x**2-x

def integrate_f(a, b, N):
    s = 0; dx = (b-a)/N
    for i in range(N):
        s += f(a+i*dx)
    return s * dx
```

In [9]:
```
%load_ext Cython
```

In [10]:
```
%%cython
cdef double fcy(double x) except? -2:
    return x**2-x

def integrate_fcy(double a, double b, int N):
    cdef int i
    cdef double s, dx
    s = 0; dx = (b-a)/N
    for i in range(N):
        s += fcy(a+i*dx)
    return s * dx
```

In [11]:
```
%timeit integrate_f(0, 1, 100)
%timeit integrate_fcy(0, 1, 100)
```

```
10000 loops, best of 3: 33.8 µs per loop
The slowest run took 5.90 times longer than the fastest. This could mean that an intermediate
result is being cached
1000000 loops, best of 3: 319 ns per loop
```

jupyter IPython-Multiple languages from inside IPython Last Checkpoint: an hour ago (autosaved)

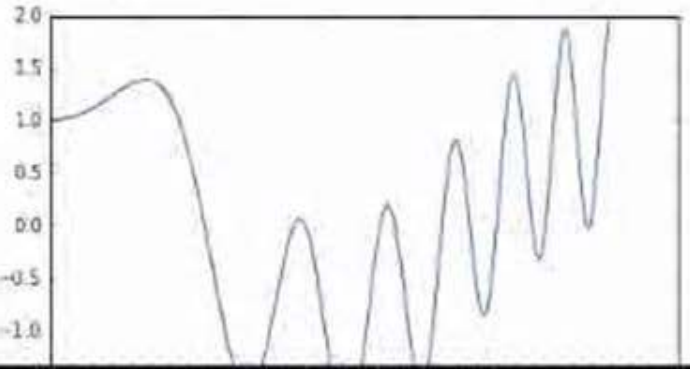
Or Fortran

In [12]:
```
%load_ext fortranmagic
```

In [13]:
```
%%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)+cos(x)
end subroutine f1
```

```
        Building module "_fortran_magic_af73882f78e66bfd353a47b6258c5810"...
                Constructing wrapper function "f1"...
                  y = f1(x,[n])
                Wrote C/API module "_fortran_magic_af73882f78e66bfd353a47b6258c5810" to file "/var/fol
ders/j1/n8xn9ftd7257n2rvkkzlj3mc0010dw/T/tmpb7m8_cup/src.macosx-10.5-x86_64-3.4/_fortran_magi
c_af73882f78e66bfd353a47b6258c5810module.c"
```

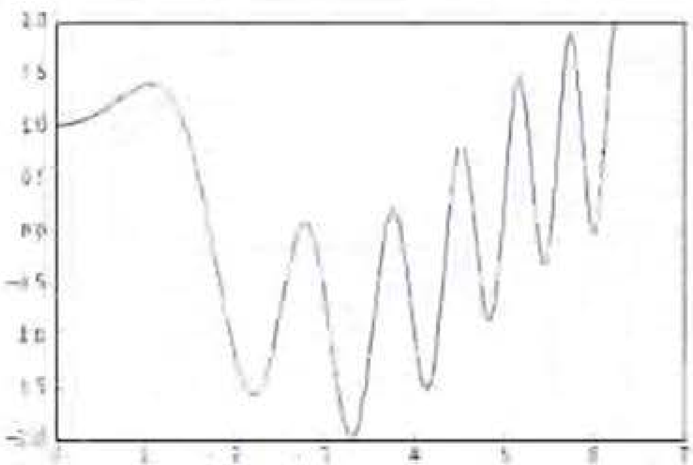
In [14]:
```
t = np.linspace(0,2* np.pi, 1000)
plt.plot(t, f1(t));
```



```
subroutine f1(x, y, n)
    real, intent(in), dimension(n) :: x
    real, intent(out), dimension(n) :: y
    !intent(hide) :: n
    y = sin(x**2)+cos(x)
end subroutine f1
```

```
        Building module "_fortran_magic_af7382f79e66bfd453a47b625ac5810"...
                Constructing wrapper function "f1"...
                  y = f1(x,[n])
        Wrote C/API module "_fortran_magic_af7382f79e66bfd453a47b625ac5810" to file "/var/fo
ders/j./n8gn9ltcl25.n2rvxz13nc00in0w/T/tmpyrvytane/src.macosx-10.5-x86_64-3.4/_fortran_magi
c_af77382f79e66bfd453a47b625ac5810module.c"
```

In [18]:
```
t = np.linspace(0,2*np.pi, 2000)
plt.plot(t, f1(t));
```



In [15]:
```
print(f1.__doc__)
```

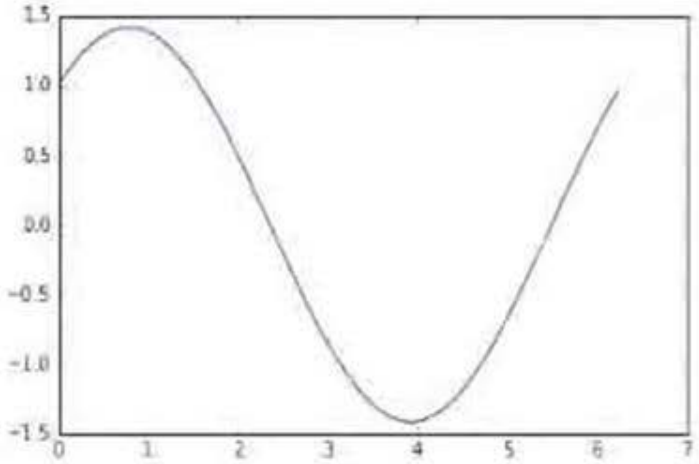
```
y = f1(x,[n])
```

jupyter IPython-Multiple languages from inside IPython Last Checkpoint: an hour ago (unsaved changes)

File Edit View Insert Cell Kernel Help Python

```
                y = f1(x,[n])
        Wrote C/API module "_fortran_magic_c369c8f26399dc4ddbc5795104d05694" to file "/var/fol
ders/j1/n8kn9ftd7257n2rvkkzlj3mc0010dw/T/tmp9gxp6k9y/src.macosx-10.5-x86_64-3.4/_fortran_magi
c_c369c8f26399dc4ddbc5795104d05694module.c"
```

In [20]:
```
t = np.linspace(0,2* np.pi, 1000)
plt.plot(t, f1(t));
```



In [15]:
```
print(f1.__doc__)
```

```
y = f1(x,[n])

Wrapper for ``f1``.

Parameters
----------
x : input rank-1 array('f') with bounds (n)

Other Parameters
----------------
n : input int, optional
```

Interactive Widgets in IJulia

Credit: Steven Johnson @ MIT.

IPython 2.0 introduced interactive widgets, which are basically:

- Javascript widgets (sliders, buttons, etcetera)
- A communications protocol for the widgets to talk to the kernel
- A Python interface to create and manipulate these.

Thanks to fantastic work by a Google Summer of Code student, Shashi Gowda, the same features are accessible from a Julia interface.

In [1]:
```
using Interact
```

In [2]:
```
@manipulate for n in 1:100
    rand(n,n)
end
```

n 6

Out[2]:
```
6x6 Array{Float64,2}:
 0.430157    0.197909   0.580212  0.109846  0.563349  0.409784
 0.481589    0.0392629  0.487369  0.216748  0.687736  0.947904
 0.377111    0.865779   0.64677   0.116375  0.634752  0.225825
 0.00827299  0.219567   0.3663    0.186471  0.22006   0.0547533
 0.087672    0.288259   0.885178  0.984019  0.84279   0.660529
```

jupyter Julia-Interactive Widgets Last Checkpoint: an hour ago (unsaved changes)

File Edit View Insert Cell Kernel Help

- Javascript widgets (sliders, buttons, etcetera)
- A communications protocol for the widgets to talk to the kernel
- A Python interface to create and manipulate these.

Thanks to fantastic work by a Google Summer of Code student, Shashi Gowda, the same features are accessible from a Julia interface.

In [7]:
```
using Interact
```

In [2]:
```
@manipulate for n in 1:100
    rand(n,n)
end
```

n 6

Out[2]:
```
6x6 Array{Float64,2}:
 0.430157    0.197909   0.580212  0.109846  0.563349  0.409784
 0.481589    0.0392629  0.487369  0.216748  0.687736  0.947904
 0.377111    0.865779   0.64677   0.116375  0.634752  0.225825
 0.00827299  0.219567   0.3663    0.186471  0.22006   0.0587533
 0.087672    0.288259   0.885178  0.984019  0.84279   0.660529
 0.398421    0.672923   0.849328  0.895379  0.802594  0.690522
```

In [3]:
```
using Color
@manipulate for r in 0:0.1:1, g in 0:0.1:1, b in 0:0.1:1, n in 1:70
    linspace(RGB(0,0,0), RGB(r,g,b), n)
end
```

r 0.5

g 0.5

b 0.5

Jupyter Julia-Interactive Widgets Last Checkpoint: an hour ago (unsaved changes)

- Javascript widgets (sliders, buttons, etcetera)
- A communications protocol for the widgets to talk to the kernel
- A Python interface to create and manipulate these.

Thanks to fantastic work by a Google Summer of Code student, Shashi Gowda, the same features are accessible from a Julia interface.

In [1]:
```
using Interact
```

In [2]:
```
@manipulate for n in 1:100
    rand(n,n)
end
```

n 19

Out[2]:
```
19x19 Array{Float64,2}:
 0.0382291  0.284352  0.984159  0.296549  …  0.99883   0.142576   0.78328
 0.553628   0.284593  0.05063   0.322332     0.399736  0.126389   0.48847
 0.28988    0.379791  0.375536  0.682090     0.299502  0.415346   0.0853446
 0.457808   0.689981  0.862578  0.26911      0.966819  0.521435   0.198432
 0.932538   0.20118   0.252168  0.165199     0.530221  0.295752   0.163636
 0.284145   0.262136  0.268621  0.712035     0.521256  0.217737   0.255430
 0.452701   0.741583  0.616511  0.686672     0.487905  0.202756   0.842251
 0.311021   0.355519  0.666759  0.765963     0.259374  0.763080   0.862784
 0.413373   0.0151    0.806950  0.328837     0.596733  0.2867256  0.0853448
 0.497386   0.612865  0.295213  0.484224     0.506015  0.004201   0.807491
 0.667906   0.342444  0.917865  0.802698  …  0.853389  0.716081   0.984235
```

In [3]:
```
using Color
@manipulate for r in 0:0.1:1, g in 0:0.1:1, b in 0:0.1:1, n in 1:50
    linspace(RGB(0,0,0), RGB(r,g,b), n)
end
```

127.0.0.1:8888/notebooks/talks/slides/siam/Julia-Interactive Widgets.ipynb

jupyter Julia-Interactive Widgets Last Checkpoint: an hour ago (unsaved changes)

File Edit View Insert Cell Kernel Help | Julia 0.3.6

n 1

Out[2]:
```
1x1 Array{Float64,2}:
 0.580781
```

In [3]:
```
using Color
@manipulate for r in 0:0.1:1, g in 0:0.1:1, b in 0:0.1:1, n in 1:70
    linspace(RGB(0,0,0), RGB(r,g,b), n)
end
```

r 0.5
g 1
b 0.5
n 35

Out[3]:

In [4]:
```
using PyPlot
```
```
INFO: Loading help data...
```

In [5]:
```
x = linspace(0,10,1000)
f = figure()
@manipulate for α = 1:0.1:4, β = 1:0.1:4, leg="a funny plot"
    withfig(f) do
```

NATURE | TOOLBOX

Interactive notebooks: Sharing the code

The free IPython notebook makes data analysis easier to record, understand and reproduce.

Helen Shen

05 November 2014

Notebook-based technical blogs

Jake VanderPlas (astronomy/ML at UW)

```
In [4]: website('http://blogs.scientificamerican.com/sa-visual/2014/09/16/visualizing-
```

Out[4]:

SCIENTIFIC AMERICAN™

Sign In | Register

Search ScientificAmerican.com

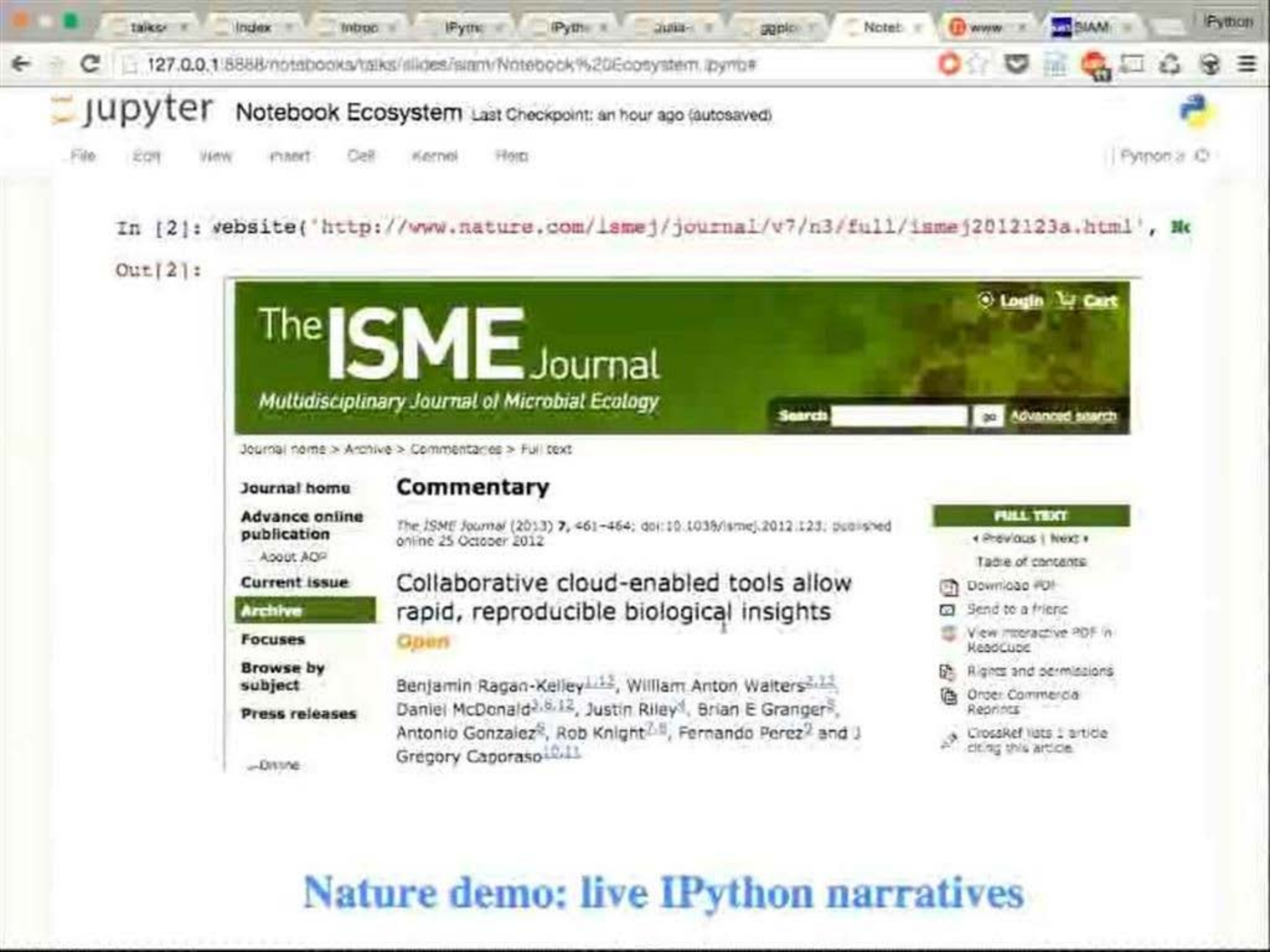
Subscription Cent

Subscribe to All Ac

Subscribe to Print

Give a Gift »

View the Latest Iss


jupyter Notebook Ecosystem Last Checkpoint: an hour ago (autosaved)
In [2]: website('http://www.nature.com/ismej/journal/v7/n3/full/ismej2012123a.html',
Out[2]:
The ISME Journal
Multidisciplinary Journal of Microbial Ecology
Login
Cart
Search
go
Advanced search
Journal home > Archive > Commentaries > Full text
Journal home
Advance online publication
About AOP
Current issue
Archive
Focuses
Browse by subject
Press releases
Commentary
The ISME Journal (2013) 7, 461–464; doi:10.1038/ismej.2012.123; published online 25 October 2012
Collaborative cloud-enabled tools allow rapid, reproducible biological insights
Open
Benjamin Ragan-Kelley, William Anton Walters, Daniel McDonald, Justin Riley, Brian E Granger, Antonio Gonzalez, Rob Knight, Fernando Perez and J Gregory Caporaso
FULL TEXT
« Previous | Next »
Table of contents
Download PDF
Send to a friend
View interactive PDF in ReadCube
Rights and permissions
Order Commercial Reprints
CrossRef lists 1 article citing this article
Nature demo: live IPython narratives

IPython interactive demo

This demonstration is hosted by Rackspace Developer+.
Click here to make the notebook full screen.

IP[y]: Notebook Nature

IPython (Python 3)

File Edit View Insert Cell Kernel Help

nature rackspace

Introduction

Welcome! You have just launched a live example of an IPython Notebook. The notebook is an open-source, interactive computing environment that lets you combine live code, narrative text, mathematics, plots and rich media in one document. Notebook documents provide a complete reproducible record of a computation and its results and can

E-alert RSS Facebook Twitter

back to article

IPython interactive demo

This demonstration is hosted by Rackspace Developer+.
Click here to make the notebook full screen

IP[y]: Notebook Nature

IPython (Python 3)

File Edit View Insert Cell Kernel Help

Introduction

Welcome! You have just launched a live example of an IPython Notebook. The notebook is an open-source, interactive computing environment that lets you combine live code, narrative text, mathematics, plots and rich media in one document. Notebook documents provide a complete reproducible record of a computation and its results and can be shared with collaagues (through, for example, email, web-hosting services such as GitHub, Dropbox, and nbviewer).

You can edit anything in this temporary demonstration notebook, including the text you are reading. To see it full-screen, click on the 'Expand' icon in the lower right corner of the frame around this notebook.

This notebook showcases some of IPython's capabilities for researchers.

This demonstration is hosted by Rackspace and is running on its bare metal offering, OnMetal. Try out these cloud services yourself through Rackspace's developer+ page.

Top story

Scientists sound alarm over DNA editing of human embryos

Experts call for halt in research to work out safety and ethics issues

Recent Read Commented Emailed

1. Flatworms: Eyes on the ocean
   Nature | 17 March 2015
2. Irish government under fire for slashing its back on basin research
   Nature | 17 March 2015
3. Crunch time for Canada's role in mega-telescope
   Nature | 17 March 2015
4. Marijuana gears up for production high in US labs
   Nature | 17 March 2015
5. Reprieve for Australian research facilities
   Nature | 17 March 2015

back to article

IPython interactive demo

This demonstration is hosted by Rackspace Developer.
Click here to make the notebook full screen.

IP[y]: Notebook Nature

(▶) in the toolbar above

A full tutorial for using the notebook interface is available here

```
In [1]: %pylab inline
        Welcome to pylab, a matplotlib-based Python environment [backend: module://IPython.zmq.pylab.backend_inline].
        For more information, type 'help(pylab)'.

        x = np.linspace(0, 5, 10)
        y = x ** 2

        figure()
        plot(x, y, 'r')
        xlabel('x')
        ylabel('y')
        title('simple plot')
        show()
```

E-alert RSS Facebook Twitter

Top story

Scientists sound alarm over DNA editing of human embryos

Experts call for halt in research to work out safety and ethics issues.

Read Commented Emailed

1. **Fisheries: Eyes on the ocean**
   Nature | 17 March 2015
2. **Irish government under fire for turning its back on basic research**
   Nature | 17 March 2015
3. **Crunch time for Canada's role in mega-telescope**
   Nature | 17 March 2015
4. **Marijuana gears up for production high in US labs**
   Nature | 17 March 2015
5. **Reprieve for Australian research facilities**
   Nature | 17 March 2015

back to article

IPython interactive demo

This demonstration is hosted by Rackspace Developer.
Click here to make the notebook full screen.

IP[y]: Notebook Nature

Python (Pylmm 3)

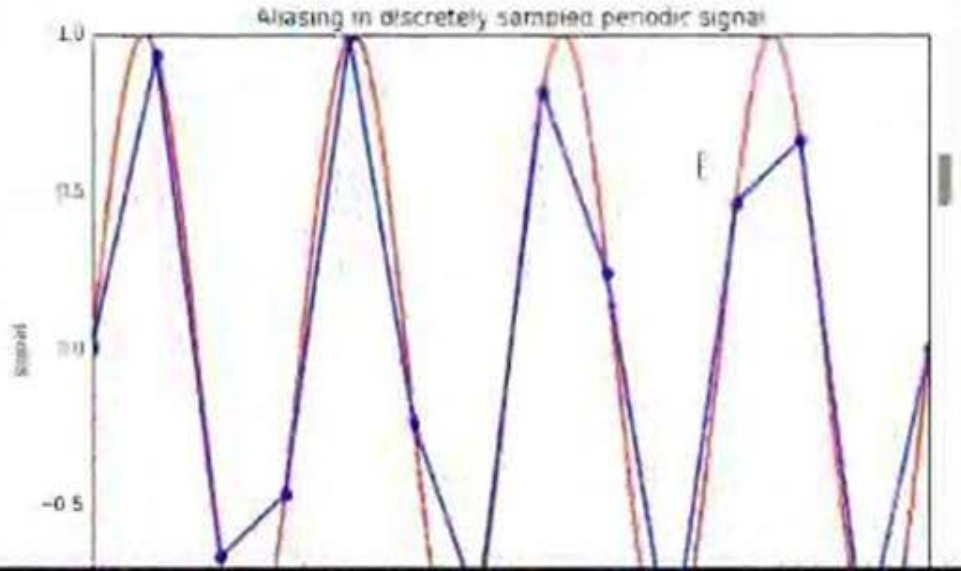
File Edit View Insert Cell Kernel Help

```
interact(plot_sine, frequency=(1.0, 22.0, 0.5), grid_po
```

frequency 4

grid_points 12

plot_origins ☑



jupyter Notebook Ecosystem Last Checkpoint: an hour ago (autosaved)

File Edit View Insert Cell Kernel Help Python 3

The scientific paper of the future:

In [3]: `website('http://www.nature.com/news/interactive-notebooks-sharing-the-code-1.`

Out[3]:

NATURE | TOOLBOX

Interactive notebooks: Sharing the code

The free IPython notebook makes data analysis easier to record, understand and reproduce.

Helen Shen

05 November 2014

PDF Rights & Permissions

Notebook-based technical blogs

jupyter Notebook Ecosystem Last Checkpoint: an hour ago (autosaved)

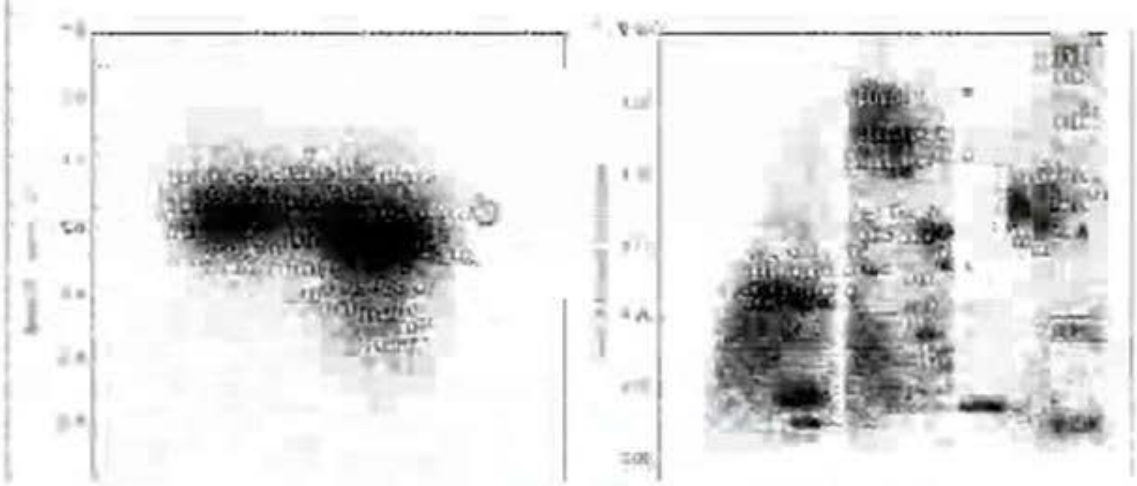
File Edit View Insert Cell Kernel Help

In [4]: website('http://blogs.scientificamerican.com/sa-visual/2014/09/16/visualising

Out[4]:

plot, perhaps resulting on a compromise? But how do you effectively plot four dimensional data?

Will this start simple, by splitting the data into two/six individuals on one chart and ordinal indicators on the other

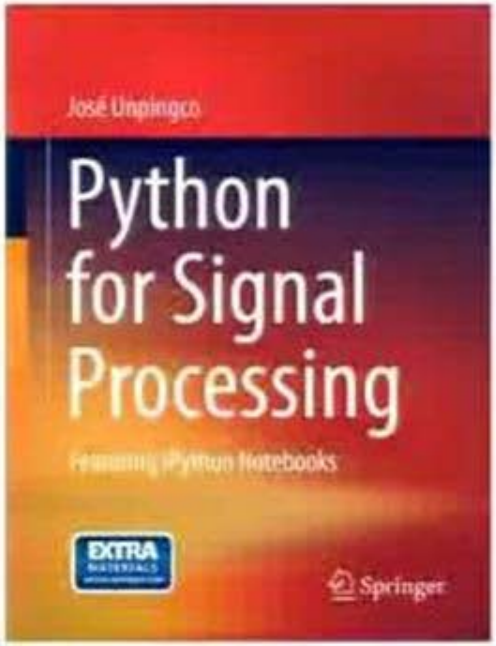


Books: "Literate Computing"

Books: "Literate Computing"

Python for Signal Processing, By Jose Unpingco

- Springer hardcover book
- Each chapter is an IPython Notebook
- Posted as a blog entry
- And all available as a Github repo



Education

9	Introduction to Programming	Alaskan High School	Eric Matthes
10	Aerodynamics-Hydrodynamics (MAE 6226)	George Washington University	Lorena Barba
11	HyperPython: hyperbolic conservation laws	KAUST	David Ketcheson
12	Quantitative Economics	NYU	Sargent and Stachurski
13	Practical Numerical Methods with Python	4 separate universities + MOOC	Barba, et al.
14	Data Science: Algorithms	Columbia - Lede Program	Chris Wiggins
15	Data Science: Databases	Columbia - Lede Program	Chris Wiggins
16	Data Science: Foundations	Columbia - Lede Program	Chris Wiggins
17	Data Science: Platforms	Columbia - Lede Program	Chris Wiggins

NBViewer: Share notebooks online


127.0.0.1:8888/notebooks/talks/slides/siam/Notebook%20Ecosystem.ipynb#
jupyter Notebook Ecosystem Last Checkpoint: an hour ago (autosaved)
File Edit View Insert Cell Kernel Help
Python 3
Matthias Bussonnier, 2012
In [7]: website('nbviewer.ipython.org', None)
Out[7]:
nbviewer
nbviewer
A simple way to share Jupyter Notebooks
URL | GitHub username | GitHub username/repo | Gist ID
Go!
Programming Languages
IPython

This is an IPython notebook backed by a Ruby kernel

I am developing IRuby a kernel in Ruby that adheres to the IPython messaging protocol. It integrates nicely with different Ruby gems as will be shown later.

What does this give you?

This gives us a very fancy web notebook interface for Ruby. It's a very good tool for programming presentations. It's basically an in-browser REPL loop, with some extra goodies.

Usage

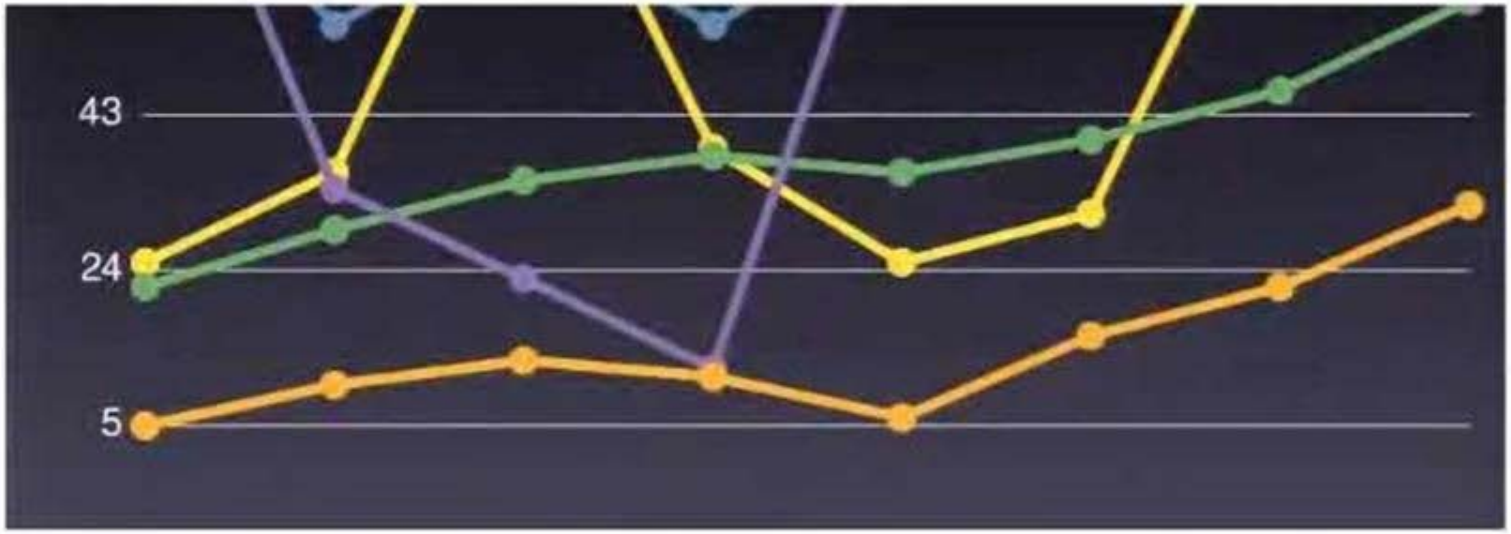
Install IRuby with:

```
gem install iruby
```

Start the IRuby notebook with:

```
iruby notebook
```

Example



Matrix & GSL

`Matrix` and `GSL::Matrix` objects are automatically displayed as $\LaTeX$.

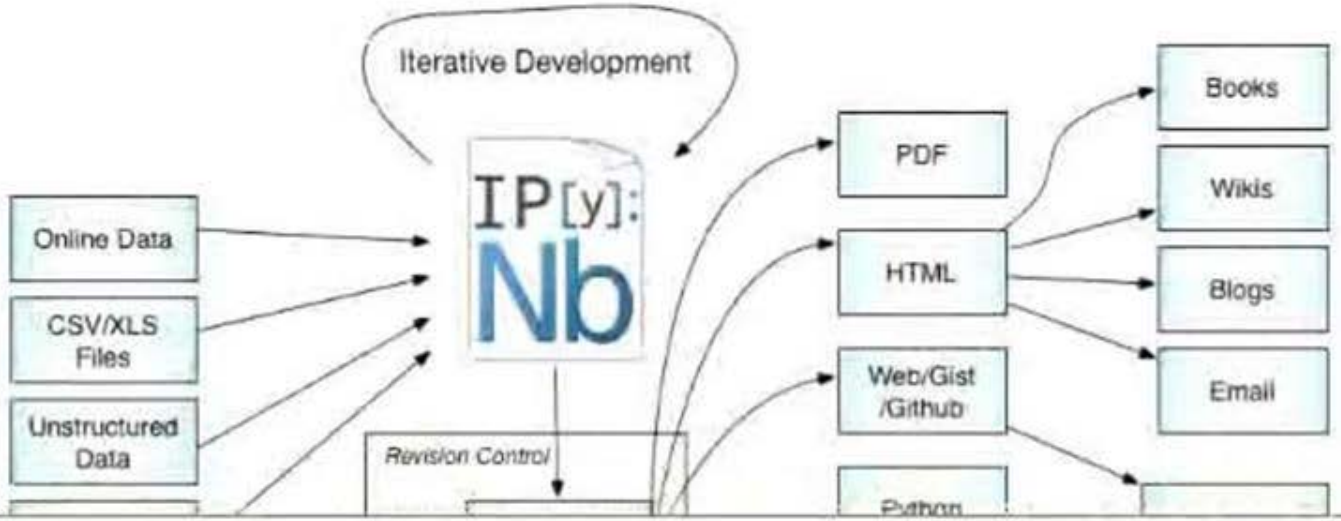
In [16]:
```
require 'matrix'
Matrix[[1,2,3],[1,2,3]]
```

Out[16]:
$$\begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}$$

Back to top

jupyter Notebook Ecosystem Last Checkpoint: an hour ago (autosaved)

Notebook Workflows: The Big Picture

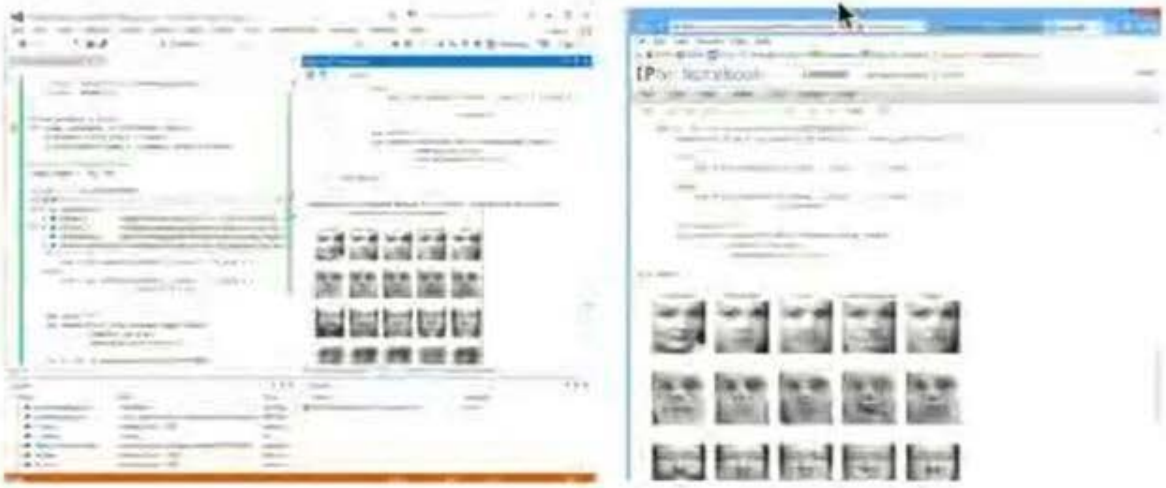


Others building atop IPython

OSS and commercial players

Microsoft Python Tools for Visual Studio

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



Microsoft Azure (S. Mortazavi, W. Ye)

Case study - Question Class Analysis Using IPython Notebook

IBM

- Effort:

Feature	LOC	Language
IPython analysis notebook	20	Python
Extract relevant experiment results (module "result_proc")	200	Python
Total:	220	1

- Setup:
  - IPython Notebook installed in Python virtual environment
    - Packages: Pandas, NumPy, SymPy, Matplotlib, scikit-learn
  - Deployed nbserver for each team member in cluster environment
  - Shared file system for direct access to experiment results
  - Workers deployed across 10 machines for parallel processing

Authorea: collaborative academic writing

- IPython Notebook installed in Python virtual environment
  - Packages: Pandas, NumPy, SymPy, Matplotlib, scikit-learn
- Deployed nbserver for each team member in cluster environment
- Shared file system for direct access to experiment results
- Workers deployed across 10 machines for parallel processing

Authorea: collaborative academic writing

```
In [10]: website('https://www.authorea.com/users/3/articles/3904/_show_article', None)
```

Out[10]:

> For example, if you are a scientist, chances are that you do a lot of data analysis and you might want to visualize and provide access to your data in some **fun, new, interactive,**

Much, much more

A Gallery of Interesting Notebooks ¶

Try Jupyter yourself!

try.jupyter.org

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

```
Populating the interactive namespace from numpy and matplotlib
```

1. Entire books or other large collections of notebooks on a topic
   - Introductory Tutorials
   - Programming and Computer Science
   - Statistics, Machine Learning and Data Science
   - Mathematics, Physics, Chemistry, Biology
   - Earth Science and Geo-Spatial data
   - Linguistics and Text Mining
   - Signal Processing
2. Scientific computing and data analysis with the SciPy Stack
   - General topics in scientific computing
   - Social data
   - Psychology and Neuroscience
   - Machine Learning
   - Physics, Chemistry and Biology
   - Economics
   - Earth science and geo-spatial data
   - Data visualization and plotting
   - Mathematics
   - Signal and Sound Processing
   - Natural Language Processing
   - Pandas for data analysis
3. General Python Programming
4. Notebooks in languages other than Python
   - Julia
   - Haskell
   - Ruby

Cookbook: Ignoring some commands in history

Cookbook: Index

Cookbook: IPython and Emacs

Cookbook: IPython and Python.NET

Cookbook: Job control extension

Cookbook: Launching IPython on OSX

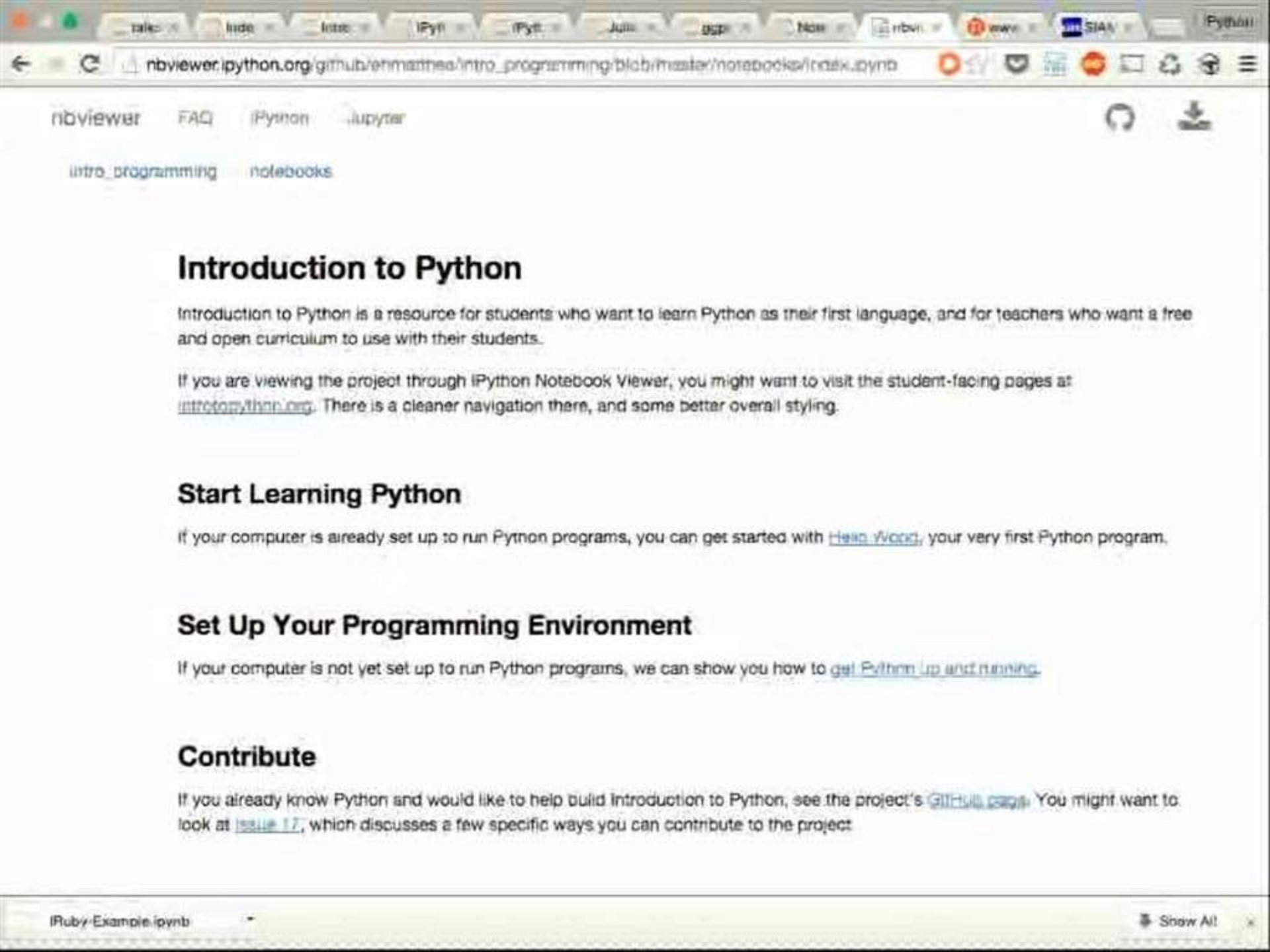
Show 123 more pages

Add a custom sidebar

Clone this wiki locally

https://github.com/ipython/ipython.

Clone in Desktop


nbviewer.ipython.org/github/ehmatthes/intro_programming/blob/master/notebooks/index.ipynb
nbviewer
FAQ
IPython
Jupyter
intro_programming
notebooks
Introduction to Python
Introduction to Python is a resource for students who want to learn Python as their first language, and for teachers who want a free and open curriculum to use with their students.
If you are viewing the project through IPython Notebook Viewer, you might want to visit the student-facing pages at introtopython.org. There is a cleaner navigation there, and some better overall styling.
Start Learning Python
If your computer is already set up to run Python programs, you can get started with Hello World, your very first Python program.
Set Up Your Programming Environment
If your computer is not yet set up to run Python programs, we can show you how to get Python up and running.
Contribute
If you already know Python and would like to help build Introduction to Python, see the project's GitHub page. You might want to look at Issue 17, which discusses a few specific ways you can contribute to the project.
IRuby-Example.ipynb
Show All

jupyter Notebook Ecosystem Last Checkpoint: an hour ago (autosaved)

> For example, if you are a scientist, chances are that you do a lot of data analysis and you might want to visualize and provide access to your data in some **fun, new, interactive,**

Much, much more

A Gallery of Interesting Notebooks

Try Jupyter yourself!

try.jupyter.org

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

```
Populating the interactive namespace from numpy and matplotlib
```