

In [1]:

```
import datetime
print("Last update: " + str(datetime.datetime.now()))
```

Last update: 2016-07-28 21:21:49.986345

In [2]:

```
%%html
<h3>Select Theme</h3><select id="theme-list0"></select>
<script src="js/d3.min.js"></script>
<script src="RISE_themes/RISE_themes.js"></script>
<!-- NOTE: This cell must be code/%%html not markdown for css to be handled -->
<!-- NOTE: table test (class "noborder"): want no borders -->
<style> .noborder, .noborder tr, .noborder th, .noborder td {
    border-collapse: collapse; border: 0px; border: none!important;
} </style>

<table class="noborder"><tbody><tr><td> Cell1 </td><td> Cell2 </td></tr></tbody></table>
<table><tbody><tr><td> BorderedCell1 </td><td> BorderedCell2 </td></tr></tbody></table>
```

## Select Theme

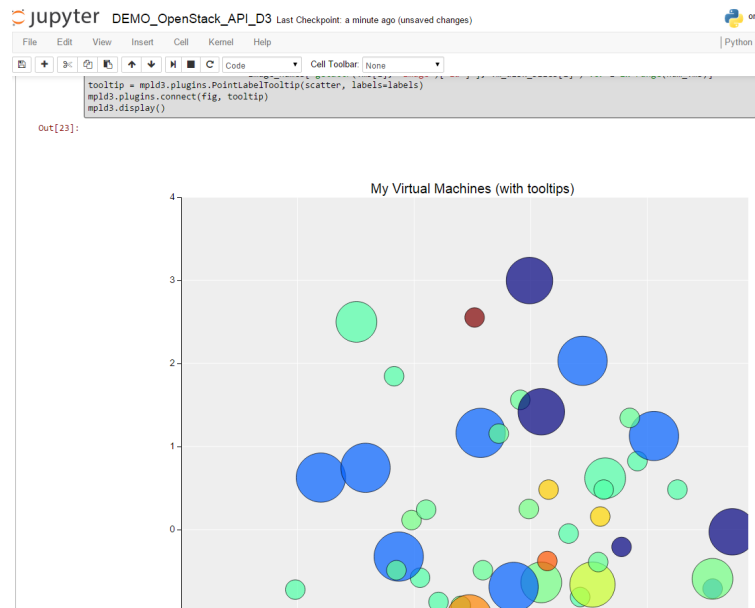
mysky ▼

Cell1 Cell2

|               |               |
|---------------|---------------|
| BorderedCell1 | BorderedCell2 |
|---------------|---------------|

# Jupyter for Everything Else

**Michael Bright, EuroPython 2016 - Bilbao, 22 July**  
**(<https://ep2016.europython.eu/conference/talks/jupyter-for-everything-else>).**



- ☐ [mjbright](<https://github.com/mjbright>)
- ☐ [@mjbright](<http://twitter.com/mjbright>)
  - ☐ mjbrightfr AT gmail
- ☐ [mjbright.github.io](<http://mjbright.github.io>)

## Solution Architect Hewlett-Packard Enterprise

Working in the EMEA OpenNFV lab, Grenoble, France.



## Outline

- Introduction: From IPython to Jupyter
- The Jupyter Project & Ecosystem:
  - Kernels, Widgets, Extensions, Tools
  - Incubating & ext. projects, Hosting
- Jupyter for Everything Else
  - Web, Command-line, Live Blogging
  - Presenting, Automated Status reports

# IP[y]: IPython - the Console

(<http://ipython.org/>)

"\*an afternoon hack\*" (Nov 2001) by Fernando Perez

A tool to help in the exploration process

- Individual exploration
- Collaborative work
- Parallel Production Runs
- Publication of **reproducible** results
- Education
- Repeat

# IP[y]: IPython - the Console

(<http://ipython.org/>)

Initial 0.0.1 version [Gist](https://gist.github.com/fperez/1579699)  
(<https://gist.github.com/fperez/1579699>)

- REPL in 259 lines
- Input/Output cells
- History
- Plotting

```
Python 3.5.1 |Anaconda 4.0.0 (64-bit)| (default, Feb 1
Type "copyright", "credits" or "license" for more info
```

```
IPython 5.0.0 -- An enhanced Interactive Python.
?          -> Introduction and overview of IPython's fe
%quickref -> Quick reference.
help       -> Python's own help system.
object?    -> Details about 'object', use 'object??' fo
```

```
In [1]: myvar=1
```

```
In [2]: my
```

```
"My Documents"
myvar
```

</center>

IP[y]:

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

## IPython - the Console

Python 3.5.1 |Anaconda 4.0.0 (64-bit)| (default, Feb 16 2016, 09:4  
Type "copyright", "credits" or "license" for more information.

IPython 5.0.0 -- 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]: myvar=1

In [2]: m

|        |             |          |        |
|--------|-------------|----------|--------|
| %macro | %matplotlib | min      | MOOCs/ |
| %magic | max         | %mkdir   | myvar  |
| map    | memoryview  | modules/ |        |

In [3]: myvar?

Type: int

String form: 1

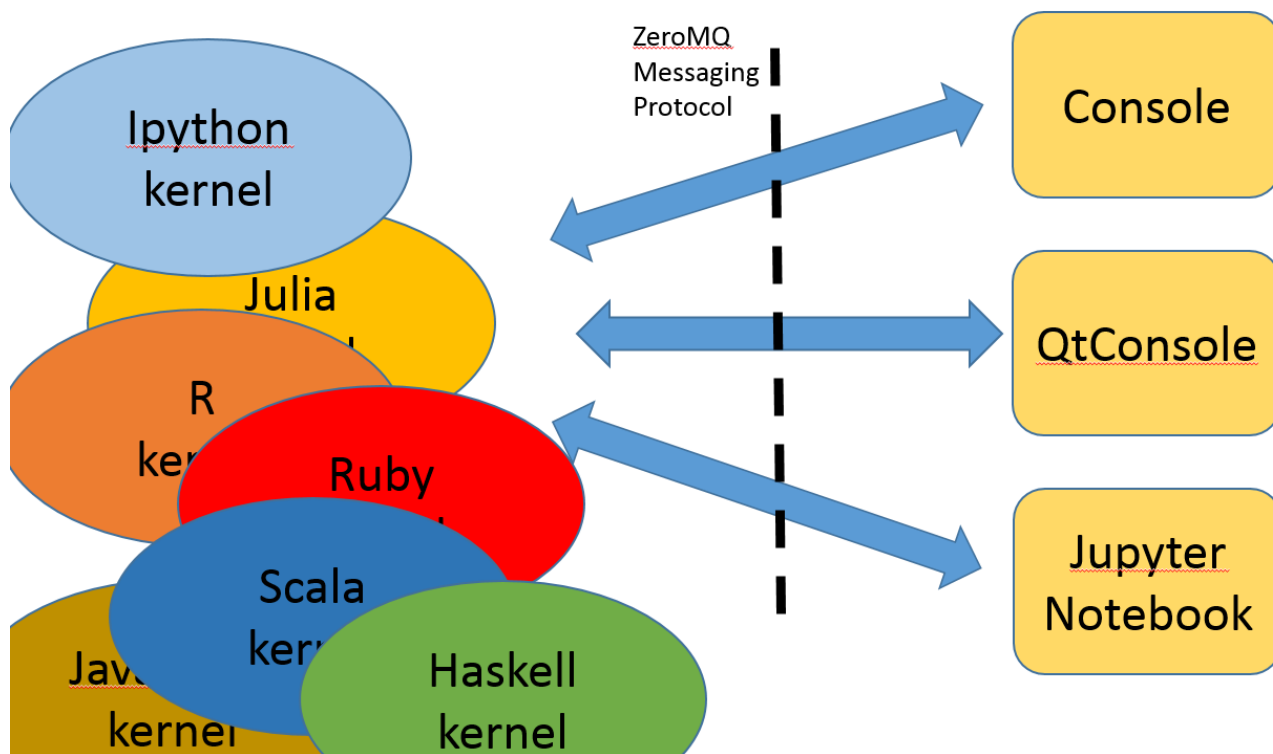
Docstring:

int(x=0) -> integer

int(x, base=10) -> integer

Convert a number or string to an integer, or return 0 if no arguments are given. If x is a number, return x.\_\_int\_\_(). For floating point numbers, this truncates towards zero.

## The Jupyter Project



A notebook runs under one kernel

# The (Jupyter) Dashboard & [Notebook]

([http://localhost:8888/notebooks/Demos/Demo\\_0\\_IN](http://localhost:8888/notebooks/Demos/Demo_0_IN))

Create & share documents of code, equations, visualizations and explanatory text as a  
(reproducible) narrative

&lt;/center&gt;

The screenshot shows the Jupyter Notebook interface. At the top, the Jupyter logo and 'Rich Output (autosaved)' are visible. Below is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', and 'Help'. A red arrow points to the 'Menus' label. To the right, it says 'Running Kernel' and 'Python 3'. Below the menu bar is a toolbar with various icons for file operations, editing, and viewing. A red arrow points to the 'Toolbar' label. The main content area contains a code cell with LaTeX equations for Maxwell's equations:

$$\begin{aligned}\nabla \times \vec{B} - \frac{1}{c} \frac{\partial \vec{E}}{\partial t} &= \frac{4\pi}{c} \vec{j} \\ \nabla \cdot \vec{E} &= 4\pi \rho \\ \nabla \times \vec{E} + \frac{1}{c} \frac{\partial \vec{B}}{\partial t} &= \vec{0} \\ \nabla \cdot \vec{B} &= 0\end{aligned}$$

A red arrow points to this code cell with the label 'Markdown'. Below the equations is an 'Audio' section with a speaker icon. The text explains that IPython makes it easy to work with sounds interactively. A red arrow points to this text with the label 'Input Cell'. Below the text is a code cell (In [27]:) that imports IPython.display and Audio, and creates an Audio object from a URL:

```
from IPython.display import Audio
Audio(url="http://www.nch.com.au/acm/8k16bitpcm.wav")
```

Below the code cell is an output cell (Out[27]:) showing an audio player with a play button and a progress bar. A red arrow points to this output cell with the label 'Output Cell'. Below the audio player is another text block explaining that a NumPy array can be auralized automatically. A red arrow points to this text with the label 'Output Cell'. Below the text is a code cell (In [28]:) that imports numpy and sets up parameters for auralization:

```
import numpy as np
max_time = 3
f1 = 220.0
f2 = 224.0
rate = 8000.0
```

# Jupyter: Public Notebooks

Jupyter notebooks are used in many domains

- scientific (physics, chemistry, biology, genomics, data analysis)
- and non-scientific (finance)

| Site             | URL  | Info   |
|------------------|--|--|
| nbviewer         | <a href="https://nbviewer.org">https://nbviewer.org</a> ( <a href="https://nbviewer.org">https://nbviewer.org</a> )  | submit your url, browse by theme   |
| github           | <a href="https://github.com">https://github.com</a> ( <a href="https://github.com">https://github.com</a> )  | > 200k notebooks [ <a href="http://blog.jupyter.org/2015/05/07/rendering-notebooks-on-github/">Announcement - May '15</a> ]<br>( <a href="http://blog.jupyter.org/2015/05/07/rendering-notebooks-on-github/">http://blog.jupyter.org/2015/05/07/rendering-notebooks-on-github/</a> ) |
| IPython gallery  | <a href="https://github.com/ipython/ipython/wiki/A-gallery-of-interesting-IPython-Notebooks">A-gallery-of-interesting-IPython-Notebooks</a><br>( <a href="https://github.com/ipython/ipython/wiki/A-gallery-of-interesting-IPython-Notebooks">https://github.com/ipython/ipython/wiki/A-gallery-of-interesting-IPython-Notebooks</a> ) | many notebooks organized by domain   |
| Notebook Gallery | <a href="http://nb.bianp.net/">http://nb.bianp.net/</a><br>( <a href="http://nb.bianp.net/">http://nb.bianp.net/</a> )   | view submitted notebooks by ' <i>most viewed</i> ' or ' <i>data</i> '  |

## Jupyter: Running notebooks

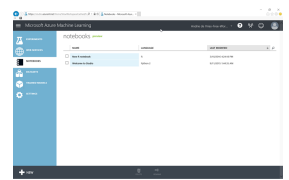
- Native OS Python distribution + Pip, or Anaconda
- JupyterHub, multi-user server
- Under Docker [e.g. *docker-stacks* images]
- Integrated into data science Cloud Hosting or plain IaaS:
  - [Azure ML Studio](https://blogs.technet.microsoft.com/machinelearning/2015/07/24/introducing-jupyter-notebooks-in-azure-ml-studio/)  
(<https://blogs.technet.microsoft.com/machinelearning/2015/07/24/introducing-jupyter-notebooks-in-azure-ml-studio/>)
  - [Google Cloud DataLab Beta](https://cloud.google.com/datalab/) (<https://cloud.google.com/datalab/>)
  - [IBM Data Scientist Workbench](https://my.datascientistworkbench.com/) (<https://my.datascientistworkbench.com/>)
- **Cloud hosted (ephemeral)**
  - [tryjupyter.org](https://try.jupyter.org/) (<https://try.jupyter.org/>) [uses *docker-demo* image]
  - **Binder** (<https://mybinder.org> (<https://mybinder.org>))



# Jupyter & Azure ML Studio

Jupyter integration in Azure ML Studio

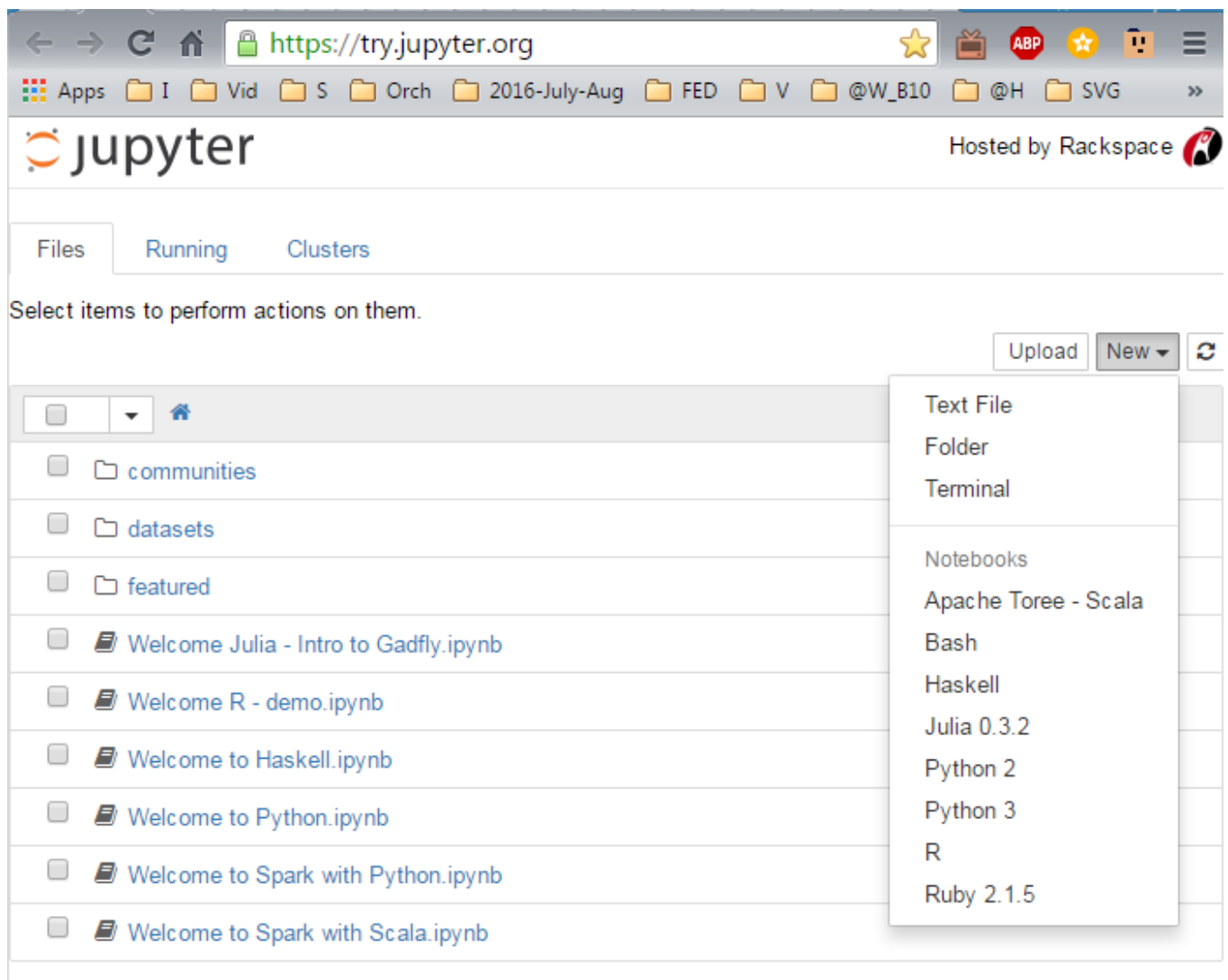
(<https://blogs.technet.microsoft.com/machinelearning/2015/07/24/introducing-jupyter-notebooks-in-azure-ml-studio/>)



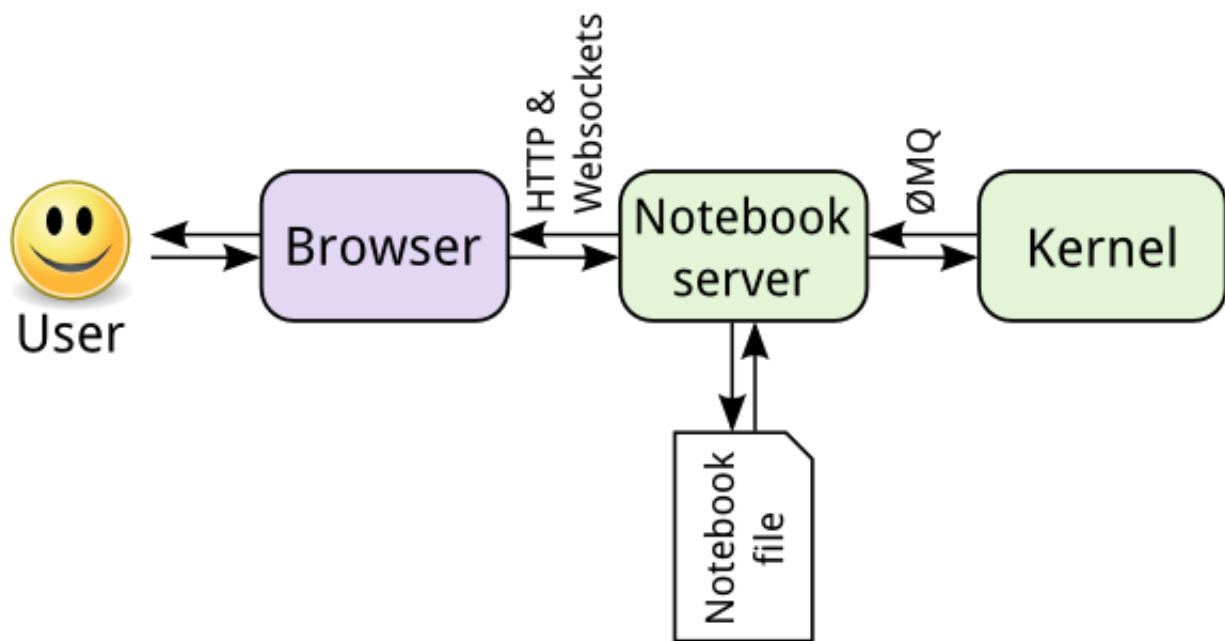
- R

- Python

## try.jupyter.org



# Kernels, Widgets & Extensions



## ~ 50 Kernels

Kernels are execution environments - typically a language [[Kernels Page]]  
 (<https://github.com/ipython/ipython/wiki/IPython-kernels-for-other-languages>)

|                  |                     |                        |                  |                            |
|------------------|---------------------|------------------------|------------------|----------------------------|
| <b>**Julia**</b> | <b>**IRKernel**</b> | <b>**IPython**</b>     | IRuby            | IGo                        |
| IScilab          | IMatlab             | Hy                     | Clojure          | <b>**Bash**</b>            |
| Lua              | Mochi               | IErlang                | <b>**Spark**</b> | <b>**MetaKernel**</b>      |
| IVisual VP       | Brainfuck           | Brython                | IOCaml           | <b>**MetaKernel_Bash**</b> |
| IScala           | IMathics            | IAldor                 | Calico Pro       | Calysto Prolog             |
| IForth           | <b>**IPerl**</b>    | <b>**IPerl6**</b>      | IPHP             | IOctave                    |
| KDB+/Q Ker       | ICryptol            | <b>**C++ (cling)**</b> | <b>**Xonsh**</b> | IJavascript                |
| cl-jupyter       | IHaskell            | IElixir                | Java 9           | Calysto LC                 |
| Redis            | jove                | Prolog                 | IFSharp          | Calysto Scheme             |

# Widgets

Widgets are eventful python objects with a representation in the browser. [[documentation]]  
 (https://ipywidgets.readthedocs.io/en/latest/examples/Widget%20Basics.html#What-are-widgets?)

Provided widgets include:

- IntSlider, FloatSlider, FloatProgress
- Buttons, Checkboxes, Radio buttons
- Dropdown menus



## Extensions

- Collection [github]: [ipython-contrib/IPython-notebook-extensions](https://github.com/ipython-contrib/IPython-notebook-extensions)  
 (https://github.com/ipython-contrib/IPython-notebook-extensions)
- Installed to <http://localhost:8888/nbextensions/> (http://localhost:8888/nbextensions/)

Many extensions available, including:

- **RISE** (<https://github.com/damianavila/RISE>) - **these slides are running under Jupyter**
- nbpresenter
- nbgrader (<https://nbgrader.readthedocs.org/en/stable/>) - creation/grading of classroom assignments

Generally installable via pip or from github repo

# nbgrader

Jupyter Problem 1 Last Checkpoint: a few seconds ago (autosaved)

File Edit View Insert Cell Kernel Help Python 3

Cell Toolbar: Create Assignment Total points: 5

Part B (3 points)

Describe the difference between an *arithmetic mean*, a *harmonic mean*, and a *geometric mean*.

Points: 3 ID: describe\_means Manually graded answer

Arithmetic mean:

$$\frac{1}{N} \sum_{i=1}^N x_i$$

Harmonic mean:

$$\left( \frac{1}{N} \sum_{i=1}^N \frac{1}{x_i} \right)^{-1}$$

Geometric mean:

$$\left( \prod_{i=1}^N x_i \right)^{\frac{1}{N}}$$

## The Ecosystem & Future Projects

# Jupyter Incubator Projects

(<https://github.com/jupyter-incubator>) [[proposals](#)](<https://github.com/jupyter-incubator/proposals>)

## [[sparkmagic](#)]

(<https://github.com/jupyter-incubator/sparkmagic>)

Jupyter magics and kernels for working with remote Spark cluster

## [[declarativewidgets](#)]

(<https://github.com/jupyter-incubator/declarativewidgets>)

Declare Widgets in HTML

## [[dashboards](#)]

(<https://github.com/jupyter-incubator/dashboards>)

Create Dashboards from Notebooks

## [[contentmanagement](#)]

(<https://github.com/jupyter-incubator/contentmanagement>)

Extensions for search, notebook modules/[[cookbooks](#)](<https://github.com/jupyter-incubator/contentmanagement/blob/master/etc/notebooks/cookbooks/ToC>), [[bundlers](#)](<https://github.com/jupyter-incubator/contentmanagement/blob/master/etc/notebooks/associated-vid>)(<https://www.youtube.com/watch?v=SJiezXPhVv8>)

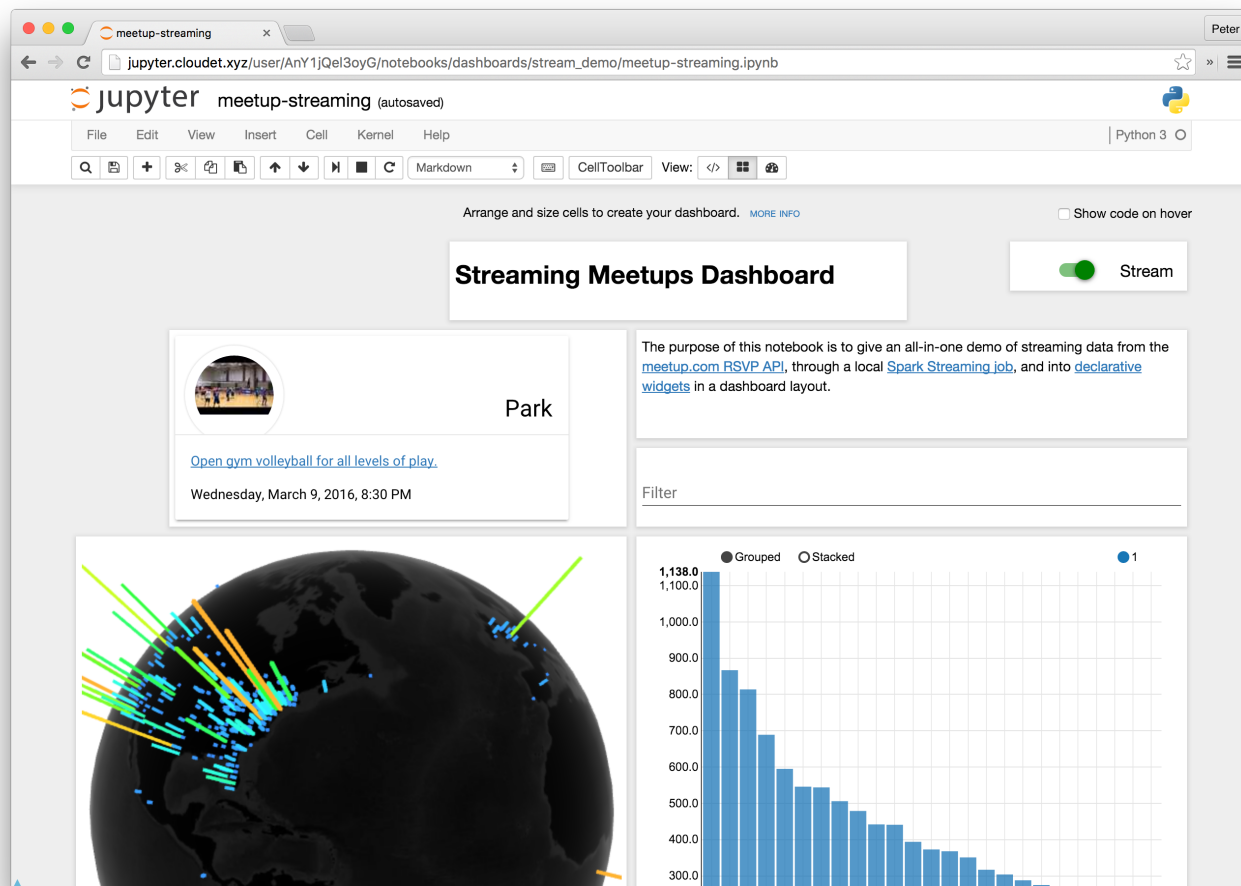
## [[kernel\\_gateway](#)]

([https://github.com/jupyter-incubator/kernel\\_gateway](https://github.com/jupyter-incubator/kernel_gateway))

Support different protocols to Jupyter server, e.g. non-nb web client

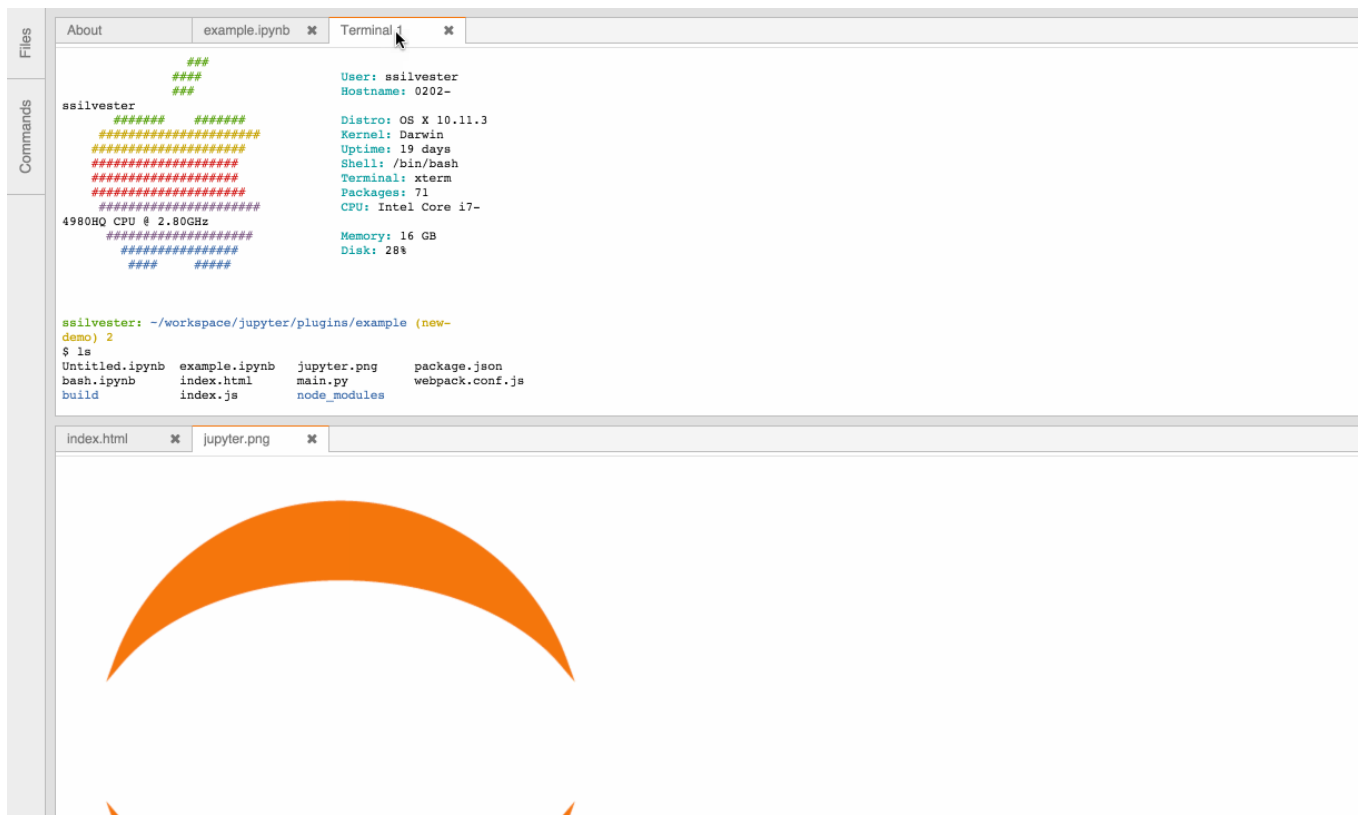
# Incubator: Jupyter Dashboards

(<https://github.com/jupyter-incubator/dashboards>) - alternative layouts



# JupyterLab - pre-Alpha

(<https://github.com/jupyter/jupyterlab/>) - the future interface



## External Jupyter Projects

Many external projects integrating Jupyter:

*Beaker, Hydrogen (ATOM), EIN (Emacs), Rodeo, SageMathCloud.*

Publishers are turning to Jupyter for books, blogs, reports, theses sometimes with live code examples.

- e.g. **Thebe** (O'Reilly) (<https://github.com/oreillymedia/thebe>)
- Nature, Scientific American Magazines

Educators

- tutorials, assignments, presentations, documenting
- MOOCs - online education:
  - notebook-based (**Edx**/**Apache Spark**)
  - jupyter-integrated (**F.U.N.**)
- in classrooms using JupyterHub

# O'Reilly Blog Article - using Thebe

(<https://www.oreilly.com/learning/introducing-pandas-objects>)

Blog post with modifiable, runnable code cells with **\*\*RUN\*\*** button :

Just as the standard alias for importing `numpy` is `np`, the standard alias for importing `pandas` is `pd` :

```
1 import numpy as np
2 import pandas as pd
```

Run

</td></tr></tbody></table></a>

## Jupyter for Everything Else

- Use of web technologies: mix-in HTML, CSS, js, SVG ...
- Use of bash kernel for command-line work
- Supplement command-line tools with graphics
- Create interactive presentations (thanks *RISE* extension !)
- Publish "live blog posts"
- Creating status reports from notebooks using nbconvert

## Everything Else: web technologies

- HTML/JavaScript/css experimentation
  - HTML, CSS, JS capabilities but **proceed with care**
  - d3.js animations if
    - Need more interactivity
    - Prototyping a D3 project
    - Reusing existing D3 e.g. from <http://bl.ocks.org> (<http://bl.ocks.org>)
- SVG

Select Theme:

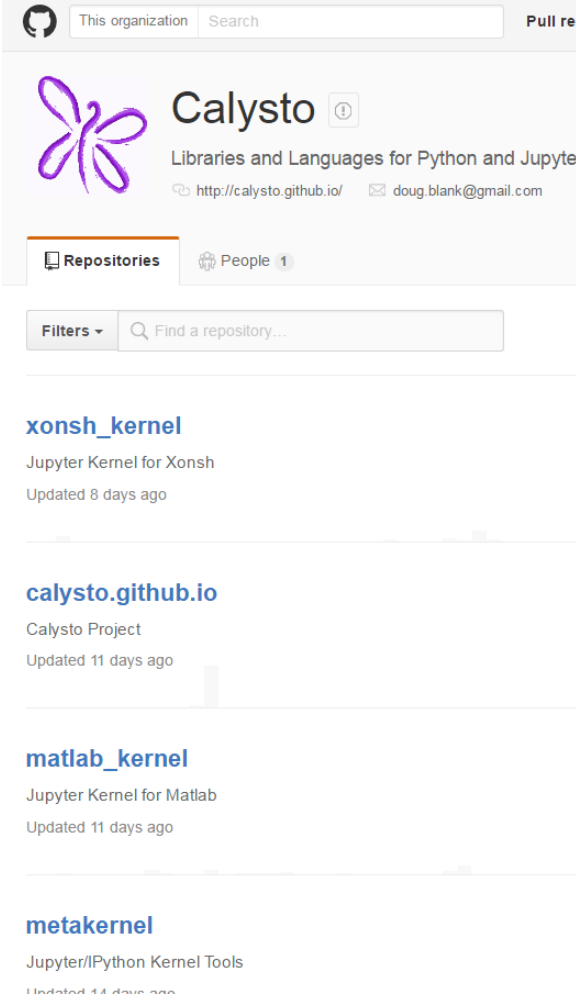


# Everything Else: command-line

Two bash kernels exist for Jupyter: `bash_kernel` and `calysto/metakernel_base`

## [Calysto](https://github.com/Calysto) Metakernels

- easy to fix for Windows/Cygwin
- Family of metakernels
- - Common magics
- Active development



The screenshot shows the GitHub organization page for Calysto. At the top, there's a header with the Calysto logo (a purple flower-like shape) and the name "Calysto". Below the logo, it says "Libraries and Languages for Python and Jupyter". There are links to "http://calysto.github.io/" and "doug.blank@gmail.com". Below this, there are tabs for "Repositories" and "People". The "Repositories" tab is selected, showing a list of repositories. The first repository is "xonsh\_kernel", described as "Jupyter Kernel for Xonsh" and "Updated 8 days ago". The second is "calysto.github.io", described as "Calysto Project" and "Updated 11 days ago". The third is "matlab\_kernel", described as "Jupyter Kernel for Matlab" and "Updated 11 days ago". The fourth is "metakernel", described as "Jupyter/IPython Kernel Tools" and "Updated 11 days ago".

# Everything Else: command-line

## Why? ... I hear you ask

Inspired by the notebook as an educational tool, I used it for CLI-b

- [\*\*Docker demos / labs \*\*]  
(<http://mjbright.blogspot.fr/2016/02/creating-docker-build-lab-with-jupyter.html>)
- Bash: 1st-class citizen with magics+
- [\*\*Example notebook\*\*]  
([Demos/Demo\\_Metakernel\\_Bash.ipynb](#))
- CLI tutorials as runnable  
"live notebook".

### **Adding HTML o/p capabilities t**

I've provided an `'html'` function which enables display of HTML

Below examples of pretty printing HTML from various sources

In [1]:

```
echo "<h1>An html header <!-- A comment --></h1>"
```

### **An html header**

In [2]:

```
echo "<h1>A table <!-- A comment --></h1>
<table><tr><th>col1</th>
<th>col2</th>
<th>col3</th>
<th>col4</th>
</tr>
<tr><td>val1</td>
<td>val2</td>
<td>val3</td>
<td>val4</td>
</tr>
<tr><td>val1</td>
<td>val2</td>
<td>val3</td>
<td>val4</td>
</tr></table>" | html
```

### **A table**

| col1 | col2 | col3 | col4 |
|------|------|------|------|
| val1 | val2 | val3 | val4 |
| val1 | val2 | val3 | val4 |

# Everything Else: Binder "live notebook"



## Turn a GitHub repo into a collection of interactive notebooks

Have a repository full of Jupyter notebooks? With Binder, you can add a badge that opens those notebooks in an executable environment, making your code immediately reproducible by anyone, anywhere.

100% free and [open source](#). Browse [examples](#). Read the [FAQ](#).

### Build a repository

## How it works

1

In the field above, enter a GitHub repository that contains Jupyter notebooks, and click Submit to start the build. All files will be included, and if there's an index.ipynb notebook it will load it. Check out an [example](#).



# Everything Else: "live tutorials"

[https://github.com/mjbright/binder\\_jupyter\\_notebook](https://github.com/mjbright/binder_jupyter_notebook)  
([https://github.com/mjbright/binder\\_jupyter\\_notebook](https://github.com/mjbright/binder_jupyter_notebook))

## Everything Else: Binder CLI "live tutorial"

Create live tutorials online on Binder.

Notebook server is launched by clicking on the binder icon



in a github repo

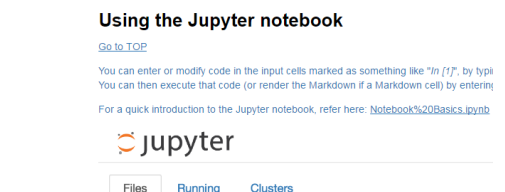
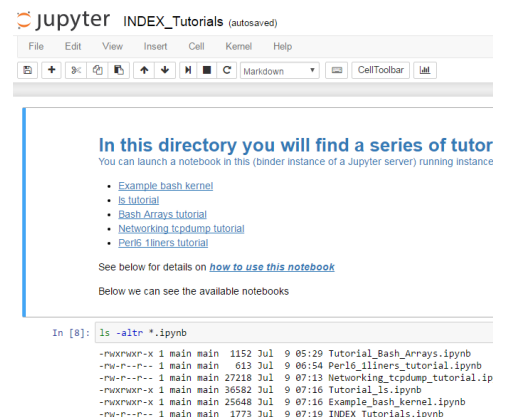
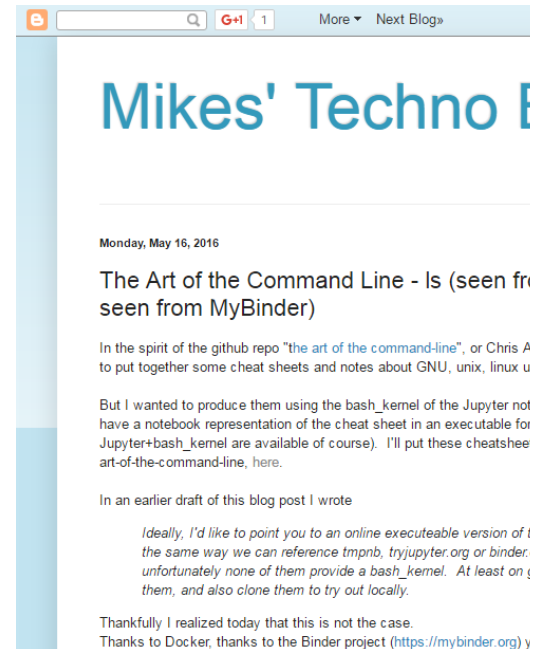
[Blog post](<http://mjbright.blogspot.fr/2016/05/the-art-of-command-line-ls-seen-from.html>) with link to "live tutorial"

In that [github \*\*repo\*\*]

([http://mybinder.org/repo/mjbright/binder\\_jupyter\\_notebook](http://mybinder.org/repo/mjbright/binder_jupyter_notebook)) is an

\*\*INDEX\*\* notebook to tutorial

notebooks in the same repo



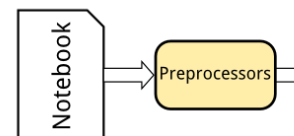
(<http://mybinder.org/repo/mjbright/b>)



## Everything Else: Cron e-mail status reports (n)

```
`` nbconvert --execute --template basic --to html Monitoring.ipynb ``
```

- --execute: run the notebook
- --template: specify o/p template
- --to: specify o/p format
- Input notebook



## Everything Else: Coming up ...

- Experiment with JupyterHub, nbgrader, Binder
  - Reimplement labs as graded assignments
- More Metakernel Bash experiments
- Make pull requests to Metakernel\_bash
- Propose this stuff outside of the Python community
- Xonsh\_kernel
  - Take advantage of new Python / unix-like shell
- CLing C++ interpreter kernel

## Questions ?

!The End](images/END\_sillywalk.jpg)

- ☐ [mjbright](https://github.com/mjbr
- ☐ [@mjbright](http://twitter.com/mjb1
- ☐ mjbrightfr AT gmail\_
- ☐ [mjbright.github.io](http://mjbright.gi

## The End ...

# References: IPython / Jupyter Books

## Learning IPython for Interactive Computing & Data Visualization

(<http://cyrille.rossant.net/books/>) Cyrille Rossant

### Introductory usage

## IPython Interactive Computing & Visualization Cookbook

(<http://cyrille.rossant.net/books/>) Cyrille Rossant

### Advanced usage

## The Jupyter GitBook

(<https://www.gitbook.com/book/carreau/jupyter-book/details>)

### Extension writing

## Documentation on ReadTheDocs

(<https://www.readthedocs.io/>)

### Extension writing

(<https://www.oreilly.com/learning/introducing-pandas-objects>)



By: J