

# Jupyter for Everything Else

Michael Bright, EuroPython 2016 - Bilbao, 22 July.



 [mjbright](#)  [@mjbright](#)  [mjbrightfr AT gmail](#)  [mjbright.github.io](#)



# About Me

**Solution Architect working at Hewlett-Packard Enterprise,  
Grenoble, France**

Working in the EMEA OpenNFV lab.

Interests:

- Docker, Docker, Docker, Jupyter, Python, OpenStack
- Run a local Python User Group in Grenoble
- From the UK, married to a Chilean, living in France for 24 years
- Argentinian Tango, Salsa, ...

TODO: add images ... Grenoble mountains, unicycle, Argentinian Tango, UK, NFV? SDN? Python UserGroup, Docker!!!, Jupyter, Linux, OStack



# OUTLINE

- Introduction: From IPython to Jupyter
- The Jupyter Project & Ecosystem:
  - Kernels, Widgets, Extensions, Tools
  - Incubating projects
  - External: Thebe, Hosting, Binder ...
- Jupyter for Everything Else
  - Blog, Present, Web, Command-line, Status reports



# IP[y]: IPYTHON - THE CONSOLE

*"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

## Initial 0.0.1 version [Gist](#)

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

```
IPython 4.0.0 -- An enhanced Interactive Python.
?          -> Introduction and overview of IPython's s
%quickref  -> Quick reference.
help       -> Python's own help system.
object?    -> Details about 'object', use 'object??' f

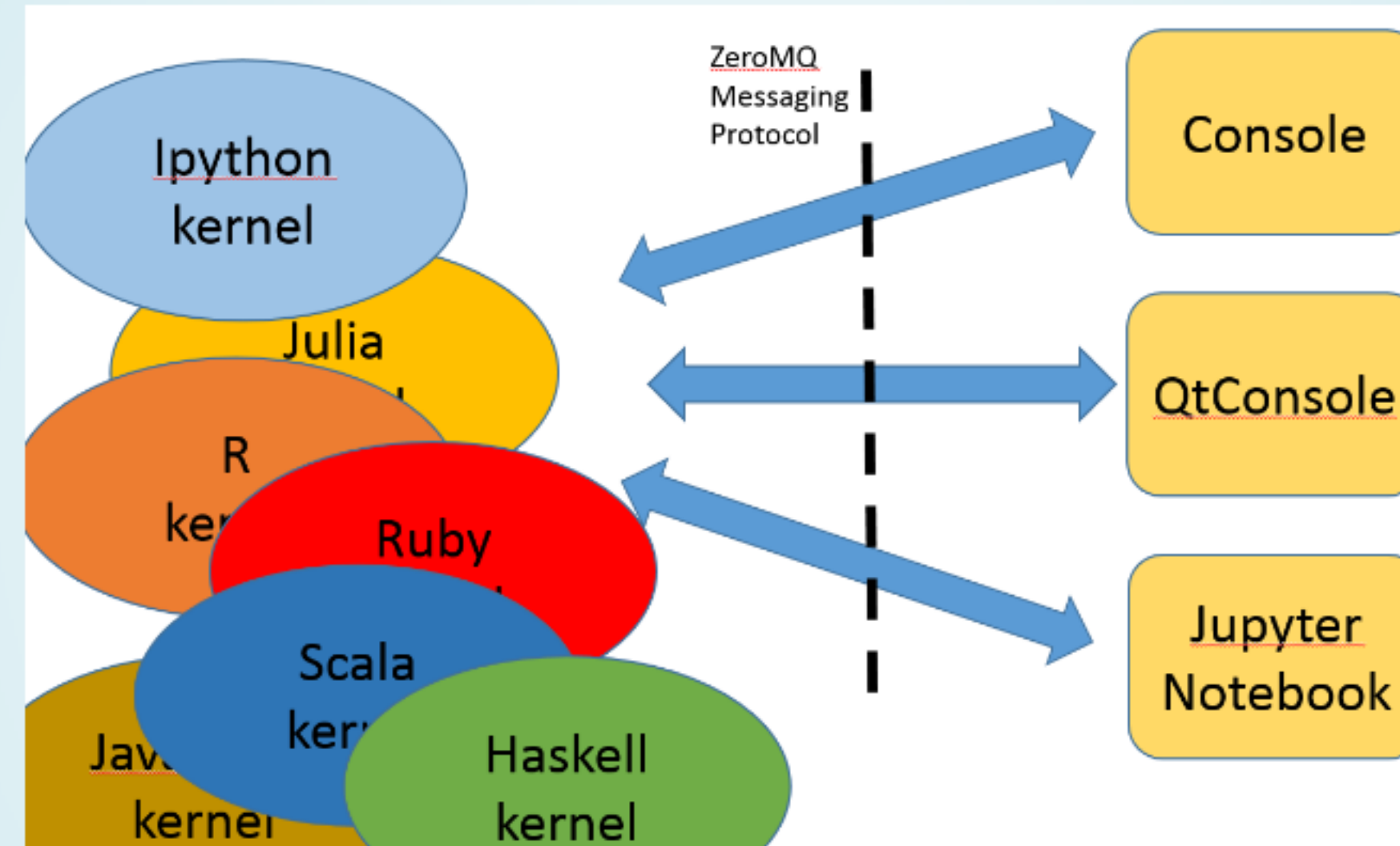
In [1]: def myfunc(msg):
...:     ''' prints msg '''
...:     print(msg)
...:

In [2]: myfunc("Hello World")
Hello World

In [3]: ?myfunc
Signature: myfunc(msg)
Docstring: prints msg
File:     d:\<ipython-input-1-e7b0ed71a6bd>
Type:     function
```



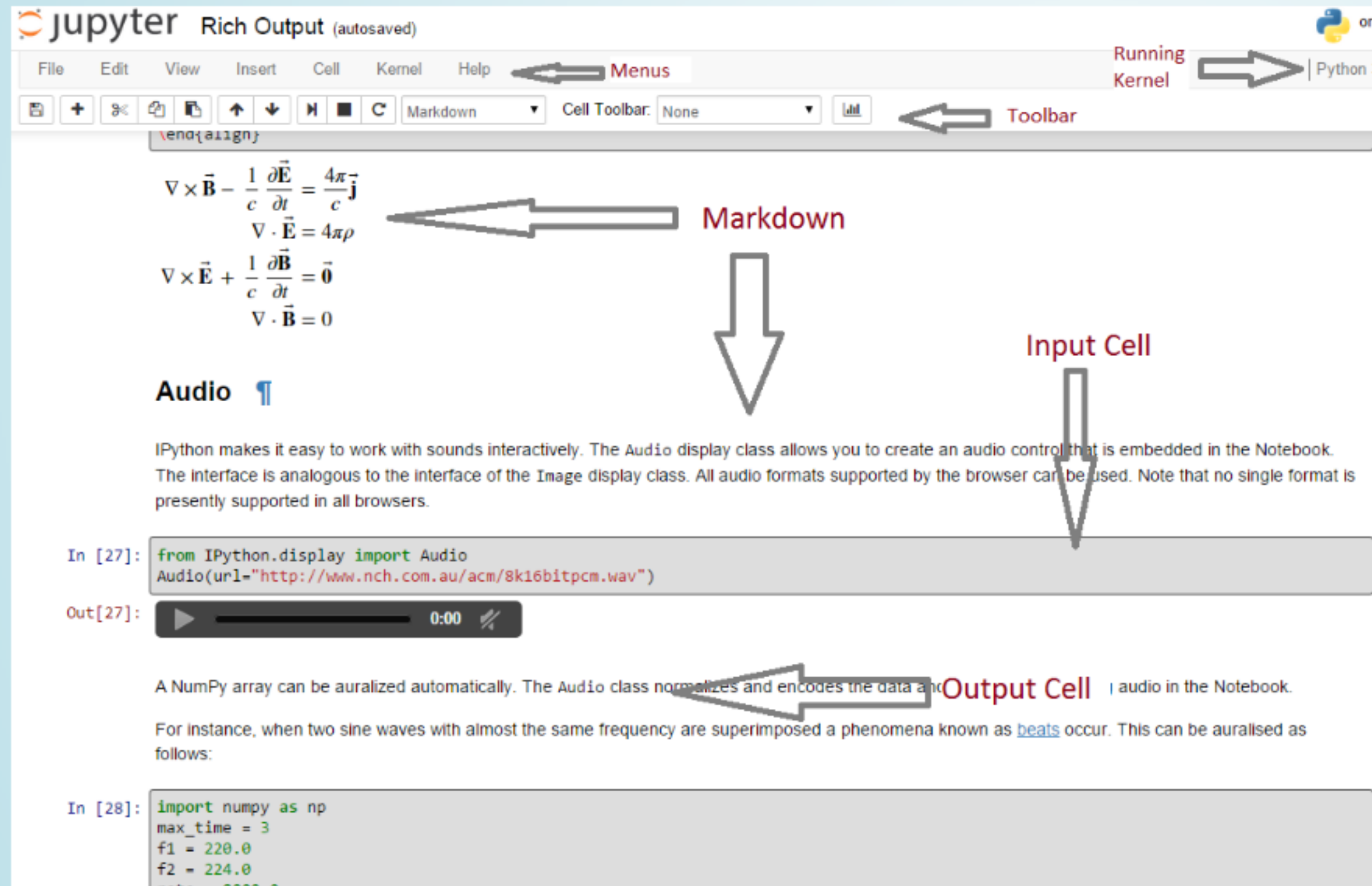
# THE JUPYTER PROJECT



A notebook runs under one kernel

# THE (JUPYTER) DASHBOARD & NOTEBOOK

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



The screenshot shows the Jupyter Notebook interface with several annotations:

- Menus:** A red arrow points to the top menu bar (File, Edit, View, Insert, Cell, Kernel, Help).
- Toolbar:** A red arrow points to the toolbar below the menu bar, which includes icons for saving, undo, redo, and other editing functions.
- Markdown:** A red arrow points to a cell containing LaTeX equations for Maxwell's equations:
$$\nabla \times \vec{B} - \frac{1}{c} \frac{\partial \vec{E}}{\partial t} = \frac{4\pi}{c} \vec{j}$$
$$\nabla \cdot \vec{E} = 4\pi \rho$$
$$\nabla \times \vec{E} + \frac{1}{c} \frac{\partial \vec{B}}{\partial t} = \vec{0}$$
$$\nabla \cdot \vec{B} = 0$$
- Input Cell:** A red arrow points to a code cell containing the following Python code:

```
from IPython.display import Audio
Audio(url="http://www.nch.com.au/acm/8k16bitpcm.wav")
```
- Output Cell:** A red arrow points to the output of the code cell, which is an audio player showing a play button and a 0:00 duration.





# JUPYTER: PUBLIC NOTEBOOKS

Jupyter notebooks are used in many

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

Site	URL	Info
nbviewer	<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>	> 200k notebooks [ <a href="#">Announcement - May '15</a> ]
IPython gallery	<a href="#">A-gallery-of-interesting-IPython-Notebooks</a>	many notebooks organized by domain
Notebook Gallery	<a href="#">[nb.binap.net]</a> <a href="#">[http://nb.bianp.net/]</a>	view submitted notebooks by 'most viewed' or 'data'





# JUPYTER: RUNNING NOTEBOOKS

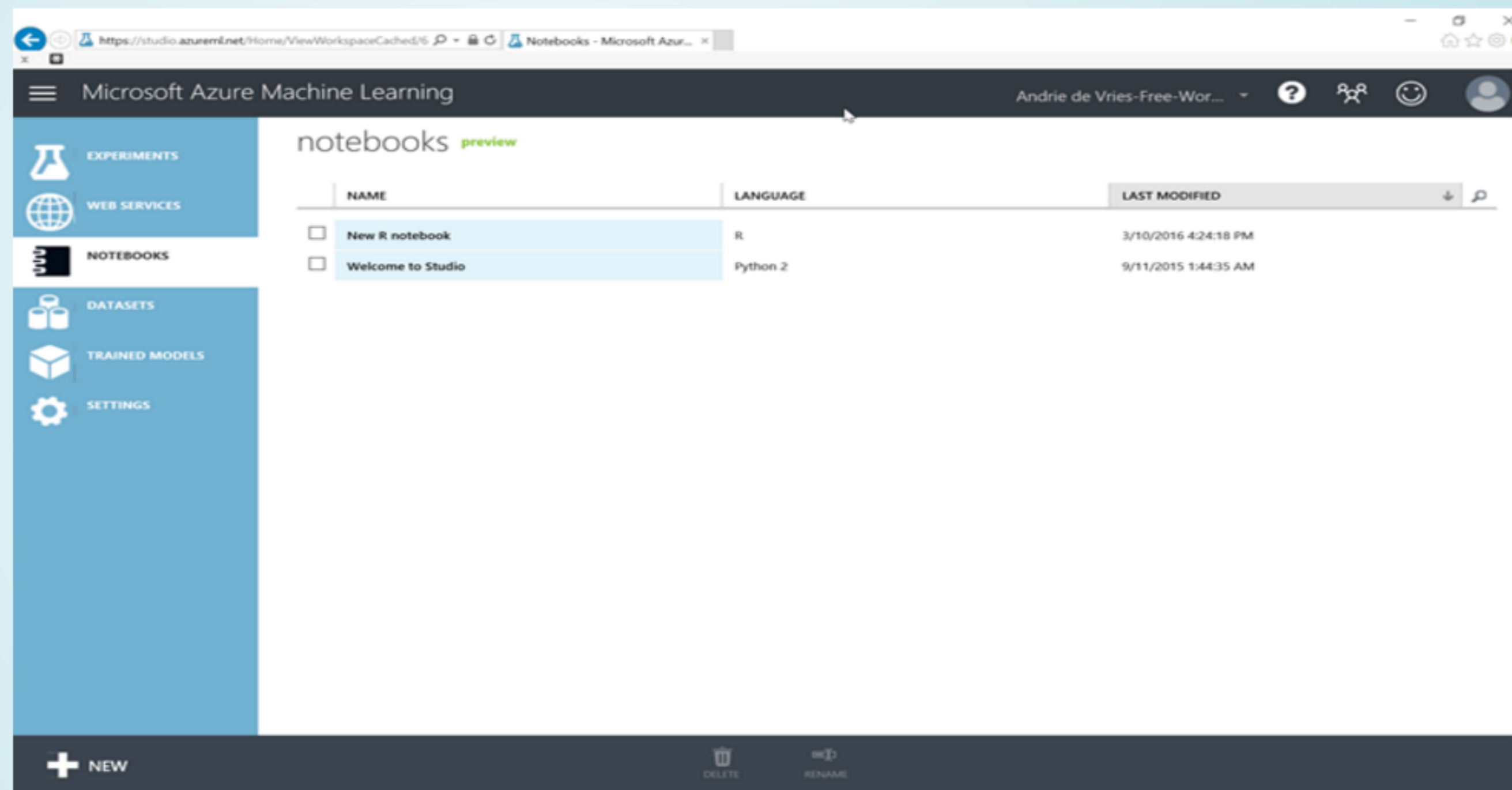
- Native OS Python distribution + Pip, or Anaconda
- JupyterHub, multi-user server
- Under Docker [e.g. *docker-stacks* images]
- Integrated into data science Cloud Hosting or plain IaaS:
  - Azure ML Studio
  - Google Cloud DataLab Beta
  - IBM Data Scientist Workbench
- Cloud hosted (ephemeral)
  - **tryjupyter.org** [uses *docker-demo* image]
  - **Binder** (<https://mybinder.org>), an example [github repo](#)



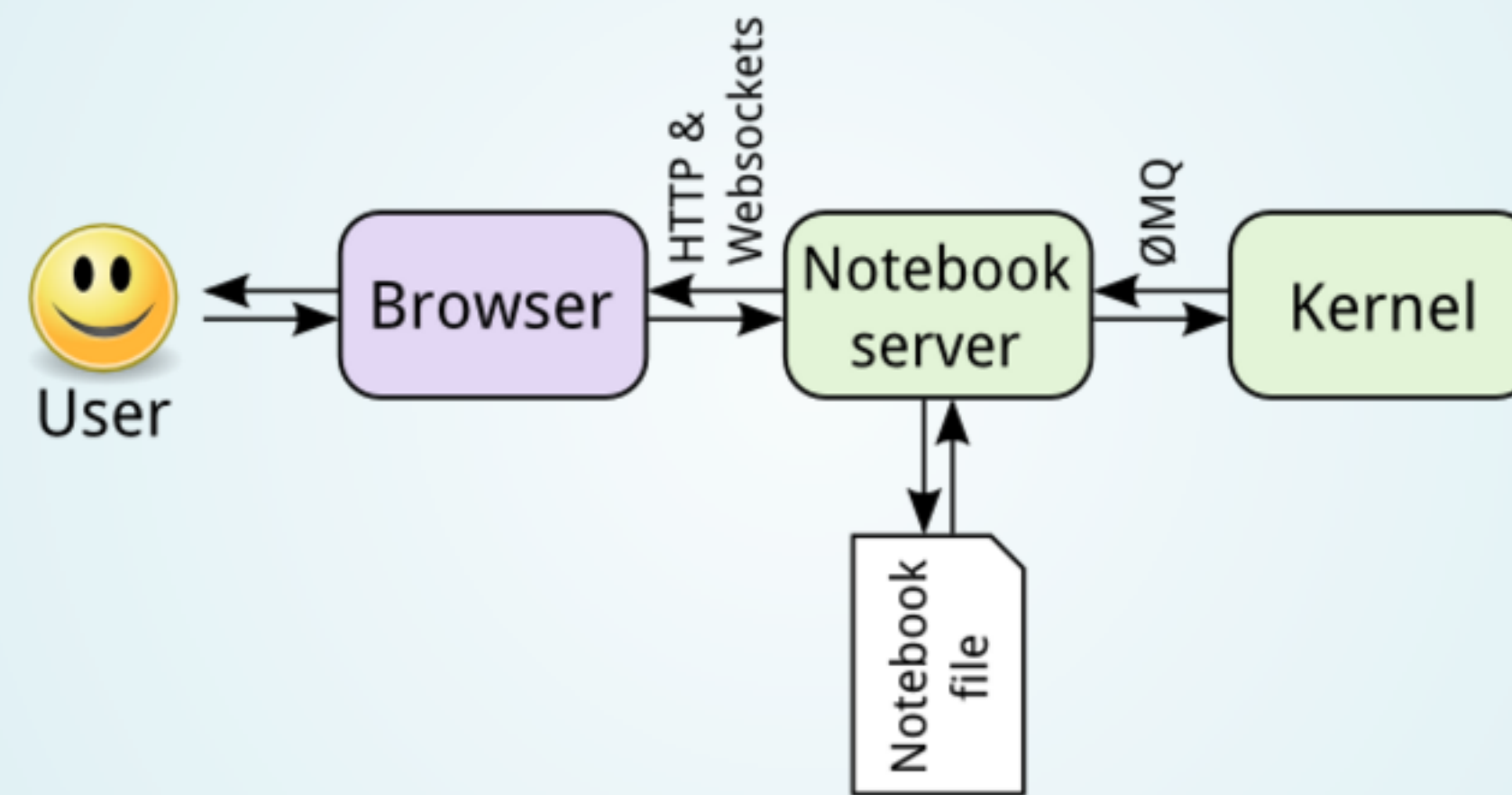
# JUPYTER & AZURE ML STUDIO

Jupyter  
Integration in  
Azure ML Studio

- R
- Python



# KERNELS, WIDGETS & EXTENSIONS



# ~ 50 KERNELS

Kernels are execution environments - typically a language [\[Kernels Page\]](#)

<b>IJulia</b>	<b>IRKernel</b>	<b>IPython</b>	<b>IRuby</b>	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

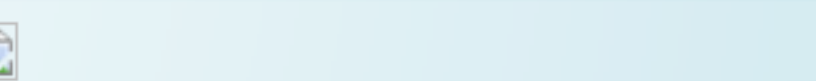


# WIDGETS

Widgets are eventful python objects with a representation in the browser.  
[\[documentation\]](#)

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)
- Installed to <http://localhost:8888/nbextensions/>


Many extensions available, including:













- **RISE** - **these slides are running under Jupyter**
- [nbgrader](#) - 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 

         Markdown  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

**sparkmagic**

Jupyter magics and kernels for working with remote Spark clusters

**declarativewidgets**

Declare Widgets in HTML

**dashboards**

Create Dashboards from Notebooks

**contentmanagement**

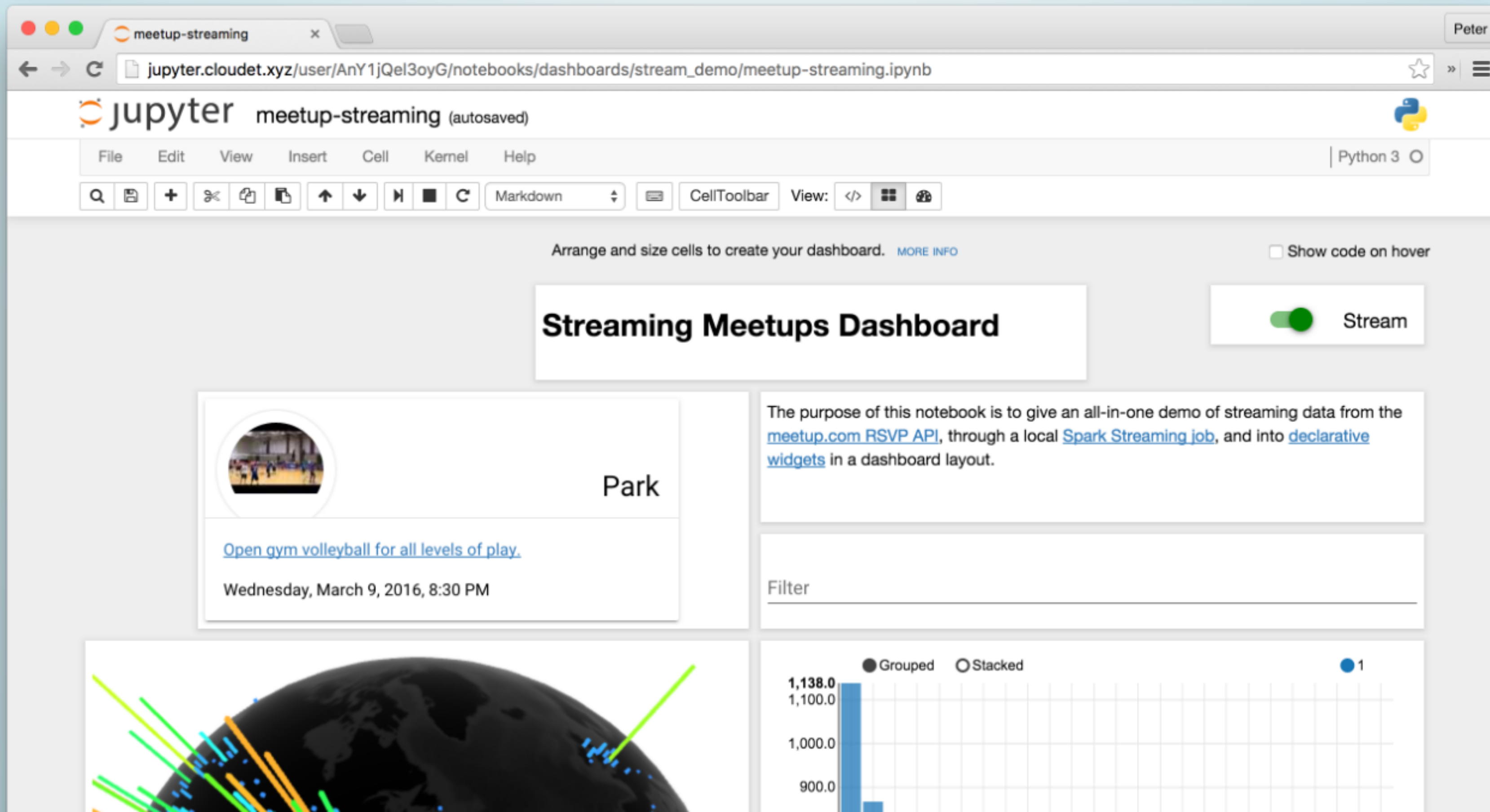
Extensions for search, notebook modules/[cookbooks](#), ToC, [bundlers](#) [vid](#)

**kernel\_gateway**

Support different protocols to Jupyter server, e.g. non-nb web clients, u-services, cluster

# INCUBATOR: JUPYTER DASHBOARDS

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



# JUPYTERLAB

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

The screenshot displays the JupyterLab user interface. On the left, there is a sidebar with two tabs: 'Files' and 'Commands'. The 'Files' tab is active, showing a file browser with a list of files and folders, including 'build', 'node\_modules', 'bash.ipynb', 'example.ipynb', 'Untitled.ipynb', 'index.html', 'index.js', 'jupyter.png', 'main.py', 'package.json', and 'webpack.conf.js'. The 'Commands' tab is also visible. The main area of the interface shows a notebook with several cells. The first cell contains Python code that prints 'hello world', 'output to stderr', and 'some more stdout text'. The second cell contains code to create an interactive slider widget. The third cell is a markdown cell with the text '\*It\* \*\*really\*\* is!'. The fourth cell contains the text 'this is a syntax error'. The notebook interface includes a toolbar with various icons for file operations and a 'Code' button. The top of the notebook shows the current file 'example.ipynb' and the selected kernel 'Python 3'.

Files

Commands

Name Last Modified

- build 4 days ago
- node\_modules 8 hours ago
- bash.ipynb 2 minutes ago
- example.ipynb a minute ago
- Untitled.ipynb 14 days ago
- index.html 9 days ago
- index.js 14 days ago
- jupyter.png 2 months ago
- main.py 20 days ago
- package.json 16 days ago
- webpack.conf.js a month ago

About example.ipynb x

Python 3

In [6]:

```
import sys
print('hello world', flush=True)
for i in range(3):
    print(i, flush=True)
print('output to stderr', file=sys.stderr, flush=True)
print('some more stdout text', flush=True)
```

hello world  
0  
1  
2

output to stderr

some more stdout text

In [1]:

```
from ipywidgets import IntSlider
IntSlider() # slider appears at the top of the notebook
```

In [ ]:

```
# Markdown Cell

*It* **really** is!
```

In [2]:

```
this is a syntax error
```





# EXTERNAL JUPYTER PROJECTS

There are many external projects such as *Beaker*, *Hydrogen (ATOM)*, *EIN (Emacs)*, *Rodeo*, *SageMathCloud* integrating Jupyter.

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

- e.g. **Thebe** (O'Reilly)
- Nature, Scientific American Magazines is a simplified notebook interface

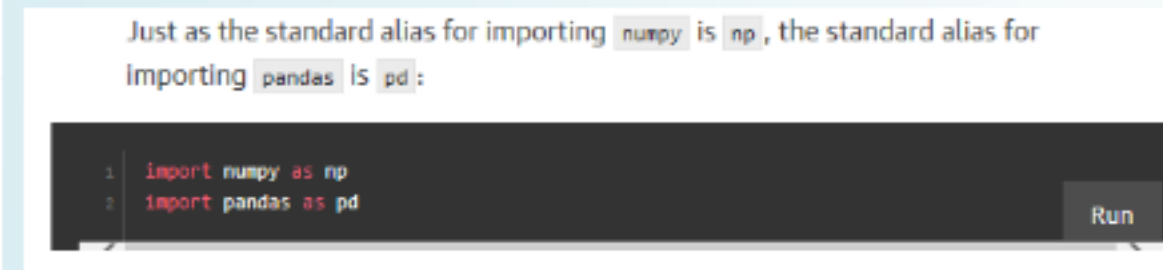
Educators

- tutorials, assignments, presentations, documenting
- MOOCs - online education:
  - notebook-based (**Edx/Apache Spark**)

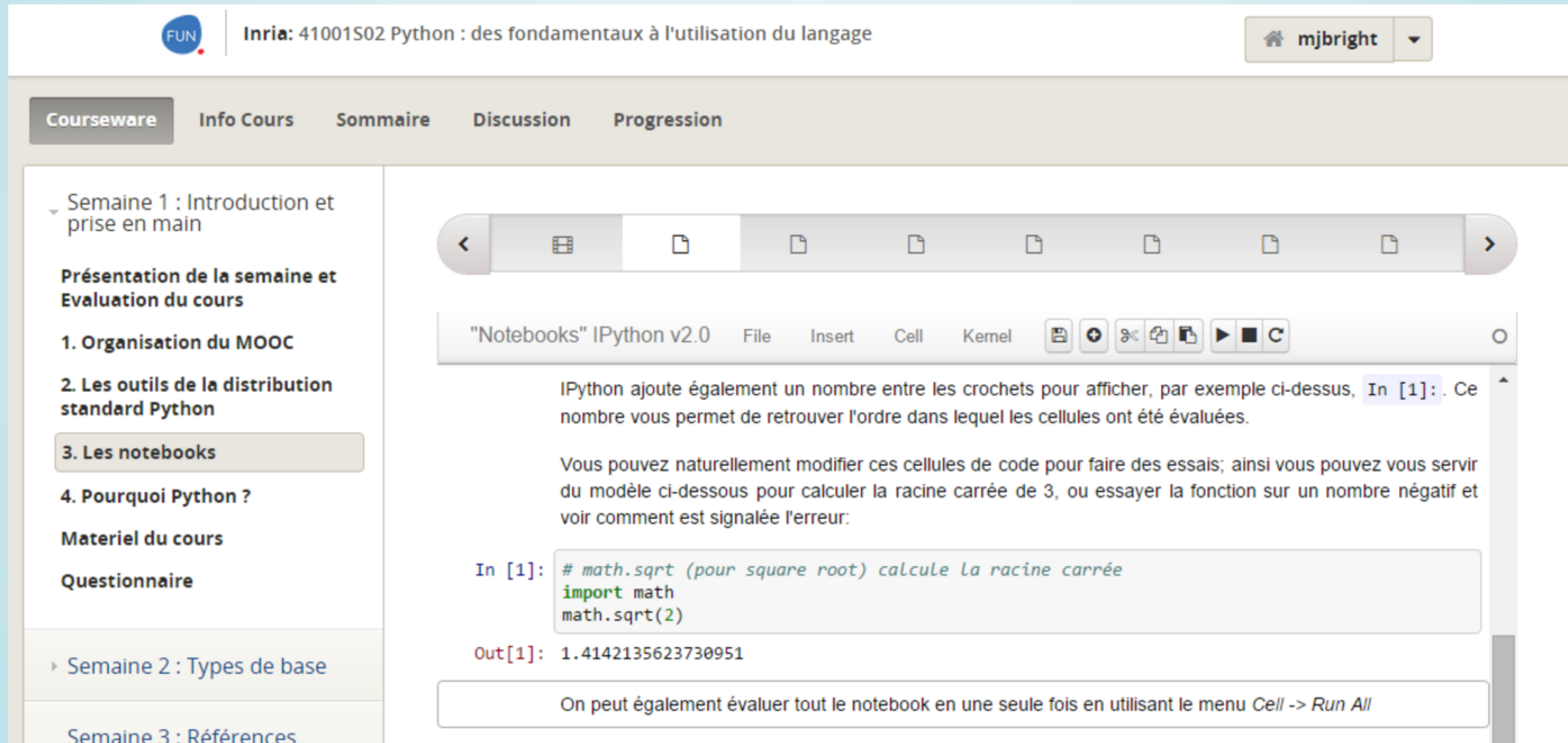


# OREILLY BLOG ARTICLE - USING THEBE

This blog post contains modifiable, runnable code cells with a **RUN** button as shown below



# MOOCS, E.G. F.U.N., GWU, EDX/SPARK...



The screenshot shows the FUN MOOC interface. At the top, the course title is 'Inria: 41001S02 Python : des fondamentaux à l'utilisation du langage' and the user 'mjbright' is logged in. The navigation bar includes 'Courseware', 'Info Cours', 'Sommaire', 'Discussion', and 'Progression'. The left sidebar lists the course structure: 'Semaine 1 : Introduction et prise en main' (expanded), 'Présentation de la semaine et Evaluation du cours', '1. Organisation du MOOC', '2. Les outils de la distribution standard Python', '3. Les notebooks' (highlighted), '4. Pourquoi Python ?', 'Materiel du cours', and 'Questionnaire'. Below this, 'Semaine 2 : Types de base' and 'Semaine 3 : Références' are listed. The main content area displays an IPython notebook interface. It features a toolbar with icons for file operations and execution. The notebook text shows a comment in French: 'IPython ajoute également un nombre entre les crochets pour afficher, par exemple ci-dessus, In [1]: . Ce nombre vous permet de retrouver l'ordre dans lequel les cellules ont été évaluées.' followed by instructions on how to modify code cells. A code cell is shown with the following Python code: 

```
In [1]: # math.sqrt (pour square root) calcule la racine carrée
import math
math.sqrt(2)
```

 The output of this cell is 'Out[1]: 1.4142135623730951'. At the bottom of the notebook interface, a message states: 'On peut également évaluer tout le notebook en une seule fois en utilisant le menu Cell -> Run All'.



# 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
  - Many HTML, CSS, JS capabilities if you **proceed with care**
  - d3.js animations if
    - Need more interactivity
    - Prototyping a D3 project
    - Reusing existing D3 e.g. from <http://bl.ocks.org>
- SVG
- [Example notebook](#)

Select Theme:



# Everything Else: web technologies

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

Select Theme: mybloodredsky ▾

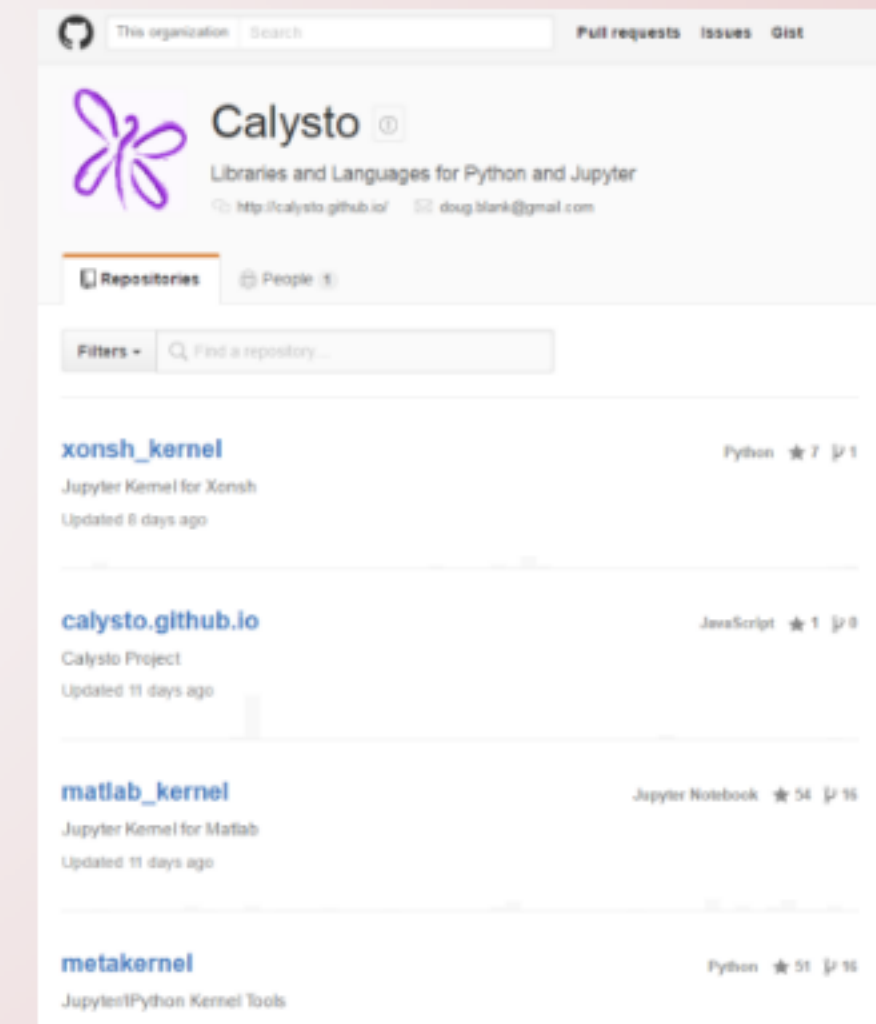


# Everything Else: command-line

Two bash kernels are available for Jupyter `bash_kernel` and `calysto/metakernel_bash`

## Calysto Metakernels

- easy to fix for Windows/Cygwin
- Provides magics
- Family of metakernels
- Under development



# Everything Else: command-line

## Why?

Inspired by the notebook as an educational tool  
I wanted to use it for command-line tasks

- **Docker demos / labs in Jupyter**
- Why not Bash as 1st-class citizen with magics, graphics ?
- **Example notebook**
- Write command-line tutorials, cheat sheets  
in an easy to maintain "live notebook"  
(runnable) format.

```
Adding HTML o/p capabilities to Bash in the Notebook
I've provided an 'htm' function which enables display of html within the notebook.
Below examples of pretty printing HTML from various sources

In [1]:
echo "<h1>An html header <!-- A comment --></h1>" | htm1

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>value1</td>
<td>value2</td>
<td>value3</td>
<td>value4</td>
</tr></table>" | htm1

A table


| col1   | col2   | col3   | col4   |
|--------|--------|--------|--------|
| val1   | val2   | val3   | val4   |
| value1 | value2 | value3 | value4 |


```

# Everything Else: Binder "live notebooks"



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

submit

## 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 first. Check out an [example](#).



# Everything Else: "live notebooks" as runnable tutorials

We can create live tutorials online on Binder.

The notebook server can be launched by clicking on

the binder icon in a [github repo](#)

launch binder

I wrote a [Blog post](#) with link to "**live notebook**" of bash tutorials

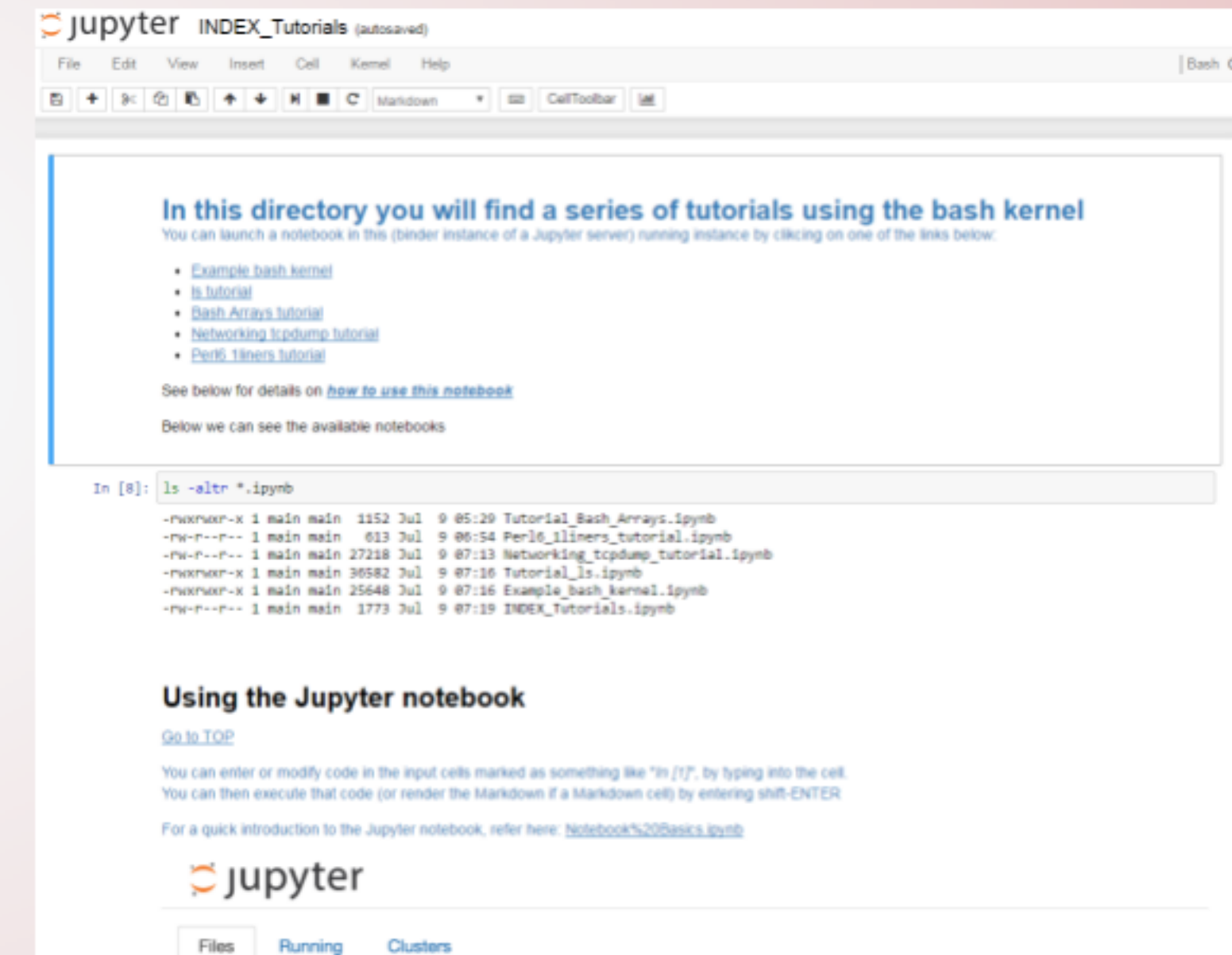




# Everything Else: Binder command-line "live notebooks"

In that [github repo](#) I created an **INDEX notebook**

other notebooks in the same repo



# Everything Else: Slideshows

This **slideshow** is made using Jupyter with the RISE extensions

RISE adds special "Slide Type" menu options to each cell to specify one of

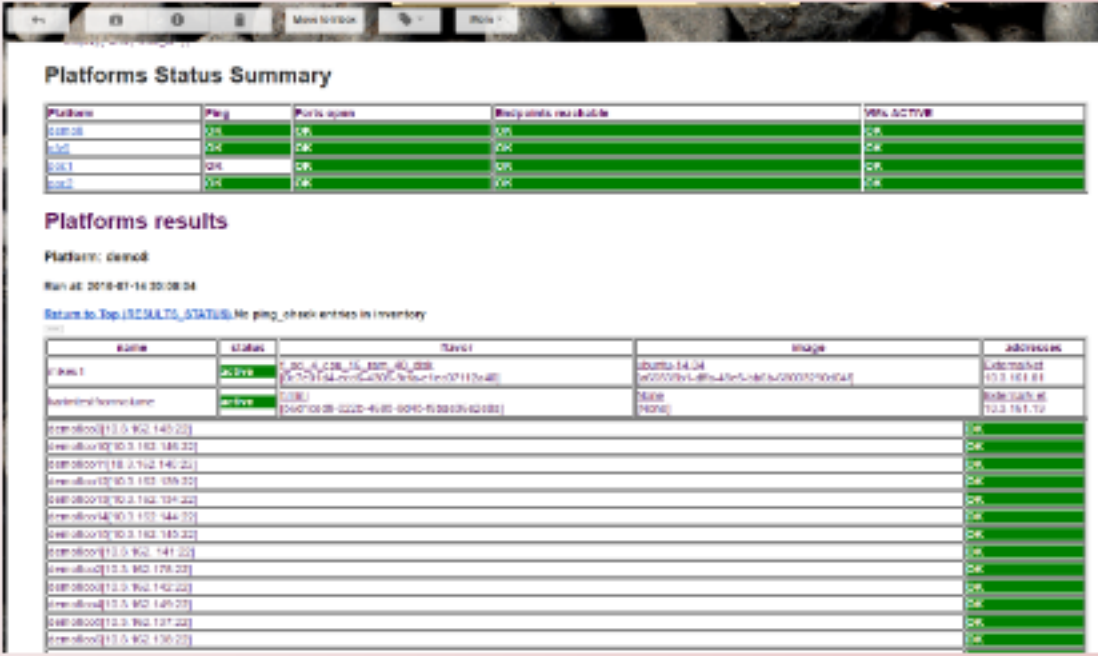
- Slide
- Sub-Slide
- Fragment
- Notes
- Skip



# Everything Else: Automated e-mail status reports

Using nbconvert we can **execute notebooks** from the command-line

- Automatically run the notebook under cron
- **nbconvert** notebook result to html
- send html **report** as e-mail via cron
- Capabilities created incrementally in notebook,  
code migrated to modules to reduce notebook (report) code



**Platforms Status Summary**

Platform	Ping	Ports open	Ports closed	VPN Active
10.10.10.1	OK	OK	OK	OK
10.10.10.2	OK	OK	OK	OK
10.10.10.3	OK	OK	OK	OK
10.10.10.4	OK	OK	OK	OK

**Platforms results**

Platform: cisco

Run at: 2016-01-14 10:00:04

[Return to Top](#) | [SCALTS STATUS](#) | [No ping status entries in inventory](#)

name	status	status	image	addresses
10.10.10.1	OK	OK	10.10.10.1	10.10.10.1
10.10.10.2	OK	OK	10.10.10.2	10.10.10.2
10.10.10.3	OK	OK	10.10.10.3	10.10.10.3
10.10.10.4	OK	OK	10.10.10.4	10.10.10.4
10.10.10.5	OK	OK	10.10.10.5	10.10.10.5
10.10.10.6	OK	OK	10.10.10.6	10.10.10.6
10.10.10.7	OK	OK	10.10.10.7	10.10.10.7
10.10.10.8	OK	OK	10.10.10.8	10.10.10.8
10.10.10.9	OK	OK	10.10.10.9	10.10.10.9
10.10.10.10	OK	OK	10.10.10.10	10.10.10.10
10.10.10.11	OK	OK	10.10.10.11	10.10.10.11
10.10.10.12	OK	OK	10.10.10.12	10.10.10.12
10.10.10.13	OK	OK	10.10.10.13	10.10.10.13
10.10.10.14	OK	OK	10.10.10.14	10.10.10.14
10.10.10.15	OK	OK	10.10.10.15	10.10.10.15
10.10.10.16	OK	OK	10.10.10.16	10.10.10.16
10.10.10.17	OK	OK	10.10.10.17	10.10.10.17
10.10.10.18	OK	OK	10.10.10.18	10.10.10.18
10.10.10.19	OK	OK	10.10.10.19	10.10.10.19
10.10.10.20	OK	OK	10.10.10.20	10.10.10.20
10.10.10.21	OK	OK	10.10.10.21	10.10.10.21
10.10.10.22	OK	OK	10.10.10.22	10.10.10.22
10.10.10.23	OK	OK	10.10.10.23	10.10.10.23
10.10.10.24	OK	OK	10.10.10.24	10.10.10.24
10.10.10.25	OK	OK	10.10.10.25	10.10.10.25
10.10.10.26	OK	OK	10.10.10.26	10.10.10.26
10.10.10.27	OK	OK	10.10.10.27	10.10.10.27
10.10.10.28	OK	OK	10.10.10.28	10.10.10.28
10.10.10.29	OK	OK	10.10.10.29	10.10.10.29
10.10.10.30	OK	OK	10.10.10.30	10.10.10.30
10.10.10.31	OK	OK	10.10.10.31	10.10.10.31
10.10.10.32	OK	OK	10.10.10.32	10.10.10.32
10.10.10.33	OK	OK	10.10.10.33	10.10.10.33
10.10.10.34	OK	OK	10.10.10.34	10.10.10.34
10.10.10.35	OK	OK	10.10.10.35	10.10.10.35
10.10.10.36	OK	OK	10.10.10.36	10.10.10.36
10.10.10.37	OK	OK	10.10.10.37	10.10.10.37
10.10.10.38	OK	OK	10.10.10.38	10.10.10.38
10.10.10.39	OK	OK	10.10.10.39	10.10.10.39
10.10.10.40	OK	OK	10.10.10.40	10.10.10.40
10.10.10.41	OK	OK	10.10.10.41	10.10.10.41
10.10.10.42	OK	OK	10.10.10.42	10.10.10.42
10.10.10.43	OK	OK	10.10.10.43	10.10.10.43
10.10.10.44	OK	OK	10.10.10.44	10.10.10.44
10.10.10.45	OK	OK	10.10.10.45	10.10.10.45
10.10.10.46	OK	OK	10.10.10.46	10.10.10.46
10.10.10.47	OK	OK	10.10.10.47	10.10.10.47
10.10.10.48	OK	OK	10.10.10.48	10.10.10.48
10.10.10.49	OK	OK	10.10.10.49	10.10.10.49
10.10.10.50	OK	OK	10.10.10.50	10.10.10.50
10.10.10.51	OK	OK	10.10.10.51	10.10.10.51
10.10.10.52	OK	OK	10.10.10.52	10.10.10.52
10.10.10.53	OK	OK	10.10.10.53	10.10.10.53
10.10.10.54	OK	OK	10.10.10.54	10.10.10.54
10.10.10.55	OK	OK	10.10.10.55	10.10.10.55
10.10.10.56	OK	OK	10.10.10.56	10.10.10.56
10.10.10.57	OK	OK	10.10.10.57	10.10.10.57
10.10.10.58	OK	OK	10.10.10.58	10.10.10.58
10.10.10.59	OK	OK	10.10.10.59	10.10.10.59
10.10.10.60	OK	OK	10.10.10.60	10.10.10.60
10.10.10.61	OK	OK	10.10.10.61	10.10.10.61
10.10.10.62	OK	OK	10.10.10.62	10.10.10.62
10.10.10.63	OK	OK	10.10.10.63	10.10.10.63
10.10.10.64	OK	OK	10.10.10.64	10.10.10.64
10.10.10.65	OK	OK	10.10.10.65	10.10.10.65
10.10.10.66	OK	OK	10.10.10.66	10.10.10.66
10.10.10.67	OK	OK	10.10.10.67	10.10.10.67
10.10.10.68	OK	OK	10.10.10.68	10.10.10.68
10.10.10.69	OK	OK	10.10.10.69	10.10.10.69
10.10.10.70	OK	OK	10.10.10.70	10.10.10.70
10.10.10.71	OK	OK	10.10.10.71	10.10.10.71
10.10.10.72	OK	OK	10.10.10.72	10.10.10.72
10.10.10.73	OK	OK	10.10.10.73	10.10.10.73
10.10.10.74	OK	OK	10.10.10.74	10.10.10.74
10.10.10.75	OK	OK	10.10.10.75	10.10.10.75
10.10.10.76	OK	OK	10.10.10.76	10.10.10.76
10.10.10.77	OK	OK	10.10.10.77	10.10.10.77
10.10.10.78	OK	OK	10.10.10.78	10.10.10.78
10.10.10.79	OK	OK	10.10.10.79	10.10.10.79
10.10.10.80	OK	OK	10.10.10.80	10.10.10.80
10.10.10.81	OK	OK	10.10.10.81	10.10.10.81
10.10.10.82	OK	OK	10.10.10.82	10.10.10.82
10.10.10.83	OK	OK	10.10.10.83	10.10.10.83
10.10.10.84	OK	OK	10.10.10.84	10.10.10.84
10.10.10.85	OK	OK	10.10.10.85	10.10.10.85
10.10.10.86	OK	OK	10.10.10.86	10.10.10.86
10.10.10.87	OK	OK	10.10.10.87	10.10.10.87
10.10.10.88	OK	OK	10.10.10.88	10.10.10.88
10.10.10.89	OK	OK	10.10.10.89	10.10.10.89
10.10.10.90	OK	OK	10.10.10.90	10.10.10.90
10.10.10.91	OK	OK	10.10.10.91	10.10.10.91
10.10.10.92	OK	OK	10.10.10.92	10.10.10.92
10.10.10.93	OK	OK	10.10.10.93	10.10.10.93
10.10.10.94	OK	OK	10.10.10.94	10.10.10.94
10.10.10.95	OK	OK	10.10.10.95	10.10.10.95
10.10.10.96	OK	OK	10.10.10.96	10.10.10.96
10.10.10.97	OK	OK	10.10.10.97	10.10.10.97
10.10.10.98	OK	OK	10.10.10.98	10.10.10.98
10.10.10.99	OK	OK	10.10.10.99	10.10.10.99
10.10.10.100	OK	OK	10.10.10.100	10.10.10.100

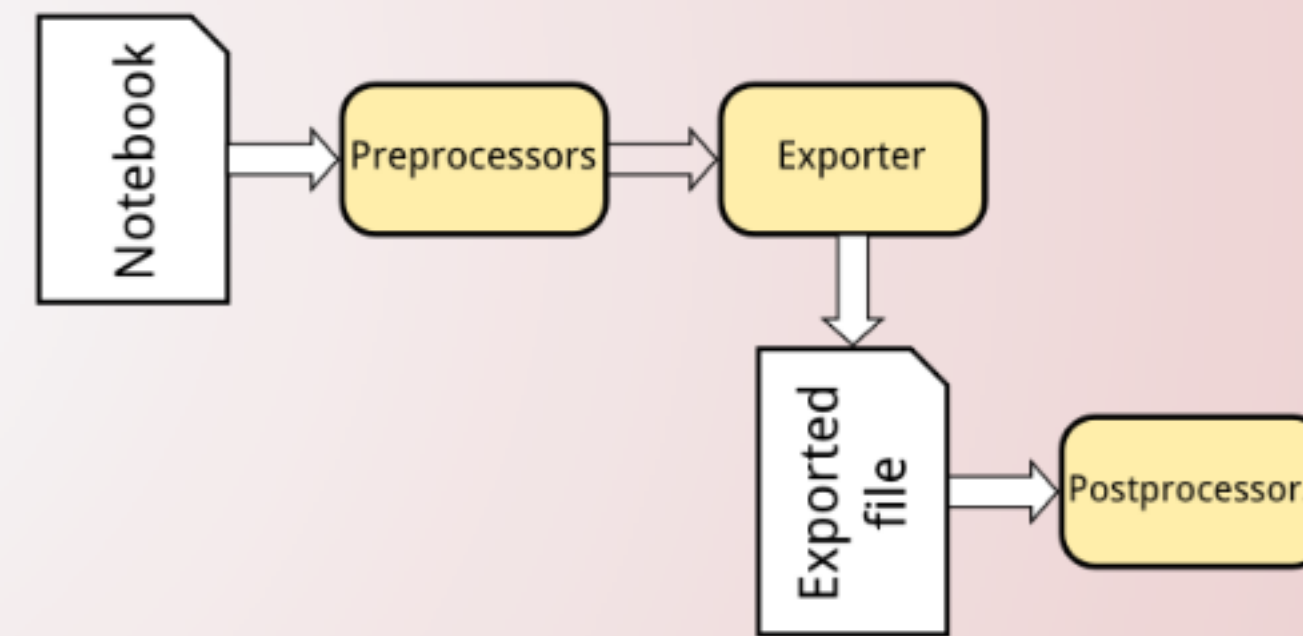


# Everything Else: Automated e-mail status reports (nbconvert)

We can use nbconvert to automatically run the notebook under cron and send the results by e-mail

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

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







# Everything Else: Coming up ...

- Experiment with JupyterHub, nbgrader, Binder
  - I'd like to reimplement some labs as a set of 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 ?



 [mjbright](#)  [@mjbright](#)  [mjbrightfr AT gmail](#)  [mjbright.github.io](#)



The End ...

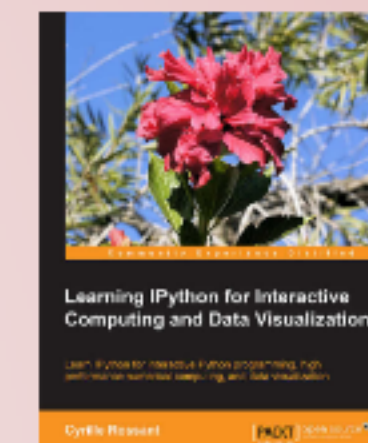


# References: IPython / Jupyter Books

## Learning IPython for Interactive Computing & Data Visualization

Cyrille Rossant

*Introductory usage*



## IPython Interactive Computing & Visualization Cookbook

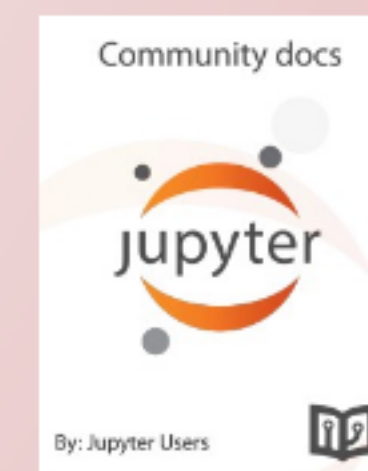
Cyrille Rossant

*Advanced usage*



## The Jupyter GitBook

*Extension writing*



## Documentation on ReadTheDocs

*Extension writing*