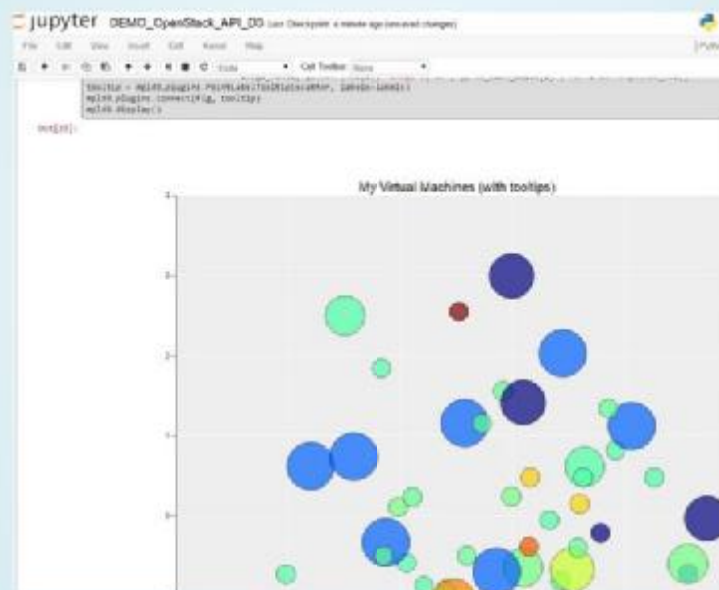






# IPython ... vers Jupyter

17 Oct. 2015, *Pyconfr - Pau*  
Michael Bright.



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



# Outline

- What are Jupyter & IPython ?
  - IPython beginnings
    - The console and notebook
    - The messaging protocol and kernels
  - Jupyter Project
- Jupyter Notebook demo
- Ecosystem
  - What is Jupyter being used for ?
  - Kernels / Extensions
  - Related Projects



What are Jupyter & IPython ?

```
IP[y]:
```



# IPython - the Console

"*an afternoon hack*" (Nov 2001) by Fernando Perez, to facilitate *Scientific Exploration*:

- Individual Exploratory Analysis
- Collaborative work
- Publication of **reproducible results**
- Education



```
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??' s

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

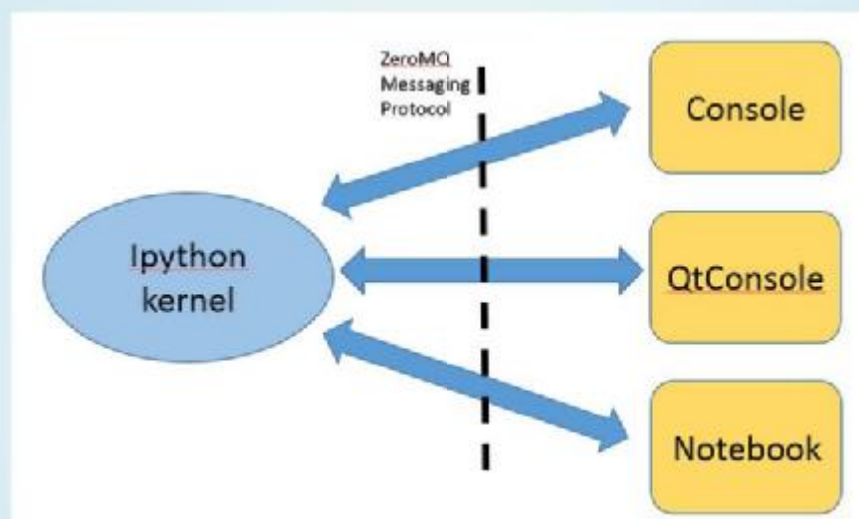
In [2]: myfunc("Hello World")
Hello World

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

IPython - console-based **REPL** (Read-Eval-Print-Loop)

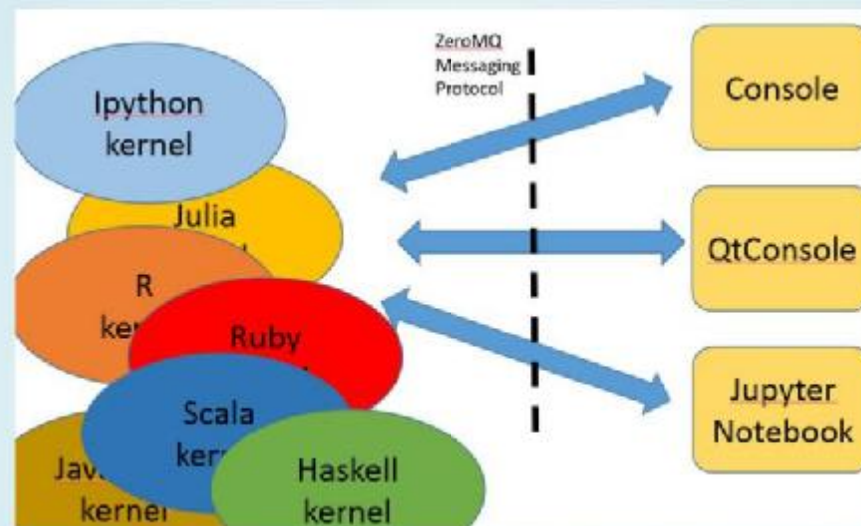


# The messaging protocol



Created in 2010, inspired by the need to support multi-processing  
it enabled multiple front-ends for the IPython kernel

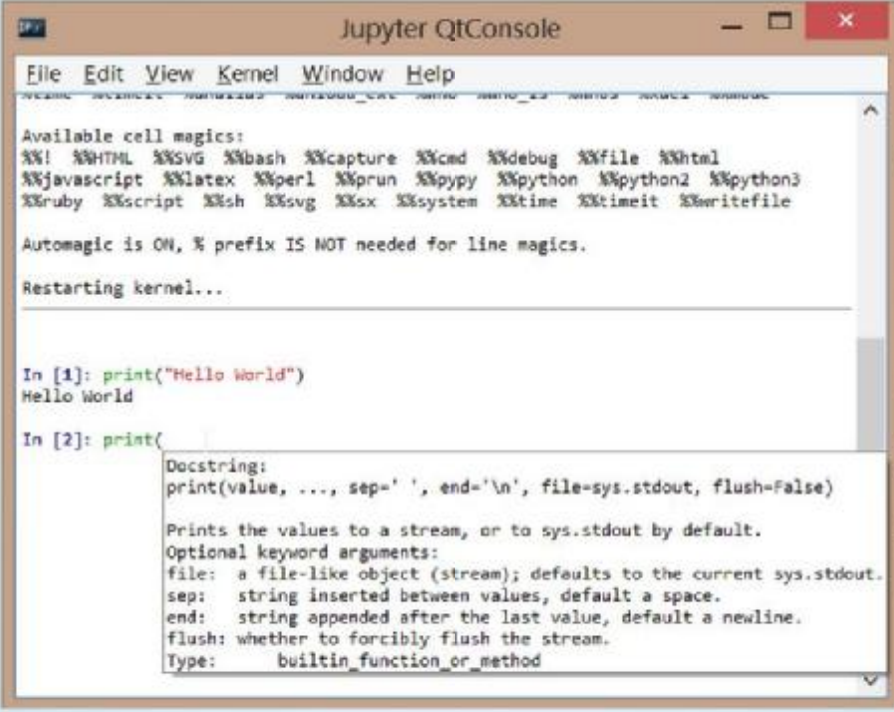
... the protocol enables different kernels



**Note:** these are persistent kernels - unlike IPython **cell magics**.  
A notebook runs under one kernel



# IPython - the QtConsole



The screenshot shows a window titled "Jupyter QtConsole" with a menu bar (File, Edit, View, Kernel, Window, Help). The main text area displays the following content:

```
Available cell magics:
%%! %%HTML %%SVG %%bash %%capture %%cmd %%debug %%file %%html
%%javascript %%latex %%perl %%prun %%pypy %%python %%python2 %%python3
%%ruby %%script %%sh %%svg %%sx %%system %%time %%timeit %%writefile

Automagic is ON, % prefix IS NOT needed for line magics.

Restarting kernel...
```

---

```
In [1]: print("Hello World")
Hello World

In [2]: print(
```

A tooltip is visible over the second line of code, showing the docstring for the `print` function:

```
Docstring:
print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)

Prints the values to a stream, or to sys.stdout by default.
Optional keyword arguments:
file: a file-like object (stream); defaults to the current sys.stdout.
sep: string inserted between values, default a space.
end: string appended after the last value, default a newline.
flush: whether to forcibly flush the stream.
Type: builtin_function_or_method
```

# The Jupyter Notebook

- A web application to create and share documents containing live code, equations, visualizations and explanatory text as a **(reproducible) narrative**.
- Used across a vast number of application domains

The screenshot shows the Jupyter Notebook web interface. At the top, the title bar reads 'jupyter Rich Output (autosaved)' and the status bar shows 'Running Kernel' and 'Python 3'. The menu bar includes 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', and 'Help'. Below the menu bar is a toolbar with icons for file operations and cell management. The main content area is divided into three sections:

- Markdown Cell:** Contains mathematical equations for Maxwell's equations:
$$\nabla \times \vec{B} - \frac{1}{c} \frac{\partial \vec{E}}{\partial t} = \frac{4\pi}{c} \vec{j}$$
$$\nabla \cdot \vec{E} = 4\pi \rho$$
$$\nabla \times \vec{E} + \frac{1}{c} \frac{\partial \vec{B}}{\partial t} = \vec{0}$$
$$\nabla \cdot \vec{B} = 0$$
Annotations: A red arrow labeled 'Menus' points to the top menu bar. A red arrow labeled 'Toolbar' points to the toolbar. A red arrow labeled 'Markdown' points to the text area.
- Input Cell:** Contains Python code to import the Audio class and load a file:

```
In [27]: from IPython.display import Audio
Audio(url="http://www.nch.com.au/acm/8k16bitpcm.wav")
```

Annotation: A red arrow labeled 'Input Cell' points to the code area.
- Output Cell:** Shows the output of the code, which is an audio player interface with a play button and a progress bar.

```
Out[27]:
```

Annotations: A red arrow labeled 'Output Cell' points to the output area. A red arrow labeled 'Audio' points to the audio player.

At the bottom right, there is a small navigation icon with the text '2/5'.



# The notebook file format (.ipynb)

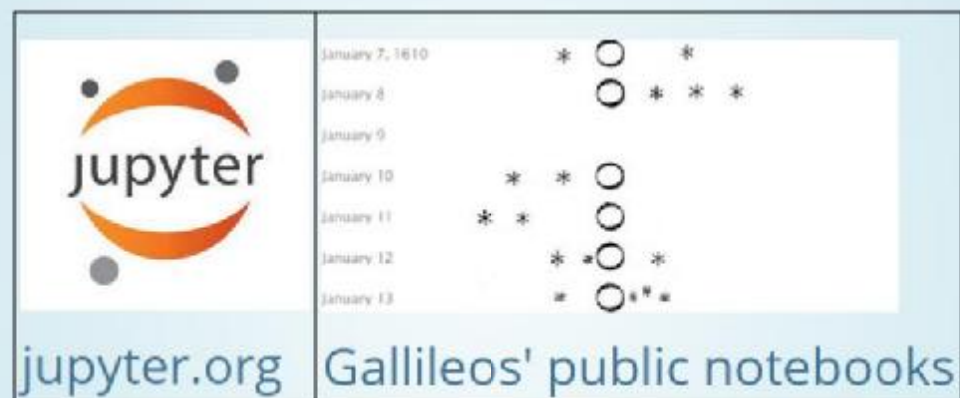
*uses JSON (JavaScript Object Notation)*

```
{ "metadata": { "kernel_spec": {  
    "language": "bash", "name": "bash" }, },  
  "cells": [  
    {  
      "cell_type": "code",  
      "source": [ "ls -alt /tmp/" ],  
      "outputs": [ {  
        "name": "stdout",  
        "text": [  
          "total 72\r\n",  
          "-rw-----  1 root root 32768 Aug 13 19:50 348\r\n",  
          "drwxrwxrwt 11 root root  4096 Aug 13 19:48 ..\r\n",  
          "drwxr-xr-x  2 root root  4096 Aug 12 14:52 .\r\n",  
          "-rw-----  1 root root 32768 Aug 12 14:52 449\r\n"  
        ]  
      }  
    ]  
  }  
}
```

# The Jupyter Project

The Jupyter Project was announced at SciPy in July 2014.

The **Big Split** was finalized in 2015 with the release of **Jupyter 1.0** and **IPython 4.0**.



A repackaging / renaming of the formerly monolithic IPython code:

- IPython - one of many kernels
- Jupyter - notebook front-end
- Facilitates use of web technologies in the notebook

# Jupyter Demo !

OK, let's go for a tour ...



# Installing Jupyter

For a native installation

- Recommended: install Jupyter using the Anaconda distribution [Continuum Analytics]  
This provides the IPython kernel only.
- or pip install jupyter

You can then customize your config and install

- Kernels for other languages/environments, e.g. Anaconda's R-Essentials package
- Extension widgets
- Notebooks !!

Or

- Alternatively, several docker images are available on DockerHub (e.g. [jupyter/demo](https://hub.docker.com/r/jupyter/docker-stacks))
- or you can try out [try.jupyter.org](https://try.jupyter.org)



# Kernels & Extensions





# Currently ~ 51 Kernels

[Kernels Page]

A Kernel represents an execution profile - typically a language but could be combined with other capabilities such as Python + Apache Spark

IJulia	IHaskell	IFSharp	IRuby	IGo
IScala	IMathics	IAldor	Calico Pro	LuaJIT/Tor
Lua	Simple exa	IErlang	IElixir	IOCaml
IForth	IPerl	IPerl6	IPHP	IOctave
IScilab	IMatlab	Bash	Clojure	Hy
Redis	jove, a ke	Ijavascrip	Calysto Sc	Calysto Pr
idl_kernel	Mochi	Lua (used	Spark	Skulpt Pyt
MetaKernel	MetaKernel	Brython	IVisual VP	Brainfuck
KDB+/Q Ker	ICryptol	C++ (cling	Xonsh	Prolog
cl-jupyter	Maxima-Jup	Calysto LC	Java 9	





# Extension Widgets

Notebook Extensions [\[Extensions page\]](#)

Many customizations and extensions available,  
some of my favourites:

- **RISE** - these slides are running under Jupyter
- `nbgrader` - creation/grading of classroom assignments
- `plotchecker` - grading of plots
- `metakernels`
- `hide_code` - hide code cells
- `clicker` - anonymous polls

Generally installable via pip or from github repo

# The Ecosystem & Future Projects



# What is Jupyter being used for ?

Professionals (science, data, financials,...)

- exploratory analyses, collaboration, publishing

Publishers

- books, blogs, reports, theses
- executable books, articles
  - e.g. **Thebe** (O'Reilly) simplified notebook interface
  - 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



# MOOCs, e.g. F.U.N.



Inria: 41001502 Python : des fondamentaux à l'utilisation du langage

mjbright

Courseware

Info Cours

Sommaire

Discussion

Progression

Semaine 1 : Introduction et prise en main

Présentation de la semaine et Evaluation du cours

1. Organisation du MOOC

2. Les outils de la distribution standard Python

3. Les notebooks

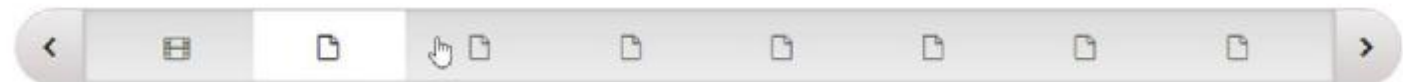
4. Pourquoi Python ?

Matériel du cours

Questionnaire

Semaine 2 : Types de base

Semaine 3 : Références



"Notebooks" IPython v2.0

File

Insert

Cell

Kernel



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.

Vous pouvez naturellement modifier ces cellules de code pour faire des essais; ainsi vous pouvez vous servir du modèle ci-dessous pour calculer la racine carrée de 3, ou essayer la fonction sur un nombre négatif et voir comment est signalée l'erreur:

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

```
Out[1]: 1.4142135623730951
```

On peut également évaluer tout le notebook en une seule fois en utilisant le menu *Cell -> Run All*



# Jupyter Ecosystem

The "*Big Split*" facilitates innovation around the platform

There are many ongoing projects

- static notebook viewers:
  - [IPython Gallery](#) --> [nbviewer](#), [github](#)
- dynamic notebook viewers:
  - [tryjupyter.org](#), [JupyterHub](#), [Bindr](#), [Quantopian](#)
- alternative front-ends:
  - [Sidecar](#), [Rodeo IDE](#), [Hydrogen \(ATOM\)](#), [EIN \(emacs\)](#)
- tools
  - [nbconvert](#) (convert between .ipynb and other formats: PDF, html, ...)
- kernels & extensions



# Jupyter Core Projects

Basic UI enhancements: e.g. shortcuts, cell marking

Notebook tests

- unit testing
- regression testing
- dependency testing

Currently *incubating sub-projects* (IBM, MSFT involvement)

- *Content Management*
  - Search, include, modularity, ToC ex
- *Dashboards*
  - Different notebook layouts, bundling as web apps ex
- *SparkMagic*
  - Connecting to external Spark Engines
- *Declarative Widgets*
  - Widgets with less code ex
- *Kernel Gateway*
  - Enable more deployment scenarios at scale





# IPython / Jupyter Books

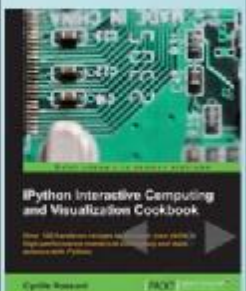
**The Jupyter GitBook**  
*Extension writing*



**Learning IPython for Interactive Computing & Data Visualization**  
Cyrille Rossant  
*Introductory usage*







**IPython Interactive Computing & Visualization Cookbook**  
Cyrille Rossant  
*Advanced usage*



Jupyter is a fast-moving Open Source project  
widely used for Data Analysis  
with a vibrant ecosystem ...

## Questions ?



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