From the lab to Jupyter
A brief history of computational notebooks

Émilien Schultz (CREST - IP Paris)

2024-02-03
What I want to do

- A too short history of Jupyter’s notebooks
- To plea for a better knowledge of scientific softwares’ history.
Where are (our) stories of scientific software?

I mean, outside the open research devroom...

Software is everywhere in research (OS and OS):
- bespoken code
- small packages
- dedicated softwares
- international stars

When social sciences look (rarely) at scientific software

- **Specific dynamics**
  - open-ended & uncertain aims
  - researchers are (usually) non-specialist developers
  - effects of funding constraints

- **Specific consequences**
  - code brittleness
  - intertwined with scientific activity
  - specialization of some researchers

- **Led to specific software journeys**

---

Taking a step back: a lot of open questions

- How to tell the stories of scientific software?
  - What are the different software journeys, especially in OS?
  - What are the steps composing such journeys and condition of evolution?
- How much are intertwined the histories of open source & academia?
  - relation between open source & open science?
  - relation between academics and software engineer?

Two Bits: The Cultural Significance of Free Software, Kelty, 2008, p.132

*The fact that UNIX spread first to university computer-science departments, and not to businesses government, or nongovernmental organizations, meant that it also became part of the core pedagogical practices of a generation of programmers and computer scientists [...]*

---

2Woolston, Chris. "Why science needs more research software engineers." Nature (2022)
Put this idea to work with the case of Jupyter’s notebooks

Innovation from research evolving to a worldwide infrastructure of data science

- Released in 2012, spreaded everywhere
  - ACM Award in 2017
- A perfect viewpoint\(^a\) of:
  - emergence
  - incremental abstraction
  - diffusion within and outside

\(^a\)Long version of this history in French
Caveat: not here to advocate about Jupyter notebooks

Postulate: you know what Jupyter’s computational notebooks are, and you have your opinion about them.

\[ I = \text{ipython and Jupyter user} \]

Émilien Schultz (CREST - IP Paris)

From the lab to Jupyter

2024-02-03
Oversight of the history

First a PhD student, then IPython, next notebooks, and finally Jupyter

![ipython-dev mailing list activity graph](image)
Let’s dive in. First, the context early 2000:

- 90’s Free software achievements
- Litterate programming paradigm is around (Knuth, 1984)
- Specialized interactive scientific softwares (Maple, 1992, Mathematica, 1988)
- A nascent scientific Python community
  - First SciPy workshop in 2002

The open source community keep track of some element of its history, i.e. **William Horton**

---

Émilien Schultz (CREST - IP Paris)  
From the lab to Jupyter  
2024-02-03
Jupyter started as a small hack

- Fernando Pérez started IPython in 2001
  - during his PhD in particles physics
  - "a simple personal fix for a problem in my own workflow"
- Grounded in his researcher’s common sense
  - A taste for interactivity in scientific programming

If you typed Python in the command line, you got a, an interactive shell, it was a very, very primitive and it didn't allow me to do the kinds of things that were very natural in interactive scientific workflows with tools like IDL or Mathematica that I used heavily or Matlab or Maple that other used which was simply to type a bit of code, see the results right there, open a plot, look at the files on, on the file system, et cetera.
The SciPy community as a amplifier

- Enthusiastic reception
  - Enthought hosted IPython on its webpage
- Increasing support
  - Feedback and contributors
  - Brian Granger, Fernando’s friend, jumped in 2004
- Securing the financial possibility to carry on
  - Possibility of a postdoc
  - Support of a team leader in Berkeley (moved there in 2008)

Fernando Pérez, 2021

Very importantly, though, there were people at UC Berkeley early on who supported me when I was still a postdoctoral scholar in Colorado doing more traditional applied mathematics research with Python tools. I wasn’t invested enough in the purely applied mathematics community to make a career just out of that. People at UC Berkeley that I connected to because of the Python community offered me a team and, eventually, a job.
Led IPython to be grounded in academia

<table>
<thead>
<tr>
<th>members</th>
<th>arrival</th>
<th>phd</th>
<th>discipline</th>
<th>position in 2020</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fernando Perez</td>
<td>2001</td>
<td>yes</td>
<td>physics</td>
<td>academic</td>
</tr>
<tr>
<td>Brian Granger</td>
<td>2004</td>
<td>yes</td>
<td>physics</td>
<td>academic</td>
</tr>
<tr>
<td>Benjamin Ragan-Kelley</td>
<td>2004</td>
<td>yes</td>
<td>engineering physics</td>
<td>academic</td>
</tr>
<tr>
<td>Matthias Bussonier</td>
<td>2012</td>
<td>yes</td>
<td>physics</td>
<td>software engineer</td>
</tr>
<tr>
<td>Jess Hamrick</td>
<td>2013</td>
<td>yes</td>
<td>psychology</td>
<td>private research scientist</td>
</tr>
<tr>
<td>Thomas Kluyver</td>
<td>2010</td>
<td>yes</td>
<td>plant science</td>
<td>academic</td>
</tr>
<tr>
<td>Jonathan Frederic</td>
<td>2013</td>
<td>no</td>
<td>physics</td>
<td>software engineer</td>
</tr>
<tr>
<td>Kyle Kelley</td>
<td>2012</td>
<td>no</td>
<td>computer science</td>
<td>software engineer</td>
</tr>
<tr>
<td>Jason Grout</td>
<td>2011</td>
<td>yes</td>
<td>mathematics</td>
<td>academic/software engineer</td>
</tr>
<tr>
<td>Sylvain Corlay</td>
<td>2014</td>
<td>yes</td>
<td>mathematics</td>
<td>ceo/software engineering</td>
</tr>
<tr>
<td>Paul Ivanov</td>
<td>2010</td>
<td>yes</td>
<td>computational neuroscience</td>
<td>software engineer</td>
</tr>
</tbody>
</table>
Notebooks were (just) a late feature of IPython (2012)

- Growth of the IPython project
  - Adding new features
  - New core contributors joining the project
  - Early private investment (Google, Bloomberg, etc.)
- 5 failed attempts of notebooks between 2005 and 2011
  - Some tech not available (web sockets)

Fernando Pérez, 2012

On December 21 2011, we released IPython 0.12 after an intense 4 1/2 months of development. Along with a number of new features and bug fixes, the main highlight of this release is our new browser-based interactive notebook: an environment that retains all the features of the familiar console-based IPython but provides a cell-based execution workflow and can contain not only code but any element a modern browser can display [...] For the IPython project this was a major milestone, as we had wanted for years to have such a system, and it has generated a fair amount of interest online.
Ten computer codes that transformed science

From Fortran to arXiv.org, these advances in programming and platforms sent biology, climate science and physics into warp speed.
Evolution to a broader open source interactive scientific computing framework grounded on open science values

- The effect of two dynamics:
  - Abstracting from the Python community
    - “Those languages are not enemies, the enemies are closed science”
  - Strengthening software engineering good practices
- Backed with a 6 millions dollars grant
  - Helmsley Trust, Gordon and Betty Moore & Alfred P. Sloan Foundation

we have quite ambitious plans for the future [...] Project Jupyter’s mission is to create open source tools for interactive scientific computing and data science in research, education and industry, with an emphasis on usability, collaboration and reproducibility
Jupyter’s community overflowed academia

- Worldwide adoption
  - notebooks became standard of data science
  - Integration in a diversity of services (Colab...) and third-party (VS Code ...)
- But still a strong kernel of academic users and contributors

Fernando Pérez, 2021

This community is not accidental: the core Jupyter team has invested significant effort into welcoming new contributors, helping users, planning and running community events (Jupyter CommunityWorkshops24, JupyterDays and JupyterCon25), and training and mentoring junior developers and designers.
Let’s stop here in this work in progress

A lot of open questions:

- Who are the current users of computationnal notebooks?
- How are researchers using them?
- How did the Jupyter ecosystem expanded?
- ...

And, at this point, does Jupyter still is a scientific software?
Wrapping up: (historical) documentation is important

Scientific softwares are crucial research instruments.

- Need to take their dynamics seriously
- Of course, computational notebooks
  - indicative of scientific programming evolutions
  - ongoing project NOOS\textsuperscript{5}
  - a Github repo for collective archive
- But also all the others
  - Rstudio
  - Gephi
  - ...

\textsuperscript{5}Célya Gruson-Daniel, Mariannig Le Béchec, Clémence Lascombes