

IPython 2013 Progress Report - Sloan Foundation

January 27, 2014

Submitted by: Fernando Perez and Brian Granger.

In December 2012, IPython was awarded a [\\$1.15 million grant](#) from the Alfred P. Sloan Foundation, slated to fund the core team to make important improvements to the Notebook as well as general IPython development for the 2013-2014 period. This is a report of the progress from this effort for 2013.

1 Activities on our 2013 roadmap

At the beginning of the funding period, we established two main items to focus our 2013 efforts on two main goals; work on both of them has proceeded according to plan:

1.1 IPython 1.0 release

Our aim was to release version 1.0 in summer 2013, with the integration of the notebook conversion pipeline (nbconvert).

IPython 1.0 was released in August 2013 and included a fully integrated version of the `IPython.nbconvert` module that is exposed to the user via a new subcommand, `ipython nbconvert`. This command can be used to convert Notebooks into HTML, LaTeX, PDF, Javascript-controlled slideshows for presentations, and more. The system provides sensible out-of-the-box defaults and can be further customized by users with templates that expose full control over the details of the output.

Work has continued on polishing and refining this part of IPython throughout the fall, as we have realized there are still a number of important and subtle issues that require further thought. But we consider the current design and implementation to be fundamentally correct and successful, and a solid base for future improvements.

1.2 IPython 2.0 release

For IPython 2.0, we aimed for a December 2013 release with an implementation of protocols and libraries for interactive communication and control of IPython kernels via Javascript-based widgets on the Notebook client (web browser).

Our current plan is to release IPython 2.0 in February 2014. This is a slight delay from our original December target, but not a problematic one: we are overall very satisfied with the architecture we have in place, it simply proved to be a substantial and challenging problem that took a bit longer to solve than expected.

As of this writing, we have implementations of the communications architecture between kernels and the notebook frontend for interactive controls, Javascript support for interactive widgets

and Python implementations of basic widgets for common tasks. This code is in the final stages of review before being merged. Similarly to what happened with `nbconvert`, we expect to continue refining and improving this part of the project throughout 2014.

2 Developer meetings

An important item in our budget was support for holding week-long, all-hands development meetings where the entire team would have a chance to perform intensive, focused work in a single location.

In January 2013, we held a small-scale meeting with only a handful of developers (the timescale was too tight to plan a full one, as the grant had just been awarded). This meeting served to plan our work items for Spring 2013, which led to the successful 1.0 release in the summer.

The first meeting with the entire team and external invited members took place in July 2013, including developers from Europe, Canada and the US. In addition to core IPython team members, we had participation from members of industry and from teams at Lawrence Berkeley National Laboratory that are using IPython extensively. It was a productive use of resources; this meeting provided the necessary space for finalizing plans for the 1.0 release as well as the detailed design discussions for our interactive Javascript widget architecture.

Our second developer meeting took place January 13-17 2014 and was equally productive. We finalized the plans for our upcoming 2.0 release and drafted the design for a first implementation of the Multiuser Notebook Server which we plan to release in the summer of 2014.

In addition to our focused development activity, at each developer meeting we have held an “open house” event meant to engage the local Bay Area community around IPython. For 2 hours we provide snacks and an opportunity for anyone to come and learn about IPython, meet the core development team, ask questions, etc. We have had very positive response to both Open Houses and find it a valuable way to reach out in person to our local user base. We plan on continuing to hold one of these at each of our future meetings.

Finally, we have adopted a policy of holding a regular, weekly development meeting as a publicly visible and archived Google+ Hangout. This makes our development process more transparent to the community and enables interested third parties to easily ask us direct questions. On occasion, we invite external participants to directly join the video conference when we need to discuss a specific topic in-depth with them. We maintain [a public record \(2013\) \(2014\)](#) of the discussions and videos from these meetings.

These public meetings have been very well received by the community, and in fact other projects such as Matplotlib and YT (a data analysis platform for astrophysics) have since adopted a similar practice.

3 Talks, seminars and other outreach

Our team has been busy engaging multiple communities, from academia to industry, by delivering talks at domain-specific conferences, seminars, lecture series and tutorials. While time-consuming and resource-intensive, we feel this is a valuable way of reaching new users who may benefit from the outcomes of this project.

The following is a summary of our activities on this front:

- F. Perez, *Keynote*, NIPS Workshop on Machine Learning Open Source Software, Lake Tahoe, NV, December 2013.

- F. Perez, *invited talk*, Worskshop on open source tools in hydrology, AGU Fall Meeting, San Francisco, December 2013.
- B. E. Granger, *IPython: the attributes of software and how they affect our work*, keynote at PyData NYC 2013, New York, NY, November 2013.
- F. Perez, *invited seminar*, Stat 157 Reproducible Research, UC Berkeley, November 2013.
- F. Perez, *Department Colloquium*, Physics Dept., University of San Francisco, November 2013.
- F. Perez, *Keynote*, CIT'2013, Medellín, Colombia, October 2013.
- F. Perez, *Python: performance and parallelism*, invited seminar, CS294 Modern Parallel Languages, UC Berkeley, October 2013.
- F. Perez, *IPython: A unified computational architecture for interactivity, publication and HPC.*, invited seminar, CS Division, Lawrence Berkeley National Laboratory, September 2013.
- B. Ragan-Kelley, *IPython is not just about Python anymore*, RuPy Conference, Budapest, August 2013.
- B. Ragan-Kelley, *IPython parallel tutorial*, Blue Brain Project, EPFL, Lausanne, August 2013.
- B. Ragan-Kelley and M. Bussonnier, *IPython Advanced Tutorial*, EuroSciPy 2013, Brussels, August 2013.
- B. E. Granger, *IPython: encouraging open, exploratory, collaborative and reproducible scientific computing*, invited talk at Code and Data Interoperability Workshop, NSF Sustainable Software for Chemistry and Materials, Virginia Tech, VA, July 2013.
- F. Perez, *Keynote*, SciPy Conference, Austin, Tx, June 2013.
- B. E. Granger, *Why you should write buggy software with as few features as possible*, SciPy Conference, Austin, TX, June 2013.
- B. E. Granger and F. Perez, *IPython in depth*, tutorial at SciPy 2013, Austin, TX, June 2013.
- F. Perez, *Python: Can a "slow" language survive in the exascale playground?*, invited talk at 2013 DEGAS Retreat (LBL/UC Berkeley), Santa Cruz, June 2013.
- B. E. Granger, *The IPython Notebook*, invited talk at Workshop on Software Infrastructure for Reproducibility in Science, NYU, NY, May 2013.
- F. Perez, *invited seminar*, eScience Institute, University of Washington, Seattle, May 2013.
- F. Perez, *IPython: a tool for the lifecycle of computational ideas*, invited talk at the 2013 UC Davis Statistical Sciences Symposium, April 2013.
- F. Perez, *invited seminar*, ISchool course on "Working with Open Data", UC Berkeley, April 2014.
- F. Perez, *invited seminar*, Stanford University, March 2013.
- F. Perez, *invited seminar*, Harvard Center for Astrophysics, Cambridge, March 2013.

- F. Perez, *invited talk for FSF awards ceremony*, Libre Planet 2013, Cambridge, March 2013.
- F. Perez, *Keynote*, PyData Silicon Valley, Santa Clara, March 2013.
- B. E. Granger, *The IPython Notebook: a comprehensive tool for data science*, Strata 2013, Santa Clara, CA, March 2013.
- B. Ragan-Kelley, *IPython parallel tutorial*, PyData Silicon Valley, March 2013.
- F. Perez, *invited seminar*, McGovern Institute for Brain Science, MIT, Cambridge, March 2013.
- F. Perez, *IPython: a tool for the lifecycle of computational ideas*, SIAM CSE'13, Boston, March 2013.
- F. Perez, B. E. Granger and B. Ragan-Kelley, *IPython in depth: high productivity interactive and parallel Python*, PyCon 2013, Santa Clara, CA, March 2013.
- F. Perez, *IPython: modern tools for interactive & web-enabled scientific computing*, invited talk, NERSC user day, Lawrence Berkeley National Laboratory, February 2013.
- F. Perez, *Openness, Reproducibility, Interactivity: a Biased View on the Relation between Science and Computing*, Reproducible Research Seminar, UC Berkeley, February 2013.
- F. Perez, *Lecture series for the 2013 Winter School in eScience, Reproducible Science And Modern Scientific Software*, held in Geilo, Norway, January 2013.

4 Papers and other publications

The following are publications that are either directly related to IPython, or that used IPython extensively in the underlying research and are accompanied by reproducible, publicly available notebooks:

- C. Ahrens, J. Nealy, F. Pérez, S. van der Walt, *Sparse Reproducing Kernels for Modeling Fiber Crossings in Diffusion Weighted Imaging*, ISBI 2013, Salt Lake City, March, 2013.
- R. S. Blumenfeld, D. P. Bliss, F. Perez, M. D'Esposito. *CoCoTools: Open-source software for building connectomes using the CoCoMac anatomical database*. *J. Cog. Neuro*, Oct. 2013 (in press) doi:10.1162/jocn_a.00498.
- B. Ragan-Kelley, W.A. Walters, D. McDonald, J. Riley, B.E. Granger, R. Knight, F. Perez and J. G. Caporaso. *Collaborative cloud-enabled tools allow rapid, reproducible biological insights*. *ISME Journal* 7, 461–464 (2013). doi:10.1038/ismej.2012.123.
- K. J. Millmand and F. Perez, *Developing open source scientific practice*, book chapter in “Reproducible research in practice”, ed. V. Stodden, CRC Press (in press).
- F. Perez, book chapter for *Mathematics as a laboratory tool*, Springer Verlag, 2013. Main author and editor: J. Milton. In press.

We note that these are papers where (some of) the authors are directly funded under the current Sloan grant; below we describe other publications that *use* IPython to enhance reproducibility but that may have been authored completely by scientists outside our team.

5 Broader impacts

5.1 Kernels in other languages: IJulia, IHaskell, IFSharp, ...

The protocols that IPython defines for communicating between kernels and clients are language-agnostic, and we envision the IPython architecture as one that enables a unified approach to computational science independent of the programming language the user prefers. We were able to put this idea to the test by collaborating with the team from MIT that develops the [Julia programming language](#), a novel language for numerical computing and data science that has drawn a lot of interest from the computer science, applied mathematics and computational statistics communities.

After initial prototyping of basic IPython-Julia integration during the spring of 2013, we invited three core Julia developers to our summer development meeting. By providing them with technical assistance from our team, the Julia developers were able to implement in a few days a fully working Julia kernel, that allows them to use all IPython clients (terminal, Qt Console and Notebook).

This kernel has now been adopted by the Julia community, enabling them to share computational research done in Julia in the format of IPython notebooks and leveraging all of our toolchain. The PI of the Julia team, Alan Edelman, taught the Fall 2013 [course in Numerical Linear Algebra at MIT](#) using this system, by running an IPython Notebook server with a Julia kernel for all his students, so they could work over the web without dealing with Julia installation issues.

A number of kernels for other languages are now being developed, in various stages of maturity. This validates our idea that our protocol can enable the sharing and reuse of computational research and improve communication between communities by having a common environment (IPython) and format (Notebook files). We are currently aware of kernels for the following languages:

- [IJulia](#)
- [IHaskell](#)
- [IFSharp](#)
- [IRuby](#)
- [IGo](#)
- [IScala](#)
- [IMathics](#)
- [IAldor](#)

5.2 NBViewer as a sharing platform

We operate the [IPython Notebook Viewer](#) as a web service that allows users to share a link to any existing notebook available on the public internet, so it can be viewed by others as a rendered HTML page without having to install any software. This enables authors who write notebooks to share their work with any colleagues by simply sending them a link the recipients can click on. This link shows a read-only version of the notebook in a web browser, but also provides an immediate download link, in case the recipient wishes to download the full notebook for local execution.

NBViewer works by wrapping the `nbconvert` notebook conversion libraries into an easy-to-use, zero-installation web service that enables very easy sharing of notebooks.

While we expected this service to be useful, we have been surprised by how popular and widely used it has become. These are the number of visit per month to the site, according to Google Analytics (between March 2013 and January 2014):

38171	March
44729	April
74218	May
74125	June
74689	July
81298	August
120310	September
125903	October
125361	November
122802	December
276812	January

A lot of traffic is generated by users tweeting links to interesting work they want to share with the public or posting on other social media channels (in addition to any sharing that may happen over email and other private channels). On occasion, we see large traffic spikes when particularly popular content is shared; recently a [notebook](#) written by Google's director of research, Peter Norvig, generated over 100,000 visits in two days.

The nbviewer service was hosted on the Heroku system for most of 2013, but we started encountering scalability problems with that setup. In late 2013, the hosting company [Rackspace](#) graciously offered us unlimited hosting services (CPU and bandwidth) for IPython, and we were able to transition nbviewer to a Rackspace server. This new setup has been able to withsdand even traffic peaks like the one mentioned above without any problems. We expect to continue using this service for the foreseeable future, which both saves us money and provides very robust performance and resilience.

5.3 Notebook-based technical blogging

Today, a lot of technical communication happens on blogs, both from academic researchers and otherwise. In 2013, we saw a number of bloggers adopt the IPython Notebook as a format for authoring blog posts. Once written, these notebooks can be converted into the format of their blogging platform of choice for final publication (typically HTML with some custom CSS). This enables blog authors to offer their readers a post that can contain code as well as narrative and results, but which readers can directly download in its original executable form if they wish to further explore the topic in detail. While authors could already share content as explained above via nbviewer, integration with blogging platforms enables them to include this format in the discussion that happens on blogs via the author's comments, as well as mixing them with other posts they may have authored in a traditional form.

One of the most prolific such authors is UW's Jake van der Plas (director of research for physical sciences at the [eScience Institute](#)), whose [blog](#) contains multiple examples of this idea. He also authored a plugin to automatically integrate notebook-authored posts into the Pelican blogging engine.

These are links to posts explaining how to integrate the IPython notebook into various blogging workflows:

- [Blogging With IPython in Blogger](#), also available in [blog post form](#), [full repo here](#). By Fernando Perez.
- [Blogging With IPython in Octopress](#), by Jake van der Plas and available as a [blog post](#).
- [Blogging With IPython in Nikola](#), also available in [blog post form](#) by Damián Avila.

5.4 Reproducible academic publications

We have been very pleased to see the adoption of the IPython notebook as a companion to academic papers to enable better reproducibility. This section contains academic papers that have been published in the peer-reviewed literature or pre-print sites such as the [ArXiv](#) that include one or more notebooks that enable (even if only partially) readers to reproduce the results of the publication.

- [powerlaw: a Python package for analysis of heavy-tailed distributions](#), by J. Alstott et al.. [Notebook of examples in manuscript](#), [ArXiv link](#) and [project repository](#).
- [Collaborative cloud-enabled tools allow rapid, reproducible biological insights](#), by B. Ragan-Kelley et al.. The [main notebook](#), the [full collection of related notebooks](#) and the [companion site](#) with the Amazon AMI information for reproducing the full paper.
- [A Reference-Free Algorithm for Computational Normalization of Shotgun Sequencing Data](#), by C.T. Brown et al.. [Full notebook](#), [ArXiv link](#) and [project repository](#).
- [The kinematics of the Local Group in a cosmological context](#) by J.E. Forero-Romero et al.. The [Full notebook](#) and also all the data in a [github repo](#).
- [Warming Ocean Threatens Sea Life](#), an article in Scientific American [backed by a notebook for its main plot](#). By Roberto de Almeida from [MarinExplore](#).
- [Extrapolating Weak Selection in Evolutionary Games](#), by Wu, García, Hauert and Traulsen. [PLOS Comp Bio paper](#) and [Figshare link](#).
- [Using neural networks to estimate redshift distributions. An application to CFHTLenS](#) by Christopher Bonnett [paper](#)(submitted to MNRAS)
- [Mechanisms for stable, robust, and adaptive development of orientation maps in the primary visual cortex](#) by Jean-Luc R. Stevens, Judith S. Law, Jan Antolik, and James A. Bednar. [Journal of Neuroscience](#), 33:15747-15766, 2013. [Notebook1](#), [Notebook2](#).

5.5 Executable books

As the IPython notebook becomes more widely adopted, in addition to informal blogging and academic papers, in 2013 several books were published that used IPython notebooks as a central element of the book's workflow:

- [Python for Signal Processing](#), by Jose Unpingco, was written entirely as a collection of notebooks. [Published by Springer Verlag](#) as a hard-cover book, the notebooks are also available in a [GitHub repository](#). The author wrote each chapter as a notebook and published it [as a blog post](#), when the book was completed, it was exported into a printable format for final typesetting.

- **Mining the Social Web**, by [Matthew Russell](#). Published by O'Reilly, in this book all code examples were developed as notebooks that are available in a [companion GitHub repository](#). Additionally, the author regularly teaches hands-on workshops based on the book where all work is done with the IPython notebook.
- **Probabilistic Programming and Bayesian Methods for Hackers Using Python and PyMC**, by [Cameron Davidson-Pilon](#). This book has not been yet published in printed format, but it is entirely available as a freely accessible [GitHub repository](#).

In addition to these books that use IPython Notebook for their subject matter, two additional books were published in 2013 that directly address IPython:

- **Learning IPython for Interactive Computing and Data Visualization**, by Cyrille Rossant, Packt Publishing, 2013. [This book](#) is entirely dedicated to IPython, and all of its code examples are available as notebooks in a [public repository on GitHub](#).
- **Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython**, by Wes McKinney, O'Reilly Media, October 2012. This book has an extensive chapter devoted to IPython, and uses IPython throughout the book as the environment for showcasing all code examples.

5.6 Courses using IPython as a central element

We have received anecdotal evidence via private correspondence of educators enthusiastically adopting the IPython Notebook and other tools as part of their educational workflow. For example, B. Granger's colleague at CalPoly's physics department, Jennifer Klays, teaches an undergraduate Computational Physics course that uses IPython extensively.

The following list shows courses that have publicly available information (class repositories, student projects, etc) where IPython is now a central element of the class:

- [Python for Data Science](#), a UC Berkeley taught by Josh Bloom (F. Perez and other members of the IPython team guest lecture in this course).
- [Scientific Python Bootcamp](#), a UC Berkeley bootcamp co-organized by Josh Bloom and Fernando Perez, with participation from other members of the IPython team.
- [Statistics 157: Reproducible and Collaborative Data Science](#), an undergraduate statistics course at UC Berkeley taught by Philip Stark and Aaron Culich.
- [Statistics 222: Statistics Master's Capstone](#), UC Berkeley graduate stats course taught by Philip Stark and D. Goldsmith.
- [Urban Informatics And Visualization](#), a UC Berkeley course by Paul Waddell.
- [EE 123: Digital Signal Processing](#), a UC Berkeley course taught by Miki Lustig.
- [Working with open data](#), a course at the UC Berkeley ISchool taught by Raymond Yee.
- [Data Science](#), a Harvard course taught by Hanspeter Pfister and Joe Blitzstein.
- [Practical Data Science](#), a course taught by professor Josh Attenberg at NYU.
- [A collection of IPython notebooks used to teach programming to high-schoolers](#), by Eric Matthes. Used at a high school in Anchorage, Alaska.

5.7 Academic research projects using IPython as infrastructure

The following is a (likely incomplete) list of projects that have made IPython an important element of their infrastructure, and for which we have publicly available references:

- The [DOE KBase](#) project for predictive biology has adopted the IPython notebook as a central architectural element of their scientist-facing interface for ‘computational narratives’. We have had several meetings with their development team, providing them with technical details about the internal design of IPython; we foresee a much closer collaboration with this project moving forward. KBase is a \$50M, large-scale effort sponsored by the DOE for improving our predictive capabilities in computational biology, led by professor Adam Arkin from UC Berkeley and Lawrence Berkeley National Laboratory.
- A project of the [Tetherless World Constellation](#), funded by NOAA and the NSF, uses the IPython Notebook as an interface to enable open, reproducible workflows in oceanography research. This work has been presented at [AGU’12](#) and [AGU’13](#).
- The Salish Sea MEOPAR project is lead by [Prof. Susan Allen](#) of the Department of Earth, Ocean, and Atmospheric Sciences at the University of British Columbia. It is part of the Canadian [Marine Environmental Observation, Prediction, And Response \(MEOPAR\) network](#) which is funded by the National Science and Engineering Research Council (NSERC). This project uses the IPython notebook as key tools in [configuring and evaluating the NEMO ocean model](#) for the Salish Sea.

5.8 Commercial projects

Since IPython is licensed under a liberal license (BSD) that allows commercial use, we have seen a rise in commercial vendors that incorporate IPython in their products. We see this as a valuable and healthy aspect of our project, as some of these vendors may choose to support IPython in the future (Enthought has long done so, and in 2013 Microsoft donated \$100,000 to the project). This is a list of some projects we are aware of in this category:

- [Microsoft Python tools for Visual Studio](#): a free plugin for VS that embeds a custom graphical console (similar to our Qt console). This console was independently developed by Microsoft and it implements our communications protocol directly.
- IBM Watson: the team that trains Watson for domain-specific applications successfully replaced their in-house workflow with a solution that uses `IPython.parallel` and the Notebook. This resulted in improved performance, reduced code complexity and deployment difficulties. They presented their results at the SciPy’13 conference: [video of the talk](#) and [PDF slides](#).
- [Enthought Canopy](#): a rich IDE developed by Enthought corporation, that includes both the IPython Qt console and the notebook.
- [Wakari](#): a web service developed by Continuum Analytics that enables users to share and execute notebooks in the cloud.
- [Sage Math Cloud](#): a startup from the University of Washington lets users run and share Sage and IPython notebooks in the cloud.

- [Graph Lab Notebook](#): another UW startup, this one focused on fast execution of graph algorithms oriented at large-scale data analytics and machine learning, exposes its libraries to their users via a customized IPython Notebook.
- [Sense Platform](#): a cloud-based data analytics environment, exposes IPython as one of its bundled systems (with a custom interface written by them).
- [Koding.com](#): a cloud based programming and analytics environment, offers not only the IPython notebook with Python, but also IJulia. This is the first commercial deployment of IJulia we have seen, a remarkably short time considering the first prototype code for it was written in our development meeting in summer 2013.

5.9 IPython used at the official Python.org website

The next-generation version of the official Python website, available currently at its [preview location](#), offers an [interactive IPython console](#) for users to experiment live with the Python language.

This is both an important seal of approval for IPython as a project, and once the new website goes live and becomes [Python.org](#), it will raise IPython's visibility even further, as every person who visits the official Python website will have the opportunity to immediately play with IPython from their browser, without installing anything.

5.10 Data-driven journalism

We see great potential in the use of the IPython notebook to explore data-intensive issues in journalistic communication with the general public. This is a complex topic that we have [briefly covered in a blog post](#), but we want to highlight two high-profile examples of this already happening:

- Jonathan Corum, the New York Times science graphics editor, [reported](#) having used the IPython Notebook to manage the data behind the complex infographic that explains the [Kepler mission to search for exoplanets](#).
- News developer Jeremy Singer-Vine from the Wall Street Journal developed [Reporter](#), a tool that enables him to create notebooks with code and data and share with the journalists who will ultimately write articles for the general public versions where the computational details are hidden, yet available as needed. This makes it much easier to revisit questions in the underlying analysis, and we hope, will ultimately lead to much more informed discussions in the public sphere where the code and data behind journalistic pieces will be available even to readers for inspection.

5.11 Awards

In March 2013, F. Perez received the [2012 Award for the Advancement of Free Software](#), for the creation and development of IPython and his contributions to the scientific Python ecosystem. The award was presented at [LibrePlanet 2013](#), the annual conference of the FSF, held at the Harvard Science Center in Cambridge, MA. This award is the most significant recognition in the community of Free and Open Source Software, having been previously given to the creators of the Python, Ruby and Perl languages, among others.

6 The IPython Notebook Gallery

In addition to all of the above links, there are many, many more examples of very interesting IPython notebooks covering a wide variety of topics, that have been developed independently by the community. We maintain a (necessarily incomplete) hand-curated list of particularly interesting notebooks in our [gallery](#), that includes everything from multi-notebook tutorials on machine learning or quantum optics to analysis of musical or public health data. We encourage the reader to visit the gallery.

While the gallery was initially created by our team, today it is actively curated by community members (it is part of our wiki, which any user with a GitHub account can edit). We now see more frequent edits from external community members than from our own team.

7 Conclusion

We are very satisfied so far both with our technical progress and with the adoption the project is finding across an extremely wide range of communities, from very technical research to high school education or journalism. We have had minor slippage in our schedule, but overall we feel the project is proceeding according to our plan and successfully achieving its goals so far.

We are grateful to the Sloan Foundation for its continued support, and would happily answer any questions regarding the results summarized in this report.