The SciPy Documentation Project (Technical Overview)

Stefan Johann Van der Walt (stefan@sun.ac.za) – *University of Stellenbosch*, South Africa

This summer saw the first NumPy Documentation Marathon, during which many thousands of lines of documentation were written. In addition, a web framework was developed which allows the community to contribute docstrings in a wiki-like fashion, without needing access to the source repository. The new reference guide, which is based on these contributions, was built using the popular Sphinx tool. While the documentation coverage is now better than ever, there is still a lot of work to be done, and we encourage interested parties to register and contribute further. This paper gives a more detailed overview of the events leading to this project, and of the technical goals achieved.

Motivation and early history

The effort was funded by the University of Central Florida, under management of Dr Joe Harrington. Dr Harrington attempted to use the SciPy tool suite as a basis for his lectures in astronomy last year, but found that students struggled to become proficient with NumPy and SciPy, much more so than with a previous package used. A lack of documentation was identified as the key deficiency, and, in response, this project was started.

Joe hired me in order to write documentation, but we soon realised that the mammoth task of documenting packages as vast as NumPy and SciPy was beyond the means of a single person. We needed to find a way to involve the larger community - to distribute the load of writing and to obtain input from people with a variety of backgrounds.

In March, at the neuro-imaging sprint hosted by NeuroSpin in Paris, the first discussions centred around such a framework took place between Fernando Perez, Gaël Varoquaux and myself. Later that month, at the IPython sprint, Emmanuelle Gouillart volunteered to do the first implementation, and we sketched a rough architecture. The plan was to coerce MoinMoin into serving up docstrings, and propagate changes back into a bzr repository of the NumPy source.

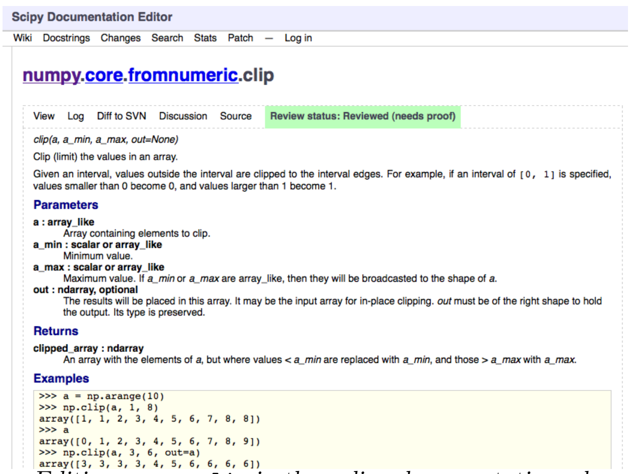
Meanwhile, Pauli Virtanen was hard at work on the same concept, and when he announced his solution to the mailing list, the obvious way forward was to join forces. He and Emmanuelle proceeded to implement the first usable version of the documentation wiki.

Documentation Web-app

The original MoinMoin-based documentation editor that Pauli and Emmanuelle produced provided everything we needed: a way of editing NumPy docstrings in ReStructuredText, and of propagating those changes back to source, albeit with a lot of effort.

Soon, however, it became clear that the MoinMoin-based approach was limited. Adding a feature to the editor often meant crudely hacking it onto a framework written with a different use case in mind. This limited our progress in (and enthusiasm for!) implementing new ideas.

Pauli started working on a web framework in Django, and completed it in record time. The new framework supported all the features of the old one, but, being a web app, allowed us to add things easily such as workflow, merge control, access control, comments and statistics.


Scipy Documentation Editor
Wiki Docstrings Changes Search Stats Patch — Log in
numpy.core.fromnumeric.clip
View Log Diff to SVN Discussion Source Review status: Reviewed (needs proof)

clip(a, a_min, a_max, out=None)

Clip (limit) the values in an array.

Given an interval, values outside the interval are clipped to the interval edges. For example, if an interval of `[0, 1]` is specified, values smaller than 0 become 0, and values larger than 1 become 1.

Parameters

a : array_like
Array containing elements to clip.
a_min : scalar or array_like
Minimum value.
a_max : scalar or array_like
Maximum value. If *a_min* or *a_max* are array_like, then they will be broadcasted to the shape of *a*.
out : ndarray, optional
The results will be placed in this array. It may be the input array for in-place clipping. *out* must be of the right shape to hold the output. Its type is preserved.

Returns

clipped_array : ndarray
An array with the elements of *a*, but where values < *a_min* are replaced with *a_min*, and those > *a_max* with *a_max*.

Examples

```
>>> a = np.arange(10)
>>> np.clip(a, 1, 8)
array([1, 1, 2, 3, 4, 5, 6, 7, 8, 8])
>>> a
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
>>> np.clip(a, 3, 6, out=a)
array([3, 3, 3, 3, 4, 5, 6, 6, 6, 6])
```

Editing `numpy.clip` *in the online documentation editor.*

The source of the web-app is available here:

https://code.launchpad.net/~pauli-virtanen/scipy/pydocweb

Documentation Standard

Even before the online editing infrastructure was completely in place, discussion started on finalising a documentation standard. Since we were going to invest a significant amount of time formatting docstrings, we wanted to do it right from the start. At the time, Jarrod Millman had already written up the format as it stood, and we completed it, aided by feedback from the community.

The documentation standard is tool-agnostic ReStructuredText, and was designed with text terminals in mind. It attempts to balance markup capabilities with ease of readability.

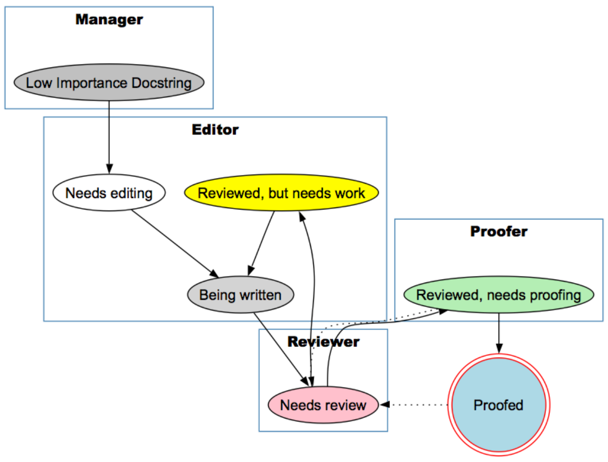
We realised that the new format would not suit all documentation tools, such as Sphinx and Epydoc, and wrote a parser which generates a higher level view of the docstring. It then becomes trivial to output it in any suitable format.

The documentation parser is available at:

https://code.launchpad.net/~stefanv/scipy/numpy-refguide

Workflow

With the new web-app, implementing a workflow became possible:



NumPy Documentation Workflow

The workflow is not rigorously enforced: we focus instead on moving docstrings into the source code as rapidly as possible. The reviewer status was designed to expose the code to more editors and thereby improve the quality of the code, rather than as a mechanism of restricting contribution.

Pauli furthermore implemented three-way merging, which allows us to update docstrings in the web application with those from the Subversion repository. The administrator is given the opportunity to fix any conflicts that may arise. After such an update, edits are made against the latest changes.

Whenever required, a patch can be generated against the latest SVN version of the code. This patch alters our software to contain the latest changes from the wiki, and takes into account special mechanisms for adding docs, such as NumPy's `add_newdocs.py`.

Progress Made

Considering that Dr Harrington funded the documentation drive, the fifty-odd functions used in his classes were given priority. After documenting those, an additional 280 were identified as important targets, of which more than half are now documented.

Writing documentation, particularly examples, leads to a lot of testing, which in turn exposes bugs. The documentation project therefore unexpectedly produced some code fixes to NumPy and Sphinx!

A reference guide of more than 300 pages was generated using the Sphinx tool (which was developed to document Python itself). A Sphinx plugin was written, using the parser mentioned above, to translate the NumPy docstrings to a format ideally suited to Sphinx.

The code used to generate the reference guide can be found either at:

https://code.launchpad.net/~stefanv/scipy/numpy-refguide

or at:

https://code.launchpad.net/~pauli-virtanen/scipy/numpy-refguide

The latest reference guide can be downloaded from:

http://mentat.za.net/numpy/refguide

Conclusion

I would like to thank everyone who contributed to the documentation effort thus far, be it by giving feedback on the mailing list, by coding or by writing documentation. Those of you who wrote more than a thousand words of documentation were rewarded with T-shirt, which is merely a small token of appreciation. In reality, the contribution you have made is an imporant one: significantly lowering a barrier to the adoption of NumPy.

While we have made good progress, there is still a lot left to do! We ask that you join us in writing, reviewing and editing docstrings, or otherwise assist in the ongoing development of the documentation application and tools.