

Sphinx

K. Jarrod Millman
Helen Wills Neuroscience Institute
University of California, Berkeley
`millman@berkeley.edu`

Applied Mathematics Perspectives 2011
Reproducible Research: Tools and Strategies for Scientific Computing
July 13, 2011

Desideratum

A good documentation tool should

- Be easy to use
- Readable in plain text format, *but* produce beautiful HTML & PDF
- Let you use the editors & tools you prefer
- Facilitate collaboration

Unsatisfactory solutions

- \LaTeX
- Lyx
- Word processors
- Wikis

What is reStructuredText?

An intuitive, easy-to-read **lightweight markup syntax** and parser component of Docutils based on existing conventions for marking up plaintext email messages, newsgroup postings, and text documentation such as `README.txt` files.

What is reStructuredText used for?

- in-line program documentation (such as Python docstrings)
- standalone program documentation
- websites
- presentations (slides)
- conference proceedings, scientific articles, and books

Paragraphs

A paragraph is one or more consecutive lines of text separated by blank lines (one is enough). For example:

```
This is a paragraph.  It's quite
short.
```

```
This paragraph will result in an indented
block of text, typically used for quoting
other text.
```

```
This is another one.
```

Text formatting

italics

****bold****

``monotype``

literal or code blocks are indicated with indented text following two colons::

```
import numpy as np
x = np.random.rand(12)
```

Headings

Heading 1

=====

Heading 2

Heading 3

~~~~~

Heading 4

^^^^^

# Lists

Some bullet points:

- \* point A
- \* point B
- \* point C

Some numbered points:

- #. point A
- #. point B
- #. point C

# Simple tables

Simple tables

-----

```
=====  =====  
Heading 1  Heading 2  
=====  =====  
entry(1,1) entry(1,2)  
entry(2,1) entry(2,2)  
=====  =====
```

# Fancy tables

```
+-----+-----+
| Heading 1 | Heading 2 |
+=====+=====+
| entry(1,1) | entry(1,2) |
+-----+-----+
| entry(2,1) | entry(2,2) |
+-----+-----+
```

# reStructuredText tools

rst2html, rst2latex, rst2odt

- <http://docutils.sourceforge.net>

rst2pdf

- <http://code.google.com/p/rst2pdf/>

rst2beamer

- <http://rst2beamer.github.com>

Sphinx

- <http://sphinx.pocoo.org/>

Proceedings

- [http://github.com/scipy/scipy\\_proceedings](http://github.com/scipy/scipy_proceedings)

# Sphinx

- Official documentation tool of python
- And numpy, scipy, ipython, matplotlib, mayavi, ...
- Content is searchable, indexed, and cross-referenced
- Built on top of reStructuredText
- Generates HTML & PDF
- Extensible!

# Narrative documentation

*One of Sphinx' goals is to coax people into writing good docs, and that unfortunately involves writing in many instances :) This is not to say that API docs don't have their value; but when I look at a new library's documentation and only see autogenerated API docs, I'm not feeling encouraged.*

*Georg Brandl, the author of Sphinx*

# Autodoc extension

*BUT ...*

Sphinx' built-in extension, `autodoc`, provides directives to extract and insert Python docstrings into documentation.

# Quickstart

```
$ sphinx-quickstart  
$ make html
```

# Configuration

Let's look at the following files:

```
conf.py
_static/default.css
_static/logo.png
_templates/layout.html
```

# Inline math support

For example,

We know that `:math:`a^2 + b^2 = c^2``.

produces

We know that  $a^2 + b^2 = c^2$ .

# Math display support

For example,

```
.. math::
```

```
(a + b)^2   &=   (a + b) (a + b) \\
             &=   a^2 + 2ab + b^2
```

produces

$$(a + b)^2 = (a + b)(a + b) \tag{1}$$

$$= a^2 + 2ab + b^2 \tag{2}$$

# SciPy extensions

- Autogenerated matplotlib plots
- Syntax-highlighting for IPython sessions

# sampledoc

Fetching the data) and you can grab a harcopy of the the [sampledoc PDF](#)' followed by a bulleted list of links: 'Getting started', 'Installing your doc directory', 'Customizing the look and feel of the site', and 'Sphinx extensions for embedded plots, math and more'."/>

http://matplotlib.sourceforge.net/sampledoc/

# sampledoc

home | search |

## sampledoc tutorial

This is a tutorial introduction to quickly get you up and running with your own sphinx documentation system. We'll cover installing sphinx, customizing the look and feel, using custom extensions for embedding plots, inheritance diagrams, syntax highlighted ipython sessions and more. If you follow along the tutorial, you'll start with nothing and end up with this site - it's the bootstrapping documentation tutorial that writes itself!

The source code for this tutorial lives in mpl svn (see [Fetching the data](#)) and you can grab a harcopy of the the [sampledoc PDF](#)

- [Getting started](#)
  - [Installing your doc directory](#)
- [Customizing the look and feel of the site](#)
- [Sphinx extensions for embedded plots, math and more](#)