
CHUGALUG Sphinx Example

Release 1.0

Doug Hellmann

September 18, 2009

CONTENTS

1	Benefits	1
2	Features	3
2.1	Input is reStructuredText	3
2.2	Output Formats	4
2.3	Footnotes and Cross-References	4
2.4	Automatically Generated Index	4
2.5	Special Handling for Source Code	5
2.6	Extensible	6
2.7	Support	6
3	Using Sphinx	7
3.1	Installation	7
3.2	Getting Started	7
3.3	HTML Output	8
3.4	PDF Output	8
4	A Few Sphinx Users	11
5	Additional Resources	13
	Bibliography	15
	Module Index	17
	Index	19

BENEFITS

- Easier markup syntax than nroff, troff, or LaTeX
- plain text source works with version control
- 1 source document -> multiple outputs
- Extensible

FEATURES

2.1 Input is reStructuredText

2.1.1 Stylistic Tags

- **bold**
- *italics*
- `literal`

2.1.2 Document Structure

- `toctree`
- `include`
- `references`
- multiple heading levels
- `images`
- `lists`
 1. `numbered`
 2. `unnumbered`
- `tables`

Simple:

A	B	A and B
False	False	False
True	False	False
False	True	False
True	True	True

Fancy: Not yet supported for LaTeX

2.1.3 Semantic Markup

- document metadata (author, synopsis, etc.)

- code objects (modules, classes, functions)

2.2 Output Formats

- **HTML**
 - Themes
 - Templates (Jinja2)
 - Built-in *Search Page*
 - Navigation links
- **PDF**
 - via LaTeX, for now
 - config options
 - stylesheet
 - [PDF of this presentation](#)

2.3 Footnotes and Cross-References

- Manually numbered ¹. (`[1]_`)
- Automatically numbered ². (`[#]_`)
- Labeled ³ (`[#label]_`)
- Symbols ⁴ and ⁵ (`[*]`)
- Citations `[PyMOTW]` (`[PyMOTW]_`)
- [Hyperlink-References](#)
- Section cross-reference *Output Formats* (`:ref: 'output-formats '`)

2.4 Automatically Generated Index

- **Insert index markers**
 - single
 - pair
 - triple
- Hyperlinks or cross-references generated in the build
- See upper right corner for link
- *Index*

¹ This is a manually numbered footnote.

² This footnote was numbered automatically.

³ This footnote is labeled.

⁴ This footnote just has a symbol.

⁵ This footnote has a different symbol.

2.5 Special Handling for Source Code

2.5.1 Syntax Highlighting

```
#!/usr/bin/env python

"""Example module.
"""

import os

class MyClass(object):
    """This is a simple class.

    This module illustrates three features of Sphinx:

    1. Pygments integration.
    2. Auto-doc features.
    3. Use of rst in docstrings.
    """

    def __init__(self, arg1):
        """Initialize MyClass instance.

        arg1
            Provide a value for the argument.
        """
        self.arg1 = arg1

    def another_method(self):
        """Returns something.
        """
        return self.arg1 * 2

def main():
    o = MyClass('foo ')
    print o.another_method()
```

2.5.2 Incorporate docstrings

MyClass

```
class MyClass (arg1)
    This is a simple class.

    This module illustrates three features of Sphinx:

    1. Pygments integration.
    2. Auto-doc features.
    3. Use of rst in docstrings.
```

Methods

`another_method()`
Returns something.

`main()`

`main()`

2.5.3 Module Index

- Modules automatically indexed
- *Module Index*

2.6 Extensible

- Write extensions in Python
- **Add custom directives**
 - intersphinx
 - doctest
 - graphviz
 - pngmath and jsmath
- **New output formats**
 - linkcheck
 - rst2pdf

2.7 Support

- **Active developer and user list**
 - <http://groups.google.com/group/sphinx-dev>
- Lots of examples in OSS

USING SPHINX

3.1 Installation

Use `pip` or `easy_install` to install sphinx inside of a virtualenv:

```
$ mkvirtualenv test_sphinx
$ pip install Sphinx
```

The dependencies installed for you include:

- docutils
- Jinja2
- Pygments

3.2 Getting Started

Starting a brand new project is as simple as running the interactive quickstart script:

```
$ sphinx-quickstart
Welcome to the Sphinx quickstart utility.
```

```
Please enter values for the following settings (just press Enter to
accept a default value, if one is given in brackets).
```

```
Enter the root path for documentation.
> Root path for the documentation [.]: example
```

```
You have two options for placing the build directory for Sphinx output.
Either, you use a directory "_build" within the root path, or you separate
"source" and "build" directories within the root path.
> Separate source and build directories (y/N) [n]: y
```

```
Inside the root directory, two more directories will be created; "_templates"
for custom HTML templates and "_static" for custom stylesheets and other static
files. You can enter another prefix (such as ".") to replace the underscore.
> Name prefix for templates and static dir [_]:
```

```
The project name will occur in several places in the built documentation.
> Project name: CHUGALUG Sphinx Example
```

```
> Author name(s): Doug Hellmann
```

Sphinx has the notion of a "version" and a "release" for the software. Each version can have multiple releases. For example, for Python the version is something like 2.5 or 3.0, while the release is something like 2.5.1 or 3.0a1. If you don't need this dual structure, just set both to the same value.

```
> Project version: 1.0
```

```
> Project release [1.0]:
```

The file name suffix for source files. Commonly, this is either ".txt" or ".rst". Only files with this suffix are considered documents.

```
> Source file suffix [.rst]:
```

One document is special in that it is considered the top node of the "contents tree", that is, it is the root of the hierarchical structure of the documents. Normally, this is "index", but if your "index" document is a custom template, you can also set this to another filename.

```
> Name of your master document (without suffix) [index]:
```

Please indicate if you want to use one of the following Sphinx extensions:

```
> autodoc: automatically insert docstrings from modules (y/N) [n]:
```

```
> doctest: automatically test code snippets in doctest blocks (y/N) [n]:
```

```
> intersphinx: link between Sphinx documentation of different projects (y/N) [n]:
```

```
> todo: write "todo" entries that can be shown or hidden on build (y/N) [n]:
```

```
> coverage: checks for documentation coverage (y/N) [n]:
```

```
> pngmath: include math, rendered as PNG images (y/N) [n]:
```

```
> jsmath: include math, rendered in the browser by JSMath (y/N) [n]:
```

Note: pngmath and jsmath cannot be enabled at the same time.

pngmath has been deselected.

```
> ifconfig: conditional inclusion of content based on config values (y/N) [n]:
```

A Makefile and a Windows command file can be generated for you so that you only have to run e.g. 'make html' instead of invoking sphinx-build directly.

```
> Create Makefile? (Y/n) [y]: y
```

```
> Create Windows command file? (Y/n) [y]: n
```

Finished: An initial directory structure has been created.

You should now populate your master file example/source/index.rst and create other documentation source files. Use the Makefile to build the docs, like so:

```
make builder
```

where "builder" is one of the supported builders, e.g. html, latex or linkcheck.

3.3 HTML Output

- All dependencies installed with Sphinx
- make html

3.4 PDF Output

- Core requires LaTeX distro such as <http://texlive.org/>

- `make latex && (cd build/latex; make all-pdf)`
- Straight-to-PDF (rst2pdf) writer in development

A FEW SPHINX USERS

- Python
- PyMOTW
- Sage
- ... many more

ADDITIONAL RESOURCES

Sphinx The Sphinx home page, including the user manual and links to projects that use Sphinx for their documentation.

sphinx-dev Google group for developers and users of Sphinx.

docutils The docutils site includes links to [rst reference guides](#) and other tools for working with rst.

Writing Technical Documentation with Sphinx, Paver, and Cog A post from my blog covering the tool chain I've built up to produce PyMOTW.

Using Sphinx and Doctests to provide Robust Documentation Chris Perkins' presentation from PyCon 2009

The source for this presentation This presentation is available on my website in HTML form. The source is hosted on BitBucket (<http://bitbucket.org/dhellmann/chugalug-sphinx-intro/>).

BIBLIOGRAPHY

[PyMOTW] This is a citation.

MODULE INDEX

E

example, 5

INDEX

A

another_method() (example.MyClass method), 6

C

code

source, 5, 6

cross reference, 4

E

example (module), 5

extensions, 6

F

first

second third, 4

H

HTML, 4

output, 8

I

index, 4

paired term, 4

second term, 4

source code, 6

installation, 7

L

LaTeX, 1, 4, 8

M

main() (in module example), 6

MyClass (class in example), 5

O

output

HTML, 8

PDF, 8

P

paired term

index, 4

PDF, 4

output, 8

R

resources, 13

reStructuredText

footnotes, 4

style, 3

S

second

third, first, 4

source

code, 5, 6

T

third

first second, 4