

---

# **sphinx-better-theme**

***Release 0.1.5***

**Sep 08, 2018**



---

## Contents

---

<b>1</b>	<b>Installation</b>	<b>3</b>
1.1	Read the Docs Configuration . . . . .	3
<b>2</b>	<b>User guide</b>	<b>5</b>
2.1	Theme options . . . . .	5
2.2	Adding static files . . . . .	6
2.3	Using CSS files . . . . .	6
2.4	Using Javascript files . . . . .	7
2.5	Suggested Sphinx options . . . . .	7
<b>3</b>	<b>Demos</b>	<b>9</b>
3.1	Sphinx style demo . . . . .	9
3.2	reStructuredText style demo . . . . .	12
<b>4</b>	<b>Why?</b>	<b>17</b>
4.1	It looks nice . . . . .	17
4.2	It's easy to style with CSS . . . . .	17
4.3	The navigation is laid out better . . . . .	18
4.4	It works well on small screens and mobile devices . . . . .	18
4.5	Deficiencies . . . . .	18
<b>5</b>	<b>Compatibility</b>	<b>19</b>
<b>6</b>	<b>Projects using sphinx-better-theme</b>	<b>21</b>
<b>7</b>	<b>History/Roadmap</b>	<b>23</b>
<b>8</b>	<b>Other themes to check out</b>	<b>25</b>
	<b>Bibliography</b>	<b>27</b>



**sphinx-better-theme** is a theme for [Sphinx](#) that looks nice, is easy to style with CSS, works well on small screens and mobile devices, and organizes the page better. See [Why?](#) for details.

You can get the source and open issues [on Github](#).



# CHAPTER 1

---

## Installation

---

Get it from PyPI:

```
> pip install sphinx-better-theme
```

Or [download the zip file](#) and run the usual command:

```
> python setup.py install
```

Once the package is installed, make these changes to `conf.py` to direct Sphinx to use the theme:

```
from better import better_theme_path
html_theme_path = [better_theme_path]
html_theme = 'better'
```

## 1.1 Read the Docs Configuration

Using sphinx-better-theme with [Read the Docs](#) is easy. You just need to tell it to install the package.

First, create a `requirements.txt` file just for your docs. It should look like this:

```
sphinx-better-theme==0.1.4
# other dependencies for your docs if any
```

I suggest putting it in your docs folder, e.g. at `docs/requirements.txt`.

Then, go to your Read the Docs admin panel. Make sure the *Use virtualenv* checkbox is enabled, and set the *Requirements file* field to the path to your `requirements.txt` file.

Read the Docs should now build and display your theme correctly, assuming your `conf.py` contains the changes described above.





For installation instructions, see *Installation*.

If you get stuck, you can look for information in Sphinx’s documentation for [using a theme](#), but in theory all the relevant information is collected right here. [Open a Github issue](#) if something’s missing.

This document assumes you’ve already set up Sphinx and have some docs written.

sphinx-better-theme is meant to be customized primarily via CSS. There are a few options that you can set in `conf.py`, but they are either functionality-related or appear in more than one annoying-to-type selector.

CSS-based customization is currently limited by the inflexibility of the markup. That situation should improve over time as sphinx-better-theme sheds more and more of its inheritance from the basic theme.

Feel free to read [the `conf.py` for this site](#) to get ideas for your own site. Make sure you also read *Suggested Sphinx options*.

## 2.1 Theme options

Unless you’re creating your own theme that inherits from sphinx-better-theme, you’re probably setting theme options in `conf.py`. Here are all the defaults:

```
html_theme_options = {
    # show sidebar on the right instead of on the left
    'rightsidebar': False,

    # inline CSS to insert into the page if you're too lazy to make a
    # separate file
    'inlinecss': '',

    # CSS files to include after all other CSS files
    # (refer to by relative path from conf.py directory, or link to a
    # remote file)
    'cssfiles': ['_static/my_style.css'], # default is empty list
```

(continues on next page)

(continued from previous page)

```
# show a big text header with the value of html_title
'showheader': True,

# show the breadcrumbs and index/next/previous links at the top of
# the page
'showrelbartop': True,
# same for bottom of the page
'showrelbarbottom': True,

# show the self-serving link in the footer
'linktotheme': True,

# width of the sidebar. page width is determined by a CSS rule.
# I prefer to define things in rem because it scales with the
# global font size rather than pixels or the local font size.
'sidebarwidth': '15rem',

# color of all body text
'textcolor': '#000000',

# color of all headings (<h1> tags); defaults to the value of
# textcolor, which is why it's defined here at all.
'headtextcolor': '',

# color of text in the footer, including links; defaults to the
# value of textcolor
'footertextcolor': '',

# Google Analytics info
'ga_ua': '',
'ga_domain': '',
}
```

## 2.2 Adding static files

(This is all vanilla Sphinx, but you'll need it for the next section.)

1. Configure a static directory:

```
html_static_path = ['_static']
```

2. Put a file in it (e.g. docs/\_static/cat.png).
3. Use it.

## 2.3 Using CSS files

1. Add your CSS file as a static file as above (like docs/\_static/style.css).
2. Add the file name to the `html_theme_options['cssfiles']` list in `conf.py` (like `html_theme_options['cssfiles'] = ['_static/style.css']`)

You should read [better's CSS files](#) or poke around with your browser's element inspector to get an idea of what selectors you should override. `better_basic.css_t` is my fork of the basic theme's CSS, and `better.css_t` is the stylistic overrides.

## 2.4 Using Javascript files

1. Add your Javascript file as a static file as above.
2. Add the file name (relative to the static directory) to the `html_theme_options['scriptfiles']` list.

## 2.5 Suggested Sphinx options

Set `html_short_title` to "Home" so the first breadcrumb says "Home" instead of your long project title:

```
html_short_title = "Home"
```

[Change your sidebars](#). Since the nav bars have useful "next" and "previous" links in this theme, you can remove the "Next Topic" and "Previous Topic" sidebar components. Here's what that looks like:

```
html_sidebars = {
    '**': ['localtoc.html', 'sourcelink.html', 'searchbox.html'],
}
```

Here are two more custom sidebar components that the [mrjob docs](#) use to improve usability:

- On the index page, don't include any table of contents in the sidebar. Instead, link to commonly used pages.
- Link to the mailing list, or whatever the preferred support channel is for your project.

mrjob's config looks like this to support those components:

```
html_sidebars = {
    '**': ['localtoc.html', 'sidebarhelp.html', 'sourcelink.html',
          'searchbox.html'],
    'index': ['indexsidebar.html', 'sidebarhelp.html', 'sourcelink.html',
             'searchbox.html'],
}
```

Those HTML files live in `docs/_templates/`. See [Sphinx's templating guide](#) for more information about how to write them.



### 3.1 Sphinx style demo

See *sphinx-better-theme: usable documentation* for the table of contents. Reproducing it here would make Sphinx mad.

Feel free to open Github issues about the specifics of the styles on this page.

#### 3.1.1 Paragraph-level markup

**note, warning**

---

**Note:** This is a note. It can have `monospaced text`.

---

**Warning:** This is a note. `monospaced text`.

**versionadded, deprecated**

New in version 0.1: Some stuff added in a version.

Deprecated since version 0.1: Some stuff deprecated in a version.

**seealso**

**See also:**

**mrjob** Another awesome open source project

**Buildy** A cool online game

**rubric**

**A paragraph heading that is not used to create a TOC node**

**centered**

LICENSE AGREEMENT

**hlist**

- A list of
- short items
- that should be
- displayed
- horizontally

### 3.1.2 Misc

**glossary**

**glossary** A directive that contains a definition list with terms and definitions. The definitions will then be referencable with the `term` role.

**term** A string defined in the glossary.

Here we reference the *glossary* term.

**productionlist**

```
try_stmt    ::=  try1_stmt | try2_stmt
try1_stmt   ::=  "try" ":" suite
                ("except" [expression ["," target]] ":" suite)+
                ["else" ":" suite]
                ["finally" ":" suite]
try2_stmt   ::=  "try" ":" suite
                "finally" ":" suite
```

### 3.1.3 Showing code examples

**Double colon**

Here is some unhighlighted code:

```
old pond...
a frog leaps in
water's sound
```

## code-block

Line numbers with the second line emphasized:

```

1 if True:
2     print "This is some Python"

```

No line numbers:

```

if True:
    print "This is some Python"

```

## Tables

This	is	a	table
1	2	3	4
5	6	7	8

### 3.1.4 Inline markup

#### References

*A link to the above heading*

*A link to the index document*

Download this demo (demo.rst)

ENV\_VAR

token

**--option** <option>  
Description of option.

*--option*, *--option-without-ref*

*term* (see [Glossary](#) for an example with a link)

#### Other semantic markup

ABBR (an abbreviation) (hover)

**command** is an OS-level command.

*dfn* is the defining instance of a term in the text.

/a/file/path/*variable*/more

*GUI control label*

Control-x Control-f (keystroke sequence)

*Content-Type* (mail header)

**MAKE\_VAR**

*manpage* (1)

*Menu* → *Selection*

*mime/type*

*Usenet newsgroup (wat?)*

**Name of an executable program** (not just `:command:` for some reason?)

unquoted?regular\*[expression]

A piece of literal text with *variables*

**PEP 8**

**RFC 1072**

## Substitutions

Release 0.1.5, version 0.1.5, today Sep 08, 2018

## 3.1.5 Python

`py_module`

**py\_func** (*arg*, *a\_long\_argument=with\_a\_default*, *foo=bar*, *baz=qux*, *more=more*, *args=args*, *for=for*, *wrap-ping=wrapping*)

A Python function definition.

`py_func()`, `py_func_no_ref()`

**class PyClass** (*arg*)

**py\_method** (*arg*)

`PyClass`, `PyClassNoRef`, `PyClass.py_method()`, `py_method_no_ref()`

## 3.2 reStructuredText style demo

This is a simple demo of a subset of reStructuredText features.

### 3.2.1 Inline markup

The *quick* brown **fox** jumps over <sup>the</sup> lazy <sub>dog</sub>. *Title reference.*

Lorem ipsum *[Ref]* dolor sit amet.

Lorem ipsum<sup>1</sup> dolor sit amet ...<sup>2</sup>

---

<sup>1</sup> Text of the first footnote.

<sup>2</sup> Text of the second footnote.



### 3.2.2 Lists and quote-like blocks

- Bulleted list
- with two items
- 1. Numbered list
- 2. with
- 3. three items
- Nested
  1. List
    - Hooray
    - Hooray
    - Hooray
    - Hooray
  1. List
    - Hooray
    - Hooray
    - Hooray
    - Hooray
- Nested
  1. List

**term (up to a line of text)** Definition of the term, which must be indented  
and can even consist of multiple paragraphs

**next term** Description.

#### Paragraph heading

##### Local table of contents

- *reStructuredText style demo*
  - *Inline markup*
  - *Lists and quote-like blocks*
    - \* *Paragraph heading*
      - *Epigraph*
      - *Compound paragraph*
      - *Raw HTML*
  - *Sidebar*
  - *Admonition blocks*

(The above ToC triggers anchors around all page headings beyond what Sphinx does.)

### Topic

A topic is like a block quote with a title, or a self-contained section with no subsections. Use the “topic” directive to indicate a self-contained idea that is separate from the flow of the document. Topics may occur anywhere a section or transition may occur. Body elements and topics may not contain nested topics.

`parsed-literal` is a literal-looking block with **parsed** *text*

### Epigraph

No matter where you go, there you are.

—Buckaroo Banzai

### Compound paragraph

This is a compound paragraph. The ‘rm’ command is very dangerous. If you are logged in as root and enter

```
cd /  
rm -rf *
```

you will erase the entire contents of your file system.

### Raw HTML

#### 3.2.3 Sidebar

### Sidebar

Sidebar are like miniature, parallel documents that occur inside other documents, providing related or reference material. A sidebar is typically offset by a border and “floats” to the side of the page; the document’s main text may flow around it. Sidebars can also be likened to super-footnotes; their content is outside of the flow of the document’s main text.

#### 3.2.4 Admonition blocks

**Attention:** attention block block block block block block block block block block

**Caution:** caution block

**Danger:** danger block

**Error:** error block

**Hint:** hint block

**Important:** important block

**Note:** note block

**Tip:** tip block

**Warning:** warning block

**custom admonition**

with content



- It looks nice.
- It's easy to style with CSS (no `_theme/` directory required).
- The navigation is laid out better.
- It works well on small screens and mobile devices.

If you find sphinx-better-theme lacking in any of these areas, *please* [open a Github issue](#).

## 4.1 It looks nice

By default, the only colors are the background color, the body text color, and the link color. Content is separated by layout and whitespace, not background color changes.

The font defaults are a little more modern. There is less variation in font styles. Content is wrapped to about 100 characters by default.

Some docs may look better with more liberal use of color. This theme supports that visual style via CSS rules.

## 4.2 It's easy to style with CSS

Unlike every other Sphinx theme I'm aware of, sphinx-better-theme lets you customize it with CSS without needing a `_theme/` directory, or anything beyond a CSS file. And you don't even need that; you can declare inline CSS in your Sphinx config file.

```
html_theme_options = {
    'inlinecss': 'color: green;',
    'cssfiles': ['_static/my_style.css'],
}
```

One of this project's major goals is to make visual customization easier so that projects can brand their docs better.

## 4.3 The navigation is laid out better

The project title is displayed in a header instead of a tiny breadcrumb link.

The links to the next and previous pages in the navigation bars at the top and bottom use the actual names of the pages instead of the opaque “next” and “previous”. This feature was lifted from the [Guzzle project’s docs](#).

## 4.4 It works well on small screens and mobile devices

The built-in themes do not work well on small screens. A few other third party themes get this right, but it’s not widespread.

## 4.5 Deficiencies

The markup isn’t easy enough to fully customize with CSS. One of the long-term goals of this project is to make the markup more semantic.

The placement of the logo image isn’t good.

[Your idea here](#)

## CHAPTER 5

---

### Compatibility

---

sphinx-better-theme is compatible with Sphinx 0.6.4+ and Jinja 2.3.1+. Older versions may work but have not been tested.





---

### Projects using sphinx-better-theme

---

- [mrjob](#) has tens of thousands of words of guides and API docs.
- [Pillow](#) is the modern fork of the Python Imaging Library and recently got a new set of docs.
- [zhang-shasha](#) is a single page of simple Python library documentation.
- [pivotal\\_tools](#) is a single page of command line tool documentation.
- [Rsyslog](#) is a **rocket-fast system** for **log** processing. It offers high-performance, great security features and a modular design.



## CHAPTER 7

---

### History/Roadmap

---

**v0.1:** Basic CSS-customizable style that looks nice in its default state

**v0.11:** Add easy Google Analytics support

**v0.12:** Improve base styles, responsive layout, document usage with Read the Docs

**v0.13:** Further style improvements, relbar shows only full titles of next/previous links

**v0.1.4:** Semantic versioning, mobile bug fixes, sidebar slimming, style improvements

**v0.1.5:** Remove a few useless options, allow remote CSS files, fix Google Analytics integration

**v0.2:** (planned) Rewrite markup to be semantic and customizable

**v1.0:** (planned) Extreme documentation polish and rounding out of edge cases



---

### Other themes to check out

---

[kr-sphinx-themes](#) has two nice, clean themes used for many popular (and good-looking) projects.

[sphinx-readable-theme](#) is a theme optimized for readability of *apidoc*-generated docs.

[sphinxtheme-readability](#) is another attempt at a clean theme.

[Cloud](#) is a feature-rich theme with a responsive layout.

A few different themes are available for download at [sphinx-themes](#).

[Read the Docs](#) uses a custom theme as the default for all docs hosted there.

The [Guzzle](#) project uses a [heavily customized theme](#) that's also used by [aws-cli](#).

The [Julia language](#) also uses an interesting custom theme.



---

## Bibliography

---

[Ref] Book or article reference, URL or whatever.





## Symbols

`-option <option>`  
command line option, [11](#)

## C

command line option  
`-option <option>`, [11](#)

## E

`ENV_VAR`, [11](#)  
environment variable  
`ENV_VAR`, [11](#)

## G

glossary, [10](#)

## P

`py_func()` (built-in function), [12](#)  
`py_method()` (PyClass method), [12](#)  
PyClass (built-in class), [12](#)  
Python Enhancement Proposals  
PEP 8, [12](#)

## R

RFC  
RFC 1072, [12](#)

## T

term, [10](#)