sphinx-better-theme

Release 0.13

Sep 01, 2018

Contents

sphinx-better-theme is a theme for Sphinx that tries to be better than the built-in themes. See *Anticipatory FAQ* for details.

You can get the source and open issues on Github.

CHAPTER 1

User guide

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. In particular, consider setting `html_short_title` to `"Home"` so the first breadcrumb says "Home" instead of your long project title.

1.1 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)
  'cssfiles': [],

```

(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 search in the sidebar. some of us think Sphinx's search is
    # garbage and just want it to go away.
    'enablesidebarsearch': 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': '',
}
```

1.2 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.

1.3 Using CSS files

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

You should read better's CSS files 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.

1.4 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.

CHAPTER 2

Demos

2.1 Sphinx style demo

See *sphinx-better-theme* 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.

2.1.1 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

2.1.2 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
```

2.1.3 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:

```
if True:
    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

2.1.4 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 0.13, version 0.13, today Sep 01, 2018

2.1.5 Python

`py_module`

py_func (*arg*, *a_long_argument=with_a_default*, *foo=bar*, *baz=qux*, *more=more*, *args=args*, *for=for*, *wrapping=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()`

2.2 reStructuredText style demo

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

2.2.1 Inline markup

The *quick* brown **fox** jumps `over` $^{\text{the}}$ lazy $_{\text{dog}}$. *Title reference.*

Lorem ipsum *[Ref]* dolor sit amet.

Lorem ipsum[1] dolor sit amet . . . [2]

[1] Text of the first footnote.
[2] Text of the second footnote.

2.2.2 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

(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

2.2.3 Sidebar

Sidebar

Sidebars 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.

2.2.4 Admonition blocks

Attention: attention block 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

CHAPTER 3

Anticipatory FAQ

This material will probably move to a blog post at some point.

3.1 Better how?

The default Sphinx theme isn't bad, and the Python 3 theme is even better. But both themes are problematic for many projects. Specifically, they are difficult to customize beyond a few inconsequential color and layout settings. Their markup and stylesheets do not lend themselves to tweaking. This project aims to mitigate those problems.

Additionally, `better-sphinx-theme` contains a few differences that offer subjective improvements over the default theme and other built-in themes:

1. Less unnecessary color and fewer different colors. I'm not opposed to the use of color on the web. I just think it's more professional to use a consistent color palette. This site doesn't use much color, but `better-sphinx-theme` ought to support color if you want it.
2. Fewer fonts and smaller variations in font styles. Headings don't need to be huge to be readable, and the sidebar doesn't always need its own set of styles.
3. Forced use of the full page width. 45-90 characters is generally agreed to be the most readable line length. Some projects make use of the full width to display tables, but many projects do just fine without them.
4. Better customization. The default theme teases you by putting some color settings in the theme configuration, but a few key components like code blocks aren't customizable at all. `sphinx-better-theme` makes it easy to change settings with little overhead by letting you add your own CSS files without all the trouble of making a new theme.

A future version of this theme will have semantic markup. The default theme is a sea of `<div>` tags. Even if the much-heralded machine-readable web never pans out, it strikes me as a nice symmetry to have a program's documentation be easily consumable by other programs. It can also make CSS rules clearer.

3.2 Why encourage customization?

Branding is important for even small projects. Take a quick look at the built-in Sphinx themes. They're all pretty distinctive, and several are for specific projects or are based on themes created for specific projects.

As a maintainer, I don't want my project's documentation to use exactly the same theme as someone else's. The layout and conventions should be consistent with other projects in the same environment (in my case, other Python projects), but it should still be possible to glance at a page and know that you're looking at Project X, rather than "a project styled using the default Sphinx template."

A logo helps, but most projects' tiny teams don't have the skills necessary to create an effective logo. Besides, fonts and colors can have an even stronger effect on the branding of a site. Just look at Django's documentation (which, by the way, uses a custom Sphinx theme). You can tell at a glance what project's web site you're on. The same goes for the difference betwen Python 2's documentation and Python 3's documentation.

3.3 How does this theme encourage customization better than the default?

It's much easier to add your own CSS. The theme's CSS rules and markup are a little easier to understand, and that situation should improve over time. Additionally, there are more meaningful theme options for disabling unnecessary widgets.

3.4 Why not just contribute to Sphinx itself?

I'll probably make an attempt eventually. For now I'd just like to validate my ideas.

CHAPTER 4

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.

CHAPTER 5

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'
```

5.1 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 must contain at least the line `sphinx-better-theme==0.13`, as well as any other dependencies your docs might have that are separate from your project's dependencies. 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 in *Installation*.

CHAPTER 6

Projects using sphinx-better-theme

- mrjob (both narrative and API docs)
- pivotal_tools (single-page command line tool documentation)

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

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

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

CHAPTER 8

Other themes to check out

Cloud is a nice, feature-rich theme from which I plan to steal more than one feature.

A few different themes are available for download at sphinx-themes.

Read the Docs uses a nice 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.

Bibliography

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

Index