
My Project

Release 0.6.0

Ryan Roemer

Oct 08, 2017

Contents

| | | |
|----------|----------------------------------|-----------|
| 1 | Contents | 3 |
| 1.1 | Sphinx Bootstrap Theme | 3 |
| 1.2 | Examples | 8 |
| 1.3 | Sidebar Examples | 11 |
| 1.4 | Subdirectory Page | 12 |
| 1.5 | History | 13 |
| 1.6 | Future Work | 19 |
| 2 | Indices and tables | 23 |

This is a basic demo site for the [Sphinx Bootstrap Theme](#) that provides a minimal working Sphinx site using the theme. Please see the [Sphinx Bootstrap Theme](#) page (also linked below in the TOC) for full details and installation instructions.

Setting up and using the theme.

1.1 Sphinx Bootstrap Theme

This [Sphinx theme](#) integrates the [Bootstrap](#) CSS / JavaScript framework with various layout options, hierarchical menu navigation, and mobile-friendly responsive design. It is configurable, extensible, and can use any number of different [Bootswatch](#) CSS themes.

1.1.1 Introduction and Demos

The theme is introduced and discussed in the following posts:

- [12/09/2011 - Twitter Bootstrap Theme for Sphinx](#)
- [11/19/2012 - Sphinx Bootstrap Theme Updates - Mobile, Dropdowns, and More](#)
- [2/12/2013 - Sphinx Bootstrap Theme 0.1.6 - Bootstrap and Other Updates](#)
- [4/10/2013 - Sphinx Bootstrap Theme 0.2.0 - Now with Bootswatch!](#)
- [9/8/2013 - Sphinx Bootstrap Theme 0.3.0 - Bootstrap v3 and more!](#)

Examples of the theme in use for some public projects:

- [Sphinx Bootstrap Theme](#): This project, with the theme option `'bootswatch_theme': "sandstone"` to use the “Sandstone” [Bootswatch](#) theme.
- [Django Cloud Browser](#): A Django reusable app for browsing cloud datastores (e.g., Amazon Web Services S3).
- [C++ Format](#): Small, safe and fast formatting library for C++.

The theme demo website also includes an [examples page](#) for some useful illustrations of getting Sphinx to play nicely with Bootstrap (also take a look at the [examples source](#) for the underlying `reStructuredText`).

1.1.2 Installation

Installation from PyPI is fairly straightforward:

1. Install the package:

```
$ pip install sphinx_bootstrap_theme
```

2. Edit the “conf.py” configuration file to point to the bootstrap theme:

```
# At the top.
import sphinx_bootstrap_theme

# ...

# Activate the theme.
html_theme = 'bootstrap'
html_theme_path = sphinx_bootstrap_theme.get_html_theme_path()
```

1.1.3 Customization

The can be customized in varying ways (some a little more work than others).

Theme Options

The theme provides many built-in options that can be configured by editing your “conf.py” file:

```
# (Optional) Logo. Should be small enough to fit the navbar (ideally 24x24).
# Path should be relative to the ``_static`` files directory.
html_logo = "my_logo.png"

# Theme options are theme-specific and customize the look and feel of a
# theme further.
html_theme_options = {
    # Navigation bar title. (Default: ``project`` value)
    'navbar_title': "Demo",

    # Tab name for entire site. (Default: "Site")
    'navbar_site_name': "Site",

    # A list of tuples containing pages or urls to link to.
    # Valid tuples should be in the following forms:
    #   (name, page)                # a link to a page
    #   (name, "/aa/bb", 1)         # a link to an arbitrary relative url
    #   (name, "http://example.com", True) # arbitrary absolute url
    # Note the "1" or "True" value above as the third argument to indicate
    # an arbitrary url.
    'navbar_links': [
        ("Examples", "examples"),
        ("Link", "http://example.com", True),
    ],

    # Render the next and previous page links in navbar. (Default: true)
    'navbar_sidebarel': True,

    # Render the current pages TOC in the navbar. (Default: true)
```



```

'navbar_pagenav': True,

# Tab name for the current pages TOC. (Default: "Page")
'navbar_pagenav_name': "Page",

# Global TOC depth for "site" navbar tab. (Default: 1)
# Switching to -1 shows all levels.
'globaltoc_depth': 2,

# Include hidden TOCs in Site navbar?
#
# Note: If this is "false", you cannot have mixed ``:hidden:`` and
# non-hidden ``toctree`` directives in the same page, or else the build
# will break.
#
# Values: "true" (default) or "false"
'globaltoc_includehidden': "true",

# HTML navbar class (Default: "navbar") to attach to <div> element.
# For black navbar, do "navbar navbar-inverse"
'navbar_class': "navbar navbar-inverse",

# Fix navigation bar to top of page?
# Values: "true" (default) or "false"
'navbar_fixed_top': "true",

# Location of link to source.
# Options are "nav" (default), "footer" or anything else to exclude.
'source_link_position': "nav",

# Bootswatch (http://bootswatch.com/) theme.
#
# Options are nothing (default) or the name of a valid theme
# such as "cosmo" or "sandstone".
'bootswatch_theme': "united",

# Choose Bootstrap version.
# Values: "3" (default) or "2" (in quotes)
'bootstrap_version': "3",
}

```

Note for the navigation bar title that if you don't specify a theme option of `navbar_title` that the “conf.py” project string will be used. We don't use the `html_title` or `html_short_title` values because by default those both contain version strings, which the navigation bar treats differently.

Bootstrap Versions

The theme supports Bootstrap v2.3.2 and v3.3.7 via the `bootstrap_version` theme option (of "2" or "3"). Some notes regarding version differences:

- Bootstrap 3 has dropped support for [sub-menus](#), so while supported by this theme, they will not show up in site or page menus.
- Internally, “navbar.html” is the Sphinx template used for Bootstrap v3 and “navbar-2.html” is the template used for v2.
- If you are unsure what to choose, choose Bootstrap 3. If you experience some form of compatibility issues, then try and use Bootstrap 2.

Extending “layout.html”

As a more “hands on” approach to customization, you can override any template in this Sphinx theme or any others. A good candidate for changes is “layout.html”, which provides most of the look and feel. First, take a look at the “layout.html” file that the theme provides, and figure out what you need to override. As a side note, we have some theme-specific enhancements, such as the `navbarextra` template block for additional content in the navbar.

Then, create your own “_templates” directory and “layout.html” file (assuming you build from a “source” directory):

```
$ mkdir source/_templates
$ touch source/_templates/layout.html
```

Then, configure your “conf.py”:

```
templates_path = ['_templates']
```

Finally, edit your override file “source/_templates/layout.html”:

```
{# Import the theme's layout. #}
{% extends "!layout.html" %}

{# Add some extra stuff before and use existing with 'super()' call. #}
{% block footer %}
    <h2>My footer of awesomeness.</h2>
    {{ super() }}
{% endblock %}
```

Adding Custom CSS

Alternately, you could add your own custom static media directory with a CSS file to override a style, which in the demo would be something like:

```
$ mkdir source/_static
$ touch source/_static/my-styles.css
```

In the new file “source/_static/my-styles.css”, add any appropriate styling, e.g. a bold background color:

```
footer {
    background-color: red;
}
```

Then, in “conf.py”, edit this line:

```
html_static_path = ["_static"]
```

From there it depends on which version of Sphinx you are using:

Sphinx <= 1.5

You will need the override template “source/_templates/layout.html” file configured as above, but with the following code:

```
{# Import the theme's layout. #}
{% extends "!layout.html" %}

{# Custom CSS overrides #}
{% set css_files = css_files + ['_static/my-styles.css'] %}
```

Note: See [Issue #159](#) for more information.

Sphinx >= 1.6.1

Add a setup function in “conf.py” with stylesheet paths added relative to the static path:

```
def setup(app):
    app.add_stylesheet("my-styles.css") # also can be a full URL
    # app.add_stylesheet("ANOTHER.css")
    # app.add_stylesheet("AND_ANOTHER.css")
```

Tip: Sphinx automatically calls your `setup` function defined in “conf.py” during the build process for you. There is no need to, nor should you, call this function directly in your code.

1.1.4 Theme Notes

Sphinx

The theme places the global TOC, local TOC, navigation (prev, next) and source links all in the top Bootstrap navigation bar, along with the Sphinx search bar on the left side.

The global (site-wide) table of contents is the “Site” navigation dropdown, which is a configurable level rendering of the `toctree` for the entire site. The local (page-level) table of contents is the “Page” navigation dropdown, which is a multi-level rendering of the current page’s `toc`.

Bootstrap

The theme offers Bootstrap v2.x and v3.x, both of which rely on jQuery v.1.9.x. As the jQuery that Bootstrap wants can radically depart from the jQuery Sphinx internal libraries use, the library from this theme is integrated via `noConflict()` as `$jqTheme`.

You can override any static JS/CSS files by dropping different versions in your Sphinx “_static” directory.

1.1.5 Contributing

Contributions to this project are most welcome. Please make sure that the demo site builds cleanly, and looks like what you want. First build the demo:

```
$ fab clean && fab demo
```

Then, view the site in the development server:

```
$ fab demo_server
```

Also, if you are adding a new type of styling or Sphinx or Bootstrap construct, please add a usage example to the “Examples” page.

Note: If you are in Python 3, Fabric isn’t available, so we have a very rough Makefile in its place. Try:

```
$ make clean && make demo
```

Then, view the site in the development server:

```
$ make demo_server
```

1.1.6 Licenses

Sphinx Bootstrap Theme is licensed under the [MIT](#) license.

[Bootstrap v2](#) is licensed under the Apache license 2.0.

[Bootstrap v3.1.0+](#) is licensed under the MIT license.

1.2 Examples

Various examples of Bootstrap styling applied to Sphinx constructs. You can view the [source](#) of this page to see the specific reStructuredText used to create these examples.

1.2.1 Headings

This is a first level heading (h1).

Sub-Heading

This is a second level heading (h2).

Sub-Sub-Heading

This is a third level heading (h3).

1.2.2 Code

The Sphinx Bootstrap Theme uses Bootstrap styling for inline code text and

```
multiline  
code text
```

Here's an included example with line numbers.

```
1  """Sphinx bootstrap theme."""  
2  import os  
3  
4  VERSION = (0, 6, 0)  
5  
6  __version__ = ".".join(str(v) for v in VERSION)  
7  __version_full__ = __version__  
8  
9  def get_html_theme_path():  
10     """Return list of HTML theme paths."""  
11     cur_dir = os.path.abspath(os.path.dirname(__file__))  
12     return [cur_dir]
```

It also works with existing Sphinx highlighting:

```
<html>
  <body>Hello World</body>
</html>
```

```
def hello():
    """Greet."""
    return "Hello World"
```

```
/**
 * Greet.
 */
function hello(): {
    return "Hello World";
}
```

1.2.3 Admonitions

The Sphinx Bootstrap Theme uses the Bootstrap `alert` classes for Sphinx admonitions.

Note

Note: This is a **note**.

Todo

Todo: This is a **todo**.

Warning

Warning: This is a **warning**.

Danger

Danger: This is **danger**-ous.

1.2.4 Footnotes

I have footnoted a first item¹ and second item². This also references the second item².

¹ My first footnote.

² My second footnote.

1.2.5 Icons

Icons are different in Bootstrap 2 and 3, so you will only see an icon below for the version of Bootstrap that we used to build these docs.

Bootstrap 2

The following template HTML:

```
<span class="icon-star-empty"></span>
```

translates to a neat star:

Bootstrap 3

The following template HTML:

```
<span class="glyphicon glyphicon-star-empty"></span>
```

translates to a neat star:

1.2.6 Tables

Here are some examples of Sphinx [tables](#). The Sphinx Bootstrap Theme removes all Sphinx `docutils` classes and replaces them with the default Bootstrap `table` class. You can add additional table classes using the Sphinx `cssclass::` directive, as demonstrated in the following tables.

Grid

A “**bordered**” grid table:

| Header1 | Header2 | Header3 | Header4 |
|-------------|---------|---------|---------|
| row1, cell1 | cell2 | cell3 | cell4 |
| row2 ... | ... | ... | |
| ... | ... | ... | |

which uses the directive:

```
.. cssclass:: table-bordered
```

Simple

A simple “**striped**” table:

| H1 | H2 | H3 |
|-------|-------|-------|
| cell1 | cell2 | cell3 |
| ... | ... | ... |
| ... | ... | ... |

which uses the directive:

```
.. cssclass:: table-striped
```

And a “**hoverable**” table:

| H1 | H2 | H3 |
|-------|-------|-------|
| cell1 | cell2 | cell3 |
| ... | ... | ... |
| ... | ... | ... |

which uses the directive:

```
.. cssclass:: table-hover
```

1.2.7 Code Documentation

An example Python function.

format_exception (*etype*, *value*, *tb*[, *limit=None*])

Format the exception with a traceback.

Parameters

- **etype** – exception type
- **value** – exception value
- **tb** – traceback object
- **limit** (*integer or None*) – maximum number of stack frames to show

Return type list of strings

An example JavaScript function.

class MyAnimal (*name*[, *age*])

Arguments

- **name** (*string*) – The name of the animal
- **age** (*number*) – an optional age for the animal

1.3 Sidebar Examples

This page demonstrates the use of a simple navigation sidebar.

1.3.1 Headings

This is a first level heading (h1).

Sub-Heading

This is a second level heading (h2).

Sub-Sub-Heading

This is a third level heading (h3).

1.3.2 Let's Rock Some Lorem!

... and now some Lorem text so that we can check scrolling

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

1.4 Subdirectory Page

This page is in a subdirectory.

1.4.1 Headings

This is a first level heading (h1).

Sub-Heading

This is a second level heading (h2).

Sub-Sub-Heading

This is a third level heading (h3).

Development history and feature wish lists.

1.5 History

1.5.1 v0.6.0

- Upgrade to Bootstrap / Bootswatch v3.3.7. (@sveneys) #164.

1.5.2 v0.5.3

- Fix sidebar jQuery issue with height. (@Sheile) #157.

1.5.3 v0.5.2

bad release

1.5.4 v0.5.1

- Fix `@import url()` CSS imports to be relative paths so that you can have a non-root / nested site.

1.5.5 v0.5.0

- Fix `css_files` breakage from Sphinx 1.6+ update. #158, #160.
- **Breaking Change:** Remove `bootswatch_css_custom` override, and instead opt for documenting idiomatic Sphinx-version specific generic overrides for custom CSS.

1.5.6 v0.4.14

- Fix visibility of multiple footnote references. (@drewhutchison) #152.

1.5.7 v0.4.13

- Fix search with larger sidebar. (@cemsbr) #148.

1.5.8 v0.4.12

- Fix typo in theme CSS rule. (@vkoby) #144.

1.5.9 v0.4.11

- Fix logo sizing issue on mobile with RTD. #142.

1.5.10 v0.4.10

- Fix logo / brand title wrapping bug. (@miketheman) #141.

1.5.11 v0.4.9

- Update to Bootstrap v3.3.6 and Bootswatch v3.3.6+1. (@ppyv)

1.5.12 v0.4.8

- Fix sidenav overflow / scrolling. #136.

1.5.13 v0.4.7

- Fix jumpy sidenav #131. (@cslarsen)

1.5.14 v0.4.6

- Switch demo to “sandstone” theme.
- Update to Bootswatch v3.3.4+1.
- Update to Bootstrap v3.3.4.
- Update Sphinx website links. (@mrmsl)

1.5.15 v0.4.5

- Add Glyphicon fonts to Bootswatch 2 static assets. Also add example using the icons so we can catch errors like this better in the future.
- Fix CSS generation with 'navbar_fixed_top': "false" and 'bootstrap_version': "2". Fixes #121. (@EricFromCanada)

1.5.16 v0.4.4

- Reset `.container` padding after Sphinx 1.3b commit overrode CSS. Fixes #114. (@EricFromCanada)
- Fix situation where Python 3 + modern Sphinx results in empty strings being interpreted as a non-existent Bootswatch theme for `theme_bootswatch_theme` by permissively allowing old (empty quotes) or new (empty or None) styles. Fixes #115. (@EricFromCanada, @peteut, @mdboom)

1.5.17 v0.4.3

- Add `fonts/` directories to bootswatch. (@gkthiruvathukal)
- Remove divider-vertical from BS3 navbar.html. (@kaycebasques)

1.5.18 v0.4.2

- Update to Bootstrap v3.2.0.

1.5.19 v0.4.1

- Fix non-fixed-top navbar. (@Danack)
- Add config option for page name tab. (@masklinn)

1.5.20 v0.4.0

- Fix bug preventing Glyphicons from working with Bootswatch themes.

1.5.21 v0.3.9

- Restyle alerts and admonitions. (@masklinn)

1.5.22 v0.3.8

- Update to Bootstrap v3.1.0. (@torbjoernk)

1.5.23 v0.3.7

- Add footnote styling. (@russell)
- Update search.html to allow search locally. (@fjfeijoo)
- Updated search template for BS3. (@russell)

1.5.24 v0.3.6

- Add bootswatch glyphicon links in bootswatch.
- Selectively hide links on navbar in certain view sizes. From here on out, *hidden-sm* in BS3 is applied to next/previous and source navigation links.
- Make navbar logo work with subdirectories. (@cdbennett)

1.5.25 v0.3.5

- Adds navbar logo's back to BS3. Switch to actual `` tags for logos. Fixes #52.
- Fix responsive Bootswatch for Bootstrap v2.

1.5.26 v0.3.4

- Switch demo to “flatly” theme.
- Fix the long-broken “Source” nav. link.
- Add missing “flatly” to bootswatch CSS static files.

1.5.27 v0.3.3

- Allow custom CSS overrides, even on Bootstrap CSS. Completes #68.
- Add offline bootswatch files. Completes #19.
- Preserve HTML inside literal code blocks. (@tristanlins)

1.5.28 v0.3.2

- Fix scroll handlers for nav. bar. (@sccolbert)
- Fix background color visibility behind rounding in code blocks. (@kosiakk)

1.5.29 v0.3.1

- Re-add navbar_links theme option lost in bad merge. (@newgene)
- Fixed display of sidebar. (@adamcharnock, @russell)

1.5.30 v0.3.0

- Add Bootstrap v3.0.0 with legacy option for v2.3.2. (@MiCHiLU)

1.5.31 v0.2.9

- Add navbar_links theme option. (@newgene)
- Add navbarextra block in “layout.html”. (@grncdr)

1.5.32 v0.2.8

- Sphinx compatible Sidebars. (@russell)
- Topnav sidebarrel can now be disabled. (@russell)
- Topnav page nav menu can now be disabled. (@russell)

1.5.33 v0.2.7

- Add custom nav bar links. (@russell) Completes #34.
- Fix wrapping of line numbers in code includes. (@russell) Fixes #35.
- Truncate long page titles in navigation bar. (@aababilov) Fixes #27.

1.5.34 v0.2.6

- Use network path for Bootswatch (@nail)
- Switch from distribute to setuptools. (Suggested by @thedrow)

1.5.35 v0.2.5

- Search page styling. (@russell)

1.5.36 v0.2.4

- Adjust the max width of field lists. (@russell)
- Update to Bootstrap v2.3.2.
- Navbar search box now uses bootstrap search-query class. (@russell)
- Field-list tables now have an inherited width. (@russell)

1.5.37 v0.2.3

- Put navbar within a *container*. (@inducer)
- Add `navbar_site_name` for renaming site nav. tab. (Suggested by @inducer)

1.5.38 v0.2.2

- Better literal markup handling for Bootstrap code formatting. (@russell)
- Scroll window when jumping to an anchor. (@russell)

1.5.39 v0.2.1

- Fix code styling collision for cross references and inline code blocks. (@russell)

1.5.40 v0.2.0

- Update to Bootstrap v2.3.1.
- Add `bootswatch_theme` option for [Bootswatch](#) CSS theme support. (@zyga)

1.5.41 v0.1.8

- Add `globaltoc_includehidden` option.

1.5.42 v0.1.7

- Add Python 3 support. (@zyga)
- Add support for `navbar_title` theme configuration. The documentation originally stated that `html_short_title` was supported for overriding the navbar title (brand), but this never actually worked. (Thanks to Tim Kedmenec for pointing this out).

1.5.43 v0.1.4

- Remove the `Site` nav button if no other pages.
- Added jQuery v1.9.1 with `noConflict()` to allow underlying Sphinx to use whatever jQuery it wants.
- Update to Bootstrap v2.3.0.
- Fix multi-word Bootstrap-styled `code` elements.

1.5.44 v0.1.3

- Convert inline code to Bootstrap-styled `code` elements.

1.5.45 v0.1.2

- Add `globaltoc_depth` theme option.
- Add Bootstrap alert styling to “note”, “warning” Sphinx directives.

1.5.46 v0.1.1

- Add Bootstrap table styling.

1.5.47 v0.1.0

- Add support for deployment via PyPI.

1.5.48 v0.0.6

- Fix logo display in navbar.

1.5.49 v0.0.4

- Get mobile (iPhone) viewport and nav menus working.
- Add new theme options `navbar_class`, `source_link_position`.

1.5.50 v0.0.3

- Update to Bootstrap v2.2.1.
- Switch to responsive CSS.
- Make navbar menus do real dropdowns recursively.

1.5.51 v0.0.2

- Update to Bootstrap v2.0. ([@oscarcp](#))

1.5.52 v0.0.1

- Original theme based on Bootstrap v1.4.0.

1.6 Future Work

1.6.1 CSS / Layout

- Search Bar: Is aligned right inconsistently in BS2 vs. BS3.
- Footnotes: Need much better styling. Maybe switch to tooltips.
- Optionally move the `localtoc` from the top navigation bar into it's own navbar below the page title.

1.6.2 Infrastructure

- Re-enable the downloads.

1.6.3 Upgrade Notes

When upgrading, you must upgrade **both** Bootstrap and Bootswatch — Bootswatch releases are custom tailored to the a specific version of Bootstrap. The version of jQuery (1.11) should be OK to leave as is unless a specific issue arises.

In the below examples we are upgrading to bootstrap 3.3.7; Bootstrap 2 versions should not have a need to be changed.

Updating Bootstrap

1. Get new `bootstrap-VERSION` and drop in `bootstrap/static`.

- New versions should be available [here](#).
- Choose “Bootstrap”:

Compiled and minified CSS, JavaScript, and fonts. No docs or original source files are included.

2. Manually `grep` and replace instances of `(jQuery)` with `(window.$jqTheme || window.jQuery)` in files in the new `bootstrap-VERSION/js` directory. See below.

3. Update `VERSION` in `sphinx_bootstrap_theme/bootstrap/layout.html`:

```
{% if theme_bootstrap_version == "3" %}
  {% set bootstrap_version, navbar_version = "3.3.7", "" %}
  {% set bs_span_prefix = "col-md-" %}
{% else %}
  {% set bootstrap_version, navbar_version = "2.3.2", "-2" %}
  {% set bs_span_prefix = "span" %}
{% endif %}
```

4. Update `VERSION` in `sphinx_bootstrap_theme/bootstrap/static/bootstrap-sphinx.css_t`

```
{% if theme_bootstrap_version == "3" %}
  {% set bootstrap_version, bootstrap_additional_css = "3.3.7", "theme" %}
{% else %}
  {% set bootstrap_version, bootstrap_additional_css = "2.3.2", "responsive" %}
{% endif %}
```

Choose your favorite text replacement tool for step 2, we will use `grep` and `sed`. The example here will be with the file `bootstrap.js` for understandability (since it is not minified).

```
# First, locate the all of the calls to jQuery
$ grep -Hn --color=auto '(jQuery)' bootstrap.js
bootstrap.js:17:}(jQuery);
bootstrap.js:77:}(jQuery);
bootstrap.js:172:}(jQuery);
bootstrap.js:298:}(jQuery);
bootstrap.js:536:}(jQuery);
bootstrap.js:749:}(jQuery);
bootstrap.js:915:}(jQuery);
bootstrap.js:1255:}(jQuery);
bootstrap.js:1776:}(jQuery);
bootstrap.js:1885:}(jQuery);
bootstrap.js:2058:}(jQuery);
bootstrap.js:2214:}(jQuery);
bootstrap.js:2377:}(jQuery);

# Make the replacement in-place while creating
# a bootstrap.js.bak file (backup of the original)
$ sed -i.bak 's/(jQuery)/(window.$jqTheme || window.jQuery)/g' bootstrap.js

# Verify there are no more (jQuery) left, and search our replacement
# for sanity checking (all the line numbers should match up)
$ grep -Hn --color=auto '(jQuery)' bootstrap.js
$ grep -Hn --color=auto '(window.$jqTheme || window.jQuery)' bootstrap.js
bootstrap.js:17:}(window.$jqTheme || window.jQuery);
bootstrap.js:77:}(window.$jqTheme || window.jQuery);
bootstrap.js:172:}(window.$jqTheme || window.jQuery);
bootstrap.js:298:}(window.$jqTheme || window.jQuery);
bootstrap.js:536:}(window.$jqTheme || window.jQuery);
bootstrap.js:749:}(window.$jqTheme || window.jQuery);
bootstrap.js:915:}(window.$jqTheme || window.jQuery);
bootstrap.js:1255:}(window.$jqTheme || window.jQuery);
bootstrap.js:1776:}(window.$jqTheme || window.jQuery);
bootstrap.js:1885:}(window.$jqTheme || window.jQuery);
bootstrap.js:2058:}(window.$jqTheme || window.jQuery);
bootstrap.js:2214:}(window.$jqTheme || window.jQuery);
bootstrap.js:2377:}(window.$jqTheme || window.jQuery);

# IMPORTANT! Check your work! Most of these were all in either
# comments or error strings (that we want to leave as is),
# but line 7 below needs to be updated!
#
# So line 7 should be changed (by you) to be
#
#     if (typeof (window.$jqTheme || window.jQuery) === 'undefined') {
$ grep -Hn --color=auto 'jQuery' bootstrap.js
bootstrap.js:7:if (typeof jQuery === 'undefined') {
bootstrap.js:8:  throw new Error('Bootstrap's JavaScript requires jQuery')
bootstrap.js:15:  throw new Error('Bootstrap's JavaScript requires jQuery version_
↪1.9.1 or higher, but lower than version 4')
```



```
bootstrap.js:17:}(window.$jqTheme || window.jQuery);
bootstrap.js:77:}(window.$jqTheme || window.jQuery);
bootstrap.js:172:}(window.$jqTheme || window.jQuery);
bootstrap.js:298:}(window.$jqTheme || window.jQuery);
bootstrap.js:536:}(window.$jqTheme || window.jQuery);
bootstrap.js:749:}(window.$jqTheme || window.jQuery);
bootstrap.js:915:}(window.$jqTheme || window.jQuery);
bootstrap.js:1255:}(window.$jqTheme || window.jQuery);
bootstrap.js:1260: * Inspired by the original jQuery.tipsy by Jason Frame
bootstrap.js:1624: // Avoid using $.offset() on SVGs since it gives incorrect_
↳results in jQuery 3.
bootstrap.js:1776:}(window.$jqTheme || window.jQuery);
bootstrap.js:1885:}(window.$jqTheme || window.jQuery);
bootstrap.js:2058:}(window.$jqTheme || window.jQuery);
bootstrap.js:2076: // jscs:disable requireDollarBeforejQueryAssignment
bootstrap.js:2078: // jscs:enable requireDollarBeforejQueryAssignment
bootstrap.js:2214:}(window.$jqTheme || window.jQuery);
bootstrap.js:2377:}(window.$jqTheme || window.jQuery);

# If all went according to plan, delete the backup
$ rm bootstrap.js.bak
```

Update Bootswatch

In this example we will walk through how to create the necessary structure using version 3.3.7 of Bootswatch. When updating in the future, replace 3.3.7 in the below with the version of Bootswatch you are upgrading to. **Make sure** that you are upgrading to the **same** version as Bootstrap!

```
# Go to a familiar working location, we choose ~/Desktop for this example
$ cd ~/Desktop

# Download the source code for bootswatch
$ git clone https://github.com/thomaspark/bootswatch.git
$ cd bootswatch

# Checkout the tagged release (use `git tag -l` to see all options)
$ git checkout v3.3.7

# We need to package every "theme/bootstrap.min.css", as well as the
# fonts directory. In the below, we are using a clever hack to include
# the fonts directory by echoing it first so `tar` at the end will know
# to copy it. We then want to find all bootstrap.min.css files, but
# need to ignore three directories: "2", "custom", and "bower_components".
#
# NOTE: the `echo` and `find` commands MUST be (in the same parentheses)
#
# You should be able to copy-paste this without the leading $
$ (echo "./fonts" && \
    find . -name "bootstrap.min.css" \
        -not -path "./2/*" \
        -not -path "./bower_components/*" \
        -not -path "./custom/*") | \
    xargs tar -cf ~/Desktop/bootswatch-flat-3.3.7.tar

# Now that we've extracted the relevant files, add them to the
# sphinx_bootstrap_theme repo
```

```
$ cd /path/to/sphinx-bootstrap-theme/sphinx_bootstrap_theme/bootstrap/static

# Make the directory relevant to your bootswatch version and enter it;
# the archive we made is not self-contained
$ mkdir bootswatch-3.3.7
$ cd bootswatch-3.3.7

# Extract the archive we just created here
$ cat ~/Desktop/bootswatch-flat-3.3.7.tar | tar -x

# Make sure the themes you were expecting, **AND** the fonts directory are here
$ ls
```

CHAPTER 2

Indices and tables

- `genindex`
- `search`

F

`format_exception()` (built-in function), [11](#)

M

`MyAnimal()` (class), [11](#)