
Sphinx RTD theme demo Documentation

Release 0.4.2

**Dave Snider, Read the Docs, Inc.
contributors**

Jan 26, 2019

1	Via Python Package	1
2	Via Git or Download	3
3	Configuration	5
3.1	Project-wide Configuration	5
3.2	Page-level Configuration	6
3.3	How the Table of Contents builds	6
4	Changelog	9
4.1	master	9
4.2	v0.4.2	9
4.3	v0.4.1	10
4.4	v0.4.0	10
4.5	v0.3.1	10
4.6	v0.3.0	11
4.7	v0.2.4	11
4.8	v0.2.3	12
4.9	v0.2.2	12
4.10	v0.2.1	12
4.11	v0.2.0	12
4.12	v0.1.10-alpha	12
4.13	v0.1.9	12
4.14	v0.1.8	13
5	Contributing or modifying the theme	15
5.1	Set up your environment	15
5.2	Releasing the Theme	16
6	Structural Elements	17
6.1	Document Section	18
7	Structural Elements 2	19
7.1	Document Section	19
8	Paragraph Level Markup	21
8.1	Inline Markup	22

8.2	Math	23
8.3	Meta	23
8.4	Blocks	23
8.5	Sidebar	25
8.6	References	26
8.7	Directives	27
8.8	Download Links	30
9	Lists & Tables	31
9.1	Lists	32
9.2	Tables	35
10	test_py_module	39
10.1	Generated Index	41
10.2	Optional parameter args	41
10.3	Data	41
11	Long Sticky Nav	43
11.1	Example Menu 1	44
11.2	Example Menu 2	44
11.3	Example Menu 3	44
11.4	Example Menu 4	44
11.5	Example Menu 5	45
11.6	Example Menu 6	45
11.7	Example Menu 7	45
11.8	Example Menu 8	45
11.9	Example Menu 9	45
11.10	Example Menu 10	45
11.11	Example Menu 11	45
11.12	Example Menu 12	45
11.13	Example Menu 13	45
11.14	Example Menu 14	45
11.15	Example Menu 15	46
11.16	Example Menu 16	46
11.17	Example Menu 17	46
11.18	Example Menu 18	46
11.19	Example Menu 19	46
11.20	Example Menu 20	46
11.21	Example Submenu 1	46
11.22	Example Submenu 2	47
	Python Module Index	49

CHAPTER 1

Via Python Package

Install the package (or add it to your `requirements.txt` file):

```
pip install sphinx_rtd_theme
```

In your `conf.py` file:

```
html_theme = "sphinx_rtd_theme"
```


CHAPTER 2

Via Git or Download

Symlink or subtree the `sphinx_rtd_theme/sphinx_rtd_theme` repository into your documentation at `docs/_themes/sphinx_rtd_theme` then add the following two settings to your Sphinx `conf.py` file:

```
html_theme = "sphinx_rtd_theme"
html_theme_path = ["_themes", ]
```


You can configure different parts of the theme.

3.1 Project-wide Configuration

3.1.1 HTML Theme Options

The theme's project-wide options are defined in the `sphinx_rtd_theme/theme.conf` file of this repository, and can be defined in your project's `conf.py` via `html_theme_options`. For example:

```
html_theme_options = {
    'canonical_url': '',
    'analytics_id': 'UA-XXXXXXX-1', # Provided by Google in your dashboard
    'logo_only': False,
    'display_version': True,
    'prev_next_buttons_location': 'bottom',
    'style_external_links': False,
    'vcs_pageview_mode': '',
    # Toc options
    'collapse_navigation': True,
    'sticky_navigation': True,
    'navigation_depth': 4,
    'includehidden': True,
    'titles_only': False
}
```

The following options are available:

Base options

- `canonical_url` String. This will specify a [canonical url](#) to let search engines know they should give higher ranking to latest version of the docs. The url points to the root of the documentation and requires a trailing slash.

- `analytics_id` String. Change the Google Analytics ID that is included on pages.
- `display_version` Bool. With this disabled, the version number isn't shown at the top of the sidebar.
- `prev_next_buttons_location` String. can take the value `bottom`, `top`, `both`, or `None` and will display the “Next” and “Previous” buttons accordingly.
- `style_external_links` Bool. Add an icon next to external links. Defaults to `False`.
- `vcs_pageview_mode` String. Changes how to view files when using `display_github`, `display_gitlab`, etc. When using Github or Gitlab this can be: `blob` (default), `edit`, or `raw`, on Bitbucket, this can be either: `view` (default) or `edit`.

TOC Options

These effect how we display the Table of Contents in the side bar. You can read more about them here: <http://www.sphinx-doc.org/en/stable/templating.html#toctree>

- `collapse_navigation` Bool. With this enabled, you will lose the `[+]` drop downs next to each section in the sidebar.
- `sticky_navigation` Bool. This causes the sidebar to scroll with the main page content as you scroll the page.
- `navigation_depth` Int. Indicate the max depth of the tree; by default, 4 levels are included; set it to -1 to allow unlimited depth.
- `includehidden` Bool. Specifies if the sidebar includes toctrees marked with the `:hidden:` option
- `titles_only` Bool. If True, removes headers within a page from the sidebar.

Note: Setting `collapse_navigation` to `False` and using a high `navigation_depth` can cause projects with many files and a deep file structure to generate HTML files that are significantly larger in file size and much longer compilation times.

3.1.2 HTML Context Options

TODO.

3.2 Page-level Configuration

Pages support metadata that changes how the theme renders. You can currently add the following:

- `:github_url:` This will force the “Edit on GitHub” to the configured URL
- `:bitbucket_url:` This will force the “Edit on Bitbucket” to the configured URL
- `:gitlab_url:` This will force the “Edit on GitLab” to the configured URL

3.3 How the Table of Contents builds

Currently the left menu will build based upon any `toctree(s)` defined in your `index.rst` file. It outputs 2 levels of depth, which should give your visitors a high level of access to your docs. If no toctrees are set the theme reverts to sphinx's usual local toctree.

It's important to note that if you don't follow the same styling for your rST headers across your documents, the toctree will misbuild, and the resulting menu might not show the correct depth when it renders.

Also note that by default the table of contents is set with `includehidden=True`. This allows you to set a hidden toc in your index file with the `:hidden:` property that will allow you to build a toc without it rendering in your index.

By default, the navigation will “stick” to the screen as you scroll. However if your toc is vertically too large, it will revert to static positioning. To disable the sticky nav altogether change the setting in `conf.py`.

4.1 master

Date TBD

4.1.1 New Features

4.1.2 Fixes

- Fix scrolling to active item in sidebar on load (#214)
- Style caption link for code and literal blocks
- Fix inconsistent font size and line height for autodoc “raises” and “returns” (#267)

4.1.3 Other Changes

4.2 v0.4.2

Date Oct 5, 2018

4.2.1 New Features

4.2.2 Fixes

- Set base font size on <html> (#668)
- Fix HTML search not working with Sphinx-1.8 (#672)

4.2.3 Other Changes

- Upload signed packages to PyPI with twine (#651)
- Do not enforce period at the end of copyright statement (666)

4.3 v0.4.1

Date July 27, 2018

4.3.1 New Features

4.3.2 Fixes

- Line height adjustments for Liberation Mono (#656)

4.3.3 Other Changes

- Add Sphinx as a dependency

4.4 v0.4.0

This version made some changes to how JS and CSS were included when the theme is used on Read the Docs.

4.4.1 New Features

4.4.2 Fixes

- Do not rely on readthedocs.org for CSS/JS (#614)
- Color accessibility improvements on the left navigation

4.4.3 Other Changes

- Write theme version and build date at top of JavaScript and CSS
- Changed code and literals to use a native font stack (#612)
- Fix small styling issues

4.5 v0.3.1

4.5.1 Fixes

- Revert part of #576 causing display issues with version selector menu
- Backwards compatibility fixes for pre-0.3.0 releases (#623)

- Fix mkd docs version selector (#622)
- Add open list spacing (#591)
- Fix table centering (#599)

4.6 v0.3.0

Note: this version resulted in some JavaScript incompatibilities when used on readthedocs.org

4.6.1 New Features

- Add html language attribute
- Allow setting ‘rel’ and ‘title’ attributes for stylesheets (#551)
- Add option to style external links
- Add github, gitlab, bitbucket page arguments option
- Add pygments support
- Add setuptools entry point allowing to use `sphinx_rtd_theme` as Sphinx `html_theme` directly.
- Add language to the JS output variable

4.6.2 Fixes

- Fix some HTML warnings and errors
- Fix many styling issues
- Fix many sidebar glitches
- Fix line number spacing to align with the code lines
- Hide Edit links on auto created pages
- Include missing font files with the theme

4.6.3 Other Changes

- Significant improvement of our documentation
- Compress our Javascript files
- Updated dependencies

4.7 v0.2.4

- Yet another patch to deal with extra builders outside Sphinx, such as the singlehtml builders from the Read the Docs Sphinx extension

4.8 v0.2.3

- Temporarily patch Sphinx issue with `singlehtml` builder by inspecting the builder in template.

4.9 v0.2.2

- Roll back toctree fix in 0.2.1 (#367). This didn't fix the issue and introduced another bug with toctrees display.

4.10 v0.2.1

- Add the `rel` HTML attribute to the footer links which point to the previous and next pages.
- Fix toctree issue caused by Sphinx `singlehtml` builder (#367)

4.11 v0.2.0

- Adds the `comments` block after the `body` block in the template
- Added “Edit on GitLab” support
- Many bug fixes

4.12 v0.1.10-alpha

Note: This is a pre-release version

- Removes Sphinx dependency
- Fixes hamburger on mobile display
- Adds a `body_begin` block to the template
- Added `prev_next_buttons_location`

4.13 v0.1.9

- Intermittent scrollbar visibility bug fixed. This change introduces a backwards incompatible change to the theme's layout HTML. This should only be a problem for derivative themes that have overridden styling of nav elements using direct descendant selectors. See [#215](#) for more information.
- Safari overscroll bug fixed
- Version added to the nav header
- Revision id was added to the documentation footer if you are using RTD
- An extra block, `extrafooter` was added to allow extra content in the document footer block
- Fixed modernizr URL

- Small display style changes on code blocks, figure captions, and nav elements

4.14 v0.1.8

- Start keeping changelog :)
- Support for third and fourth level headers in the sidebar
- Add support for Sphinx 1.3
- Add sidebar headers for :caption: in Sphinx toctree
- Clean up sidebar scrolling behavior so it never scrolls out of view

Contributing or modifying the theme

The `sphinx_rtd_theme` is primarily a `sass` project that requires a few other sass libraries. I'm using `bower` to manage these dependencies and `sass` to build the css. The good news is I have a very nice set of `grunt` operations that will not only load these dependencies, but watch for changes, rebuild the sphinx demo docs and build a distributable version of the theme. The bad news is this means you'll need to set up your environment similar to that of a front-end developer (vs. that of a python developer). That means installing node and ruby.

See also:

If you are unsure of appropriate actions to take while interacting with our community please read our Code of Conduct.

5.1 Set up your environment

1. Install `sphinx` into a virtual environment.

```
pip install sphinx sphinxcontrib-httpdomain
```

2. Install `sass`.

```
gem install sass
```

3. Install node, bower, grunt, and theme dependencies.

```
# Install node
brew install node

# Install bower and grunt
npm install -g bower grunt-cli

# Now that everything is installed, let's install the theme dependencies.
npm install
```

Now that our environment is set up, make sure you're in your virtual environment, go to this repository in your terminal and run `grunt`:

```
grunt
```

This default task will do the following **very cool things that make it worth the trouble**:

1. Install and update any bower dependencies.
2. Run sphinx and build new docs.
3. Watch for changes to the sass files and build css from the changes.
4. Rebuild the sphinx docs anytime it notices a change to `.rst`, `.html`, `.js` or `.css` files.

5.2 Releasing the Theme

When you release a new version, you should do the following:

1. Bump the version in `sphinx_rtd_theme/__init__.py`, `bower.json` and `package.json` – we try to follow [semver](#), so be careful with breaking changes.
2. Update the changelog (`docs/changelog.rst`) with the version information.
3. Run a `grunt build` to rebuild all the theme assets.
4. Commit that change.
5. Tag the release in git: `git tag $NEW_VERSION`.
6. Push the tag to GitHub: `git push --tags origin`.
7. Upload the package to PyPI:

```
$ rm -rf dist/
$ python setup.py sdist bdist_wheel
$ twine upload --sign --identity security@readthedocs.org dist/*
```

Structural Elements

Table of Contents

- *Structural Elements*
 - *Document Section*
 - * *Document Subsection*
 - *Document Subsubsection*
 - *Document Paragraph*
- *Structural Elements 2*
 - *Document Section*
 - * *Document Subsection*

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec lorem neque, interdum in ipsum nec, finibus dictum velit. Ut eu efficitur arcu, id aliquam erat. In sit amet diam gravida, imperdiet tellus eu, gravida nisl. Praesent aliquet odio eget libero elementum, quis rhoncus tellus tincidunt. Suspendisse quis volutpat ipsum. Sed lobortis scelerisque tristique. Aenean condimentum risus tellus, quis accumsan ipsum laoreet ut. Integer porttitor maximus suscipit. Mauris in posuere sapien. Aliquam accumsan feugiat ligula, nec fringilla libero commodo sed. Proin et erat pharetra.

Etiam turpis ante, luctus sed velit tristique, finibus volutpat dui. Nam sagittis vel ante nec malesuada. Praesent dignissim mi nec ornare elementum. Nunc eu augue vel sem dignissim cursus sed et nulla. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Pellentesque dictum dui sem, non placerat tortor rhoncus in. Sed placerat nulla at rhoncus iaculis.

6.1 Document Section

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed condimentum nulla vel neque venenatis, nec placerat lorem placerat. Cras purus eros, gravida vitae tincidunt id, vehicula nec nulla. Fusce aliquet auctor cursus. Phasellus ex neque, vestibulum non est vitae, viverra fringilla tortor. Donec vestibulum convallis justo, a faucibus lorem vulputate vel. Aliquam cursus odio eu felis sodales aliquet. Aliquam erat volutpat. Maecenas eget dictum mauris. Suspendisse arcu eros, condimentum eget risus sed, luctus efficitur arcu. Cras ut dictum mi. Nulla congue interdum lorem, semper semper enim commodo nec.

6.1.1 Document Subsection

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam efficitur in eros et blandit. Nunc maximus, nisl at auctor vestibulum, justo ex sollicitudin ligula, id faucibus urna orci tristique nisl. Duis auctor rutrum orci, in ornare lacus condimentum quis. Quisque arcu velit, facilisis quis interdum ac, hendrerit auctor mauris. Curabitur urna nibh, porttitor at ante sit amet, vestibulum interdum dolor. Duis dictum elit orci, tincidunt imperdiet sem pellentesque et. In vehicula pellentesque varius. Phasellus a turpis sollicitudin, bibendum massa et, imperdiet neque. Integer quis sapien in magna rutrum bibendum. Integer cursus ex sed magna vehicula finibus. Proin tempus orci quis dolor tempus, nec condimentum odio vestibulum. Etiam efficitur sollicitudin libero, tincidunt volutpat ligula interdum sed.

Document Subsubsection

Donec non rutrum lorem. Aenean sagittis metus at pharetra fringilla. Nunc sapien dolor, cursus sed nisi at, pretium tristique lectus. Sed pellentesque leo lectus, et convallis ipsum euismod a. Integer at leo vitae felis pretium aliquam fringilla quis odio. Sed pharetra enim accumsan feugiat pretium. Maecenas at pharetra tortor. Morbi semper eget mi vel finibus. Cras rutrum nulla eros, id feugiat arcu pellentesque ut. Sed finibus tortor ac nisi ultrices viverra. Duis feugiat malesuada sapien, at commodo ante porttitor ac. Curabitur posuere mauris mi, vel ornare orci scelerisque sit amet. Suspendisse nec fringilla dui.

Document Paragraph

Pellentesque nec est in odio ultrices elementum. Vestibulum et hendrerit sapien, quis vulputate turpis. Suspendisse potenti. Curabitur tristique sit amet lectus non viverra. Phasellus rutrum dapibus turpis sed imperdiet. Mauris maximus viverra ante. Donec eu egestas mauris. Morbi vulputate tincidunt euismod. Integer vel porttitor neque. Donec at lacus suscipit, lacinia lectus vel, sagittis lectus.

Structural Elements 2

Etiam turpis ante, luctus sed velit tristique, finibus volutpat dui. Nam sagittis vel ante nec malesuada. Praesent dignissim mi nec ornare elementum. Nunc eu augue vel sem dignissim cursus sed et nulla. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Pellentesque dictum dui sem, non placerat tortor rhoncus in. Sed placerat nulla at rhoncus iaculis.

7.1 Document Section

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed condimentum nulla vel neque venenatis, nec placerat lorem placerat. Cras purus eros, gravida vitae tincidunt id, vehicula nec nulla. Fusce aliquet auctor cursus. Phasellus ex neque, vestibulum non est vitae, viverra fringilla tortor. Donec vestibulum convallis justo, a faucibus lorem vulputate vel. Aliquam cursus odio eu felis sodales aliquet. Aliquam erat volutpat. Maecenas eget dictum mauris. Suspendisse arcu eros, condimentum eget risus sed, luctus efficitur arcu. Cras ut dictum mi. Nulla congue interdum lorem, semper semper enim commodo nec.

7.1.1 Document Subsection

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam efficitur in eros et blandit. Nunc maximus, nisl at auctor vestibulum, justo ex sollicitudin ligula, id faucibus urna orci tristique nisl. Duis auctor rutrum orci, in ornare lacus condimentum quis. Quisque arcu velit, facilisis quis interdum ac, hendrerit auctor mauris. Curabitur urna nibh, porttitor at ante sit amet, vestibulum interdum dolor. Duis dictum elit orci, tincidunt imperdiet sem pellentesque et. In vehicula pellentesque varius. Phasellus a turpis sollicitudin, bibendum massa et, imperdiet neque. Integer quis sapien in magna rutrum bibendum. Integer cursus ex sed magna vehicula finibus. Proin tempus orci quis dolor tempus, nec condimentum odio vestibulum. Etiam efficitur sollicitudin libero, tincidunt volutpat ligula interdum sed. Praesent congue sagittis nisl et suscipit. Vivamus sagittis risus et egestas commodo. Cras venenatis arcu in pharetra interdum. Donec quis metus porttitor tellus cursus lobortis. Quisque et orci magna. Fusce rhoncus mi mi,



Fig. 1: This is a caption for a figure. Text should wrap around the caption.

at vehicula massa rhoncus quis. Mauris augue leo, pretium eget molestie vitae, efficitur nec nulla. In hac habitasse platea dictumst. Sed sit amet imperdiet purus.

Paragraph Level Markup

Table of Contents

- *Paragraph Level Markup*
 - *Inline Markup*
 - *Math*
 - *Meta*
 - *Blocks*
 - * *Literal Blocks*
 - * *Line Blocks*
 - * *Block Quotes*
 - * *Doctest Blocks*
 - * *Code Blocks*
 - *Emphasized lines with line numbers*
 - *Sidebar*
 - * *Code with Sidebar*
 - *References*
 - * *Footnotes*
 - * *Citations*
 - * *Glossary*
 - * *Targets*
 - *Directives*

- * *Contents*
- * *Centered text*
- * *Images & Figures*
 - *Images*
 - *Figures*
- * *Admonitions*
- * *Topics, Sidebars, and Rubrics*
- * *Target Footnotes*
- * *Replacement Text*
- * *Compound Paragraph*
- *Download Links*

8.1 Inline Markup

Paragraphs contain text and may contain inline markup: *emphasis*, **strong emphasis**, inline literals, standalone hyperlinks (<http://www.python.org>), external hyperlinks ([Python⁵](#)), internal cross-references ([example](#)), external hyperlinks with embedded URIs ([Python web site](#)), footnote references (manually numbered¹, anonymous auto-numbered³, labeled auto-numbered², or symbolic^{*0}), citation references (¹²), substitution references ([☐](#)), and inline hyperlink targets (see [Targets](#) below for a reference back to here). Character-level inline markup is also possible (although exceedingly ugly!) in *reStructuredText*. Problems are indicated by **problematic** text (generated by processing errors; this one is intentional).

Also with `sphinx.ext.autodoc`, which I use in the demo, I can link to `test_py_module.test.Foo`. It will link you right my code documentation for it.

The default role for interpreted text is *Title Reference*. Here are some explicit interpreted text roles: a PEP reference (**PEP 287**); an RFC reference (**RFC 2822**); a `subscript`; a `superscript`; and explicit roles for *standard inline* markup.

GUI labels are a useful way to indicate that *Some action* is to be taken by the user. The GUI label should not run over line-height so as not to *interfere* with text from adjacent lines.

Key-bindings indicate that the read is to press a button on the keyboard or mouse, for example MMB and Shift-MMB. Another useful markup to indicate a user action is to use `menuselection` this can be used to show short and long menus in software. For example, and `menuselection` can be seen here that breaks is too long to fit on this line. `My → Software → Some menu → Some sub menu 1 → sub menu 2`.

Let's test wrapping and whitespace significance in inline literals: This is an example of `--inline-literal --text, --including some-- strangely--hyphenated-words`. Adjust-the-width-of-your-browser-window to see how the text is wrapped. `-- ----` Now note the spacing between the words of this sentence (words should be grouped in pairs).

⁵ <http://www.python.org/>

¹ A footnote contains body elements, consistently indented by at least 3 spaces.

This is the footnote's second paragraph.

³ This footnote is numbered automatically and anonymously using a label of “#” only.

² Footnotes may be numbered, either manually (as in¹) or automatically using a “#”-prefixed label. This footnote has a label so it can be referred to from multiple places, both as a footnote reference (²) and as a hyperlink reference ([label](#)).

⁰ Footnotes may also use symbols, specified with a “*” label. Here's a reference to the next footnote:†⁰.

¹² This citation has some `code` blocks in it, maybe some **bold** and *italics* too. Heck, lets put a link to a meta citation¹³ too.

If the `--pep-references` option was supplied, there should be a live link to PEP 258 here.

8.2 Math

This is a test. Here is an equation: $X_{0:5} = (X_0, X_1, X_2, X_3, X_4)$. Here is another:

$$\nabla^2 f = \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial f}{\partial r} \right) + \frac{1}{r^2 \sin \theta} \frac{\partial f}{\partial \theta} \left(\sin \theta \frac{\partial f}{\partial \theta} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2 f}{\partial \phi^2} \quad (8.1)$$

You can add a link to equations like the one above (8.1) by using `:eq:`.

8.3 Meta

8.4 Blocks

8.4.1 Literal Blocks

Literal blocks are indicated with a double-colon (“::”) at the end of the preceding paragraph (over there -->). They can be indented:

```
if literal_block:
    text = 'is left as-is'
    spaces_and_linebreaks = 'are preserved'
    markup_processing = None
```

Or they can be quoted without indentation:

```
>> Great idea!
>
> Why didn't I think of that?
```

8.4.2 Line Blocks

This is a line block. It ends with a blank line.

Each new line begins with a vertical bar (“|”).

Line breaks and initial indents are preserved.

Continuation lines are wrapped portions of long lines; they begin with a space in place of the vertical bar.

The left edge of a continuation line need not be aligned with the left edge of the text above it.

This is a second line block.

Blank lines are permitted internally, but they must begin with a “|”.

Take it away, Eric the Orchestra Leader!

A one, two, a one two three four

Half a bee, philosophically,
 must, *ipso facto*, half not be.
But half the bee has got to be,
 vis a vis its entity. D’you see?

But can a bee be said to be
 or not to be an entire bee,
 when half the bee is not a bee,
 due to some ancient injury?

Singing...

8.4.3 Block Quotes

Block quotes consist of indented body elements:

My theory by A. Elk. Brackets Miss, brackets. This theory goes as follows and begins now. All brontosaurus are thin at one end, much much thicker in the middle and then thin again at the far end. That is my theory, it is mine, and belongs to me and I own it, and what it is too.

—Anne Elk (Miss)

8.4.4 Doctest Blocks

```
>>> print 'Python-specific usage examples; begun with ">>>"  
Python-specific usage examples; begun with ">>>"  
>>> print '(cut and pasted from interactive Python sessions)'  
(cut and pasted from interactive Python sessions)
```

8.4.5 Code Blocks

```
# parsed-literal test  
curl -O http://someurl/release-0.4.2.tar-gz
```

Listing 1: Code Blocks can have captions.

```
{  
"windows": [  
  {  
    "panes": [  
      {  
        "shell_command": [  
          "echo 'did you know'",  
          "echo 'you can inline'"  
        ]  
      },  
      {  
        "shell_command": "echo 'single commands'"  
      },  
      "echo 'for panes'"  
    ]  
  },  
  ]  
}
```

(continues on next page)

(continued from previous page)

```

    "window_name": "long form"
  }
],
"session_name": "shorthands"
}

```

Emphasized lines with line numbers

```

1 def some_function():
2     interesting = False
3     print 'This line is highlighted.'
4     print 'This one is not...'
5     print '...but this one is.'

```

8.5 Sidebar

Ch'ien / The Creative



Above CH'IEN THE CREATIVE, HEAVEN

Below CH'IEN THE CREATIVE, HEAVEN

The first hexagram is made up of six unbroken lines. These unbroken lines stand for the primal power, which is light-giving, active, strong, and of the spirit. The hexagram is consistently strong in character, and since it is without weakness, its essence is power or energy. Its image is heaven. Its energy is represented as unrestricted by any fixed conditions in space and is therefore conceived of as motion. Time is regarded as the basis of this motion. Thus the hexagram includes also the power of time and the power of persisting in time, that is, duration.

The power represented by the hexagram is to be interpreted in a dual sense in terms of its action on the universe and of its action on the world of men. In relation to the universe, the hexagram expresses the strong, creative action of the Deity. In relation to the human world, it denotes the creative action of the holy man or sage, of the ruler or leader of men, who through his power awakens and develops their higher nature.

8.5.1 Code with Sidebar

A code example

With a sidebar on the right.

Listing 2: Literal includes can also have captions.

```
1 # -*- coding: utf-8 -*-
2 """Test Module for sphinx_rtd_theme."""
3
4
5 class Foo:
6
7     """Docstring for class Foo.
8
9     This text tests for the formatting of docstrings generated from output
10     ``sphinx.ext.autodoc``. Which contain reST, but sphinx nests it in the
11     ``<dl>``, and ``<dt>`` tags. Also, ``<tt>`` is used for class, method names
12     and etc, but those will *always* have the ``.descname`` or
13     ``.descclassname`` class.
14
15     Normal ``<tt>`` (like the <tt> I just wrote here) needs to be shown with
16     the same style as anything else with ````this type of markup```.
17
18     It's common for programmers to give a code example inside of their
19     docstring::
20
21         from test_py_module import Foo
22
23         myclass = Foo()
24         myclass.dothismethod('with this argument')
25         myclass.flush()
26
27         print(myclass)
28
29     Here is a link to :py:meth:`capitalize`.
30     Here is a link to :py:meth:`__init__`.
31
32     """
33
34     #: Doc comment for class attribute Foo.bar.
35     #: It can have multiple lines.
36     bar = 1
37
38
39     flox = 1.5    #: Doc comment for Foo.flox. One line only.
40
```

8.6 References

8.6.1 Footnotes

8.6.2 Citations

Here's a reference to the above,¹², and a [nonexistent] citation.

Here is another type of citation: *citation*

8.6.3 Glossary

This is a glossary with definition terms for thing like *Writing*:

Documentation Provides users with the knowledge they need to use something.

Reading The process of taking information into ones mind through the use of eyes.

Writing The process of putting thoughts into a medium for other people to *read*.

8.6.4 Targets

This paragraph is pointed to by the explicit “example” target. A reference can be found under *Inline Markup*, above. *Inline hyperlink targets* are also possible.

Section headers are implicit targets, referred to by name. See *Targets*, which is a subsection of ‘**Body Elements**’.

Explicit external targets are interpolated into references such as “Python⁵”.

Targets may be indirect and anonymous. Thus *this phrase* may also refer to the *Targets* section.

Here’s a ‘**hyperlink reference without a target**’, which generates an error.

8.7 Directives

8.7.1 Contents

These are just a sample of the many reStructuredText Directives. For others, please see: <http://docutils.sourceforge.net/docs/ref/rst/directives.html>.

8.7.2 Centered text

You can create a statement with centered text with `.. centered::`

This is centered text!

8.7.3 Images & Figures

Images



An image directive (also clickable – a hyperlink reference):

Figures

A figure directive with center alignment



Fig. 1: A figure is an image with a caption and/or a legend:

re	Revised, revisited, based on ‘re’ module.
Structured	Structure-enhanced text, structuredtext.
Text	Well it is, isn’t it?

This paragraph is also part of the legend.



Fig. 2: This caption should be centered.

8.7.4 Admonitions

Attention: Directives at large.

Caution: Don’t take any wooden nickels.

Danger: Mad scientist at work!

Error: Does not compute.

Hint: It’s bigger than a bread box.

Important:

- Wash behind your ears.
- Clean up your room.
 - Including the closet.
 - The bathroom too.
 - * Take the trash out of the bathroom.

- * Clean the sink.
 - Call your mother.
 - Back up your data.
-

Note: This is a note. Equations within a note: $G_{\mu\nu} = 8\pi G(T_{\mu\nu} + \rho_{\Lambda}g_{\mu\nu})$.

Tip: 15% if the service is good.

Example
Thing1
Thing2
Thing3

Warning: Strong prose may provoke extreme mental exertion. Reader discretion is strongly advised.

And, by the way...

You can make up your own admonition too.

8.7.5 Topics, Sidebars, and Rubrics

Sidebar Title

Optional Subtitle

This is a sidebar. It is for text outside the flow of the main text.

This is a rubric inside a sidebar

Sidebars often appears beside the main text with a border and background color.

Topic Title

This is a topic.

This is a rubric

8.7.6 Target Footnotes

8.7.7 Replacement Text

I recommend you try `Python`, *the* best language around⁵.

8.7.8 Compound Paragraph

This paragraph contains a literal block:

```
Connecting... OK
Transmitting data... OK
Disconnecting... OK
```

and thus consists of a simple paragraph, a literal block, and another simple paragraph. Nonetheless it is semantically *one* paragraph.

This construct is called a *compound paragraph* and can be produced with the “compound” directive.

8.8 Download Links

This long long long long long long long long long long long long long long long
long download link should be blue, normal weight text with a leading icon, and
should wrap white-spaces

Table of Contents

- *Lists & Tables*
 - *Lists*
 - * *Enumerated Lists*
 - * *Definition Lists*
 - * *Option Lists*
 - * *Field list*
 - * *Bullet Lists*
 - *Second list level*
 - *But deeper down the rabbit hole*
 - * *Hlists*
 - * *Numbered List*
 - *Tables*
 - * *Grid Tables*
 - *Giant Tables*
 - * *List Tables*

9.1 Lists

9.1.1 Enumerated Lists

1. Arabic numerals.
 - (a) lower alpha)
 - i. (lower roman)
 - A. upper alpha.
 - B. upper roman)
2. Lists that don't start at 1:
 - (c) Three
 - (d) Four
 - (c) C
 - (d) D
 - (c) iii
 - (d) iv
3. List items may also be auto-enumerated.

9.1.2 Definition Lists

Term Definition

Term [classifier] Definition paragraph 1.
Definition paragraph 2.

Term Definition

9.1.3 Option Lists

For listing command-line options:

-a	command-line option “a”
-b file	options can have arguments and long descriptions
--long	options can be long also
--input=file	long options can also have arguments
--very-long-option	The description can also start on the next line. The description may contain multiple body elements, regardless of where it starts.
-x, -y, -z	Multiple options are an “option group”.
-v, --verbose	Commonly-seen: short & long options.
-1 file, --one=file, --two file	Multiple options with arguments.
/V	DOS/VMS-style options too

There must be at least two spaces between the option and the description.

9.1.4 Field list

Author David Goodger

Address 123 Example Street Example, EX Canada A1B 2C3

Contact docutils-develop@lists.sourceforge.net

Authors Me; Myself; I

organization humankind

date \$Date: 2012-01-03 19:23:53 +0000 (Tue, 03 Jan 2012) \$

status This is a “work in progress”

revision \$Revision: 7302 \$

version 1

copyright This document has been placed in the public domain. You may do with it as you wish. You may copy, modify, redistribute, reattribute, sell, buy, rent, lease, destroy, or improve it, quote it at length, excerpt, incorporate, collate, fold, staple, or mutilate it, or do anything else to it that your or anyone else’s heart desires.

field name This is a generic bibliographic field.

field name 2 Generic bibliographic fields may contain multiple body elements.

Like this.

Dedication For Docutils users & co-developers.

abstract This document is a demonstration of the reStructuredText markup language, containing examples of all basic reStructuredText constructs and many advanced constructs.

9.1.5 Bullet Lists

- A bullet list
 - Nested bullet list.
 - Nested item 2.

- Item 2.

Paragraph 2 of item 2.

- Nested bullet list.
 - Nested item 2.
 - * Third level.
 - * Item 2.
 - Nested item 3.
- `inline literal`
 - `inline literal`
 - `inline literal`

Second list level

- here is a list in a second-level section.
 - yahoo
 - yahoo
 - yahoo
 - here is an inner bullet oh
 - * one more with an inline literally. yahoo
- heh heh. child. try to beat this embed:

```
1  # -*- coding: utf-8 -*-
2  """Test Module for sphinx_rtd_theme."""
3
4
5  class Foo:
6
7      """Docstring for class Foo.
8
9      This text tests for the formatting of docstrings generated from_
↪output
10     ``sphinx.ext.autodoc``. Which contain reST, but sphinx nests it in_
↪the
```

- and another. yahoo
 - yahoo
 - hi
- and hehe

But deeper down the rabbit hole

- I kept saying that, “deeper down the rabbit hole”. yahoo
 - I cackle at night yahoo.
- I’m so lonely here in GZ guangzhou
- A man of python destiny, hopes and dreams. yahoo
 - yahoo
 - * yahoo hi
 - * destiny

9.1.6 Hlists

- First item
- Second item
- Third item
- Forth item
- Fifth item
- Sixths item

Hlist with images



Fig. 1: This is a short caption for a figure.

-



Fig. 2: This is a long caption for a figure. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec porttitor dolor in odio posuere, vitae ornare libero mattis. In lobortis justo vestibulum nibh aliquet, non.

-

9.1.7 Numbered List

1. One,
2. Two.
3. Three with long text. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed feugiat sagittis neque quis eleifend. Duis rutrum lectus sit amet mattis suscipit.
 - 1. Using bullets and letters. (A)
 - 2. Using bullets and letters. (B)
 - 3. Using bullets and letters. (C)

9.2 Tables

9.2.1 Grid Tables

Here's a grid table followed by a simple table:

Header row, column 1 (header rows optional)	Header 2	Header 3	Header 4
body row 1, column 1	column 2	column 3	column 4
body row 2	Cells may span columns.		
body row 3	Cells may span rows.	• Table cells • contain • body elements.	
body row 4			
body row 5	Cells may also be empty: -->		

Inputs		Output
A	B	A or B
False	False	False
True	False	True
False	True	True
True	True	True

Giant Tables

Header 1	Header 2	Header 3	Header 1	Header 2	Header 3	Header 1	Header 2	Header 3	Header 1	Header 2	Header 3
body row 1	column 2	column 3	body row 1	column 2	column 3	body row 1	column 2	column 3	body row 1	column 2	column 3
body row 1	column 2	column 3	body row 1	column 2	column 3	body row 1	column 2	column 3	body row 1	column 2	column 3
body row 1	column 2	column 3	body row 1	column 2	column 3	body row 1	column 2	column 3	body row 1	column 2	column 3
body row 1	column 2	column 3	body row 1	column 2	column 3	body row 1	column 2	column 3	body row 1	column 2	column 3

9.2.2 List Tables

Table 1: List tables can have captions like this one.

List table	Header 1	Header 2	Header 3 long. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam sit amet mauris arcu.	
Stub 1	Row 1	Column 2	Column 3 long. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam sit amet mauris arcu.	
Stub 2	Row 2	Column 2	Column 3 long. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam sit amet mauris arcu.	
Stub 3	Row 3	Column 2	Column 3 long. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam sit amet mauris arcu.	

Table 2: This is a list table with images in it.

 Fig. 3: This is a short caption for a figure.	 Fig. 4: This is a long caption for a figure. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec port-titor dolor in odio posuere, vitae ornare libero mattis. In lobortis justo vestibulum nibh aliquet, non.
---	---

CHAPTER 10

test_py_module

Table of Contents

- `test_py_module`
 - *Generated Index*
 - *Optional parameter args*
 - *Data*

Test Module for sphinx_rtd_theme.

```
class test_py_module.test.Foo(qux, spam=False)
```

Docstring for class Foo.

This text tests for the formatting of docstrings generated from output `sphinx.ext.autodoc`. Which contain reST, but sphinx nests it in the `<dl>`, and `<dt>` tags. Also, `<tt>` is used for class, method names and etc, but those will *always* have the `.descname` or `.descclassname` class.

Normal `<tt>` (like the `<tt>` I just wrote here) needs to be shown with the same style as anything else with ```this type of markup```.

It's common for programmers to give a code example inside of their docstring:

```
from test_py_module import Foo

myclass = Foo()
myclass.dothismethod('with this argument')
myclass.flush()

print(myclass)
```

Here is a link to `capitalize()`. Here is a link to `__init__()`.

```
__init__(qux, spam=False)
    Start the Foo.
```

Parameters

- **qux** (*string*) – The first argument to initialize class.
- **spam** (*bool*) – Spam me yes or no...

add (*val1*, *val2*)

Return the added values.

Parameters

- **val1** (*int*) – First number to add.
- **val2** (*int*) – Second number to add.

Return type `int`**another_function** (*a*, *b*, ***kwargs*)

Here is another function.

Parameters

- **a** (*int*) – The number of green hats you own.
- **b** (*int*) – The number of non-green hats you own.
- **kwargs** (*float*) – Additional keyword arguments. Each keyword parameter should specify the name of your favorite cuisine. The values should be floats, specifying the mean price of your favorite dish in that cooking style.

Returns A 2-tuple. The first element is the mean price of all dishes across cuisines. The second element is the total number of hats you own: $a + b$.**Return type** `tuple`**Raises** **ValueError** – When *a* is not an integer.

New in version 1.0: This was added in 1.0

Changed in version 2.0: This was changed in 2.0

Deprecated since version 3.0: This is deprecated since 3.0

bar = 1

Doc comment for class attribute Foo.bar. It can have multiple lines.

baz = 2

Docstring for class attribute Foo.baz.

capitalize (*myvalue*)

Return a string as uppercase.

Parameters **myvalue** (*string*) – String to change**Return type** `string`**flox** = 1.5

Doc comment for Foo.flox. One line only.

qux = None

Doc comment for instance attribute qux.

spam = None

Docstring for instance attribute spam.

10.1 Generated Index

Part of the sphinx build process in generate and index file: genindex.

10.2 Optional parameter args

At this point optional parameters *cannot be generated from code*. However, some projects will manually do it, like so:

This example comes from [django-payments module docs](#).

```
class payments.dotpay.DotpayProvider(seller_id, pin[, channel=0[, lock=False ], lang='pl' ])  
    This backend implements payments using a popular Polish gateway, Dotpay.pl.
```

Due to API limitations there is no support for transferring purchased items.

Parameters

- **seller_id** – Seller ID assigned by Dotpay
- **pin** – PIN assigned by Dotpay
- **channel** – Default payment channel (consult reference guide)
- **lang** – UI language
- **lock** – Whether to disable channels other than the default selected above

10.3 Data

```
test_py_module.test.Data_item_1  
test_py_module.test.Data_item_2  
test_py_module.test.Data_item_3
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce congue elit eu hendrerit mattis.

Some data link [Data_item_1](#).

CHAPTER 11

Long Sticky Nav

Table of Contents

- *Long Sticky Nav*
 - *Example Menu 1*
 - *Example Menu 2*
 - *Example Menu 3*
 - *Example Menu 4*
 - *Example Menu 5*
 - *Example Menu 6*
 - *Example Menu 7*
 - *Example Menu 8*
 - *Example Menu 9*
 - *Example Menu 10*
 - *Example Menu 11*
 - *Example Menu 12*
 - *Example Menu 13*
 - *Example Menu 14*
 - *Example Menu 15*
 - *Example Menu 16*
 - *Example Menu 17*
 - *Example Menu 18*

- *Example Menu 19*
- *Example Menu 20*
- *Example Submenu 1*
 - * *Submenu 1*
 - *Subsubmenu 1*
 - *Subsubmenu 2*
 - * *Submenu 2*
 - *Subsubmenu 1*
 - * *Submenu 3*
 - * *Submenu 4*
 - * *Submenu 5*
- *Example Submenu 2*
 - * *Submenu 1*
 - *Subsubmenu 1*
 - * *Submenu 2*
 - *Subsubmenu 1*
 - * *Submenu 3*
 - * *Submenu 4*
 - * *Submenu 5*

This section demonstrates how the ‘sticky_navigation’ setting behaves when the menu is very long. When this section is selected, it will make the menu and the main area scroll when you are at the top of the page.

11.1 Example Menu 1

Just a place holder. . .

11.2 Example Menu 2

Just a place holder. . .

11.3 Example Menu 3

Just a place holder. . .

11.4 Example Menu 4

Just a place holder. . .

11.5 Example Menu 5

Just a place holder. . .

11.6 Example Menu 6

Just a place holder. . .

11.7 Example Menu 7

Just a place holder. . .

11.8 Example Menu 8

Just a place holder. . .

11.9 Example Menu 9

Just a place holder. . .

11.10 Example Menu 10

Just a place holder. . .

11.11 Example Menu 11

Just a place holder. . .

11.12 Example Menu 12

Just a place holder. . .

11.13 Example Menu 13

Just a place holder. . .

11.14 Example Menu 14

Just a place holder. . .

11.15 Example Menu 15

Just a place holder. . .

11.16 Example Menu 16

Just a place holder. . .

11.17 Example Menu 17

Just a place holder. . .

11.18 Example Menu 18

Just a place holder. . .

11.19 Example Menu 19

Just a place holder. . .

11.20 Example Menu 20

Just a place holder. . .

11.21 Example Submenu 1

Just a place holder. . .

11.21.1 Submenu 1

Just a place holder. . .

Subsubmenu 1

Just a place holder. . .

Subsubmenu 2

Just a place holder. . .

11.21.2 Submenu 2

Just a place holder. . .

Subsubmenu 1

Just a place holder. . .

11.21.3 Submenu 3

Just a place holder. . .

11.21.4 Submenu 4

Just a place holder. . .

11.21.5 Submenu 5

Just a place holder. . .

11.22 Example Submenu 2

Just a place holder. . .

11.22.1 Submenu 1

Just a place holder. . .

Subsubmenu 1

Just a place holder. . .

11.22.2 Submenu 2

Just a place holder. . .

Subsubmenu 1

Just a place holder. . .

11.22.3 Submenu 3

Just a place holder. . .

11.22.4 Submenu 4

Just a place holder. . .

11.22.5 Submenu 5

Just a place holder. . .

t

`test_py_module.test`, [39](#)

Symbols

`__init__()` (`test_py_module.test.Foo` method), [39](#)

A

`add()` (`test_py_module.test.Foo` method), [40](#)

`another_function()` (`test_py_module.test.Foo` method), [40](#)

B

`bar` (`test_py_module.test.Foo` attribute), [40](#)

`baz` (`test_py_module.test.Foo` attribute), [40](#)

C

`capitalize()` (`test_py_module.test.Foo` method), [40](#)

D

`Data_item_1` (in module `test_py_module.test`), [41](#)

`Data_item_2` (in module `test_py_module.test`), [41](#)

`Data_item_3` (in module `test_py_module.test`), [41](#)

Documentation, [27](#)

F

`flox` (`test_py_module.test.Foo` attribute), [40](#)

`Foo` (class in `test_py_module.test`), [39](#)

P

`payments.dotpay.DotpayProvider` (class in `test_py_module.test`), [41](#)

Python Enhancement Proposals
PEP 287, [22](#)

Q

`qux` (`test_py_module.test.Foo` attribute), [40](#)

R

Reading, [27](#)

RFC

RFC 2822, [22](#)

S

`spam` (`test_py_module.test.Foo` attribute), [40](#)

T

`test_py_module.test` (module), [39](#)

W

Writing, [27](#)