

```
usepackagecharter
usepackage[defaultsans]lato
usepackageinconsolata
```

---

# **Sphinx RTD Theme Template**

***Release 1.000***

**Jul 27, 2019**



---

## Contents:

---

<b>1</b>	<b>Changes from the standard Sphinx theme</b>	<b>3</b>
<b>2</b>	<b>Resulting visual changes</b>	<b>5</b>
<b>3</b>	<b>To Use</b>	<b>7</b>
<b>4</b>	<b>Prerequisites</b>	<b>9</b>
	<b>Python Module Index</b>	<b>31</b>
	<b>Index</b>	<b>33</b>







---

## Changes from the standard Sphinx theme

---

1. Added *Sphinx RTD Theme*
2. Added a number of useful built in *Sphinx* extensions
  - a. intersphinx
  - b. autodoc
  - c. mathjax
  - d. viewcode
  - e. todo
  - f. githubpages
  - g. napoleon - Allows parses NumPy and Google style docstrings
3. Added reference to use *Pygments* code syntax highlighter (note if you want to modify this theme copy the `pygments.css` file into `../docs/source/html/_static/css` directory and change any of the values and it will overwrite the default *Pygment* highlighting theme options)
4. Added reference to an additional theme (*Cloud Sphinx Theme*) that provides for better table formatting options (delete the reference to the extension in the Sphinx config file (`conf.py`) if you do not want to install/use this)
5. Sets a number of default html theme options, such as showing the version in the title, adding next/previous buttons top and bottom, etc
6. Changed default *code block* and `inline code` fonts to **Inconsolata** and bumped up font size from **12px to 14px** for a clearer/crisper layout via a custom `layout.html` file and `custom.css` file, added word-wrap in code blocks
7. Changed default *description* fonts to **Inconsolata** and bumped up font size from **14px to 16px** for a clearer layout, via custom `layout.html` file and `custom.css` file. Where the font size for the descriptions now more closely match the other default font sizes used for the typical body text (its much smaller by default)



8. Changed default max page width from **800px to 1200px** max to better suit displaying both more text and longer code lines via custom `custom.css` file (suits 150 characters per line which is my default goto style). This is based on the following code style:-

**Black Code** style modified to use max 150 characters per line and preferred single quotes using the following settings:-

Listing 1: Visual Studio Code settings (settings.json)

```
"python.formatting.provider": "black",
"python.formatting.blackArgs": [
    "--line-length=150",
    "--skip-string-normalization",
],
```

Resulting visual changes

Citation in text [Blah2015].  
.. [Blah2015] This is a reference, 2015.

Footnotes

The footnote section needs to be added at the end ...

Footnote [#?].  
.. comment:: ...  
.. rubric:: Footnotes  
.. [#?] Footnote text ...

Footnote <sup>1</sup>

Footnotes

1

Footnote text ...

class main.AClass [source]

Bases: object

Class docstring, with reference to the module, or another class module.AnotherClass and its function module.AnotherClass.foo().

class main.AnotherClass [source]

Bases: object

Anotherclass docstring.

foo(arg1, arg2) [source]

A method's docstring with parameters and return value.

Use all the cool Sphinx capabilities in this description, e.g. to give usage examples ...

Example

ORIGINAL FORMATTING

Citation in text [Blah2015].  
.. [Blah2015] This is a reference, 2015.

Footnotes

The footnote section needs to be added at the end ...

Footnote [#?].  
.. comment:: ...  
.. rubric:: Footnotes  
.. [#?] Footnote text ...

Footnote <sup>1</sup>

Footnotes

1

Footnote text ...

class main.AClass [source]

Bases: object

Class docstring, with reference to the module, or another class module.AnotherClass and its function module.AnotherClass.foo().

class main.AnotherClass [source]

Bases: object

Anotherclass docstring.

foo(arg1, arg2) [source]

A method's docstring with parameters and return value.

Use all the cool Sphinx capabilities in this description, e.g. to give usage examples ...

Example

NEW FORMATTING

5



- Copy the `docs` folder to your repository
- Modify `conf.py` line `sys.path.insert(0, os.path.abspath('../..'))` to the path(s) where modules are stored that you want to be processed on the basis of docstrings within the modules themselves
- Run `make html` at the command prompt from within the `../docs` directory to build html files
- View the built html files within the `_build` directory
- Note if you subsequently modify the `custom.css` file then you may need to run `make clean` followed by `make html` to pickup the css changes
- Review demo files under `_demo` directory for examples of formatting, etc. This directory can be removed, and references to it removed from `index.rst` file. Note the example content is taken from the official RTD pages, and due to the wider default page width, and the use of the *Sphinx RTD Theme* some of the formatting may look different (for example footnotes, and some of the examples of text wrapping around images). Some of the intentional errors have been removed so the documents build remotely on <https://sphinx-rtd-theme-template.readthedocs.io/en/latest/>



## CHAPTER 4

---

### Prerequisites

---

- Sphinx - run `pip install -U Sphinx` at the command prompt (<http://www.sphinx-doc.org>)
- Sphinx RTD Theme - run `pip install sphinx_rtd_theme` at the command prompt (<https://sphinx-rtd-theme.readthedocs.io/en/stable/>)
- Pygments Code Syntax Highlighter - run `pip install Pygments` at the command prompt (<http://pygments.org/>)
- Cloud Sphinx Theme - run `pip install cloud_sptheme` at the command prompt (adds more functionality with respect to tables, see [https://cloud-sptheme.readthedocs.io/en/latest/lib/cloud\\_sptheme.ext.table\\_styling.html](https://cloud-sptheme.readthedocs.io/en/latest/lib/cloud_sptheme.ext.table_styling.html))

### 4.1 Support

If you require support or have any feature requests related to the *Sphinx RTD Theme Template* package please feel free to raise an issue on [Github](#).

For some further information and tips on setting up the theme you can refer to this blog post:- <https://engineervsheep.com/2019/parabola-8/>

### 4.2 License

This project is licensed under the MIT license.

#### 4.2.1 TEST PAGE

This test page is a test!

## 4.2.2 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.

### 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.

### 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.

### 4.2.3 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.

## 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.

## 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.

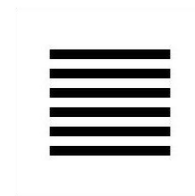


Fig. 1: This is a caption for a figure. Text should wrap around the caption.



Quisque et orci magna. Fusce rhoncus mi mi, 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.

## 4.2.4 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*

## Inline Markup

Paragraphs contain text and may contain inline markup: *emphasis*, **strong emphasis**, inline literals, standalone hyperlinks (<http://www.python.org>), external hyperlinks ([Python<sup>2</sup>](#)), internal cross-references ([example](#)), external hyperlinks with embedded URIs ([Python web site](#)), footnote references (manually numbered<sup>1</sup>, anonymous auto-numbered<sup>1</sup>, labeled auto-numbered<sup>1</sup>, or symbolic<sup>1</sup>), citation references (<sup>12</sup>), 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`.

Also with `sphinx.ext.autodoc`, which I use in the demo, I can link to `_test_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).

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

## 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:

$$\nabla^2 f = \frac{1}{r^2} \frac{\partial}{\partial r} \left( r^2 \frac{\partial f}{\partial r} \right) + \frac{1}{r^2 \sin \theta} \frac{\partial f}{\partial \theta} \left( \sin \theta \frac{\partial f}{\partial \theta} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2 f}{\partial \phi^2} \quad (4.1)$$

You can add a link to equations like the one above (4.1) by using `:eq:`.

<sup>2</sup> <http://www.python.org/>

<sup>1</sup> A footnote contains body elements, consistently indented by at least 3 spaces.

This is the footnote's second paragraph.

<sup>12</sup> This citation has some `code blocks` in it, maybe some **bold** and *italics* too. Heck, lets put a link to a meta citation<sup>13</sup> too.

<sup>13</sup> This citation will have two backlinks.

## Meta

## Blocks

### 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?
```

### 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...

## 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 brontosauruses 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)

## 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)
```

## Code Blocks

```
# parsed-literal test
curl -O http://someurl/release-1.000.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'"
    ],
    "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.'
```

## 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.<sup>11</sup>

## 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."""
```

(continues on next page)

---

<sup>11</sup> This is the citation I made, let's make this extremely long so that we can tell that it doesn't follow the normal responsive table stuff.

(continued from previous page)

```
class Foo:
```

```
    """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 :py:meth:`capitalize`.
```

```
    Here is a link to :py:meth:`__init__`.
```

```
    """
```

```
    #: Doc comment for class attribute Foo.bar.
```

```
    #: It can have multiple lines.
```

```
    bar = 1
```

```
    flox = 1.5 #: Doc comment for Foo.flox. One line only.
```

## References

## Footnotes

## Citations

Here's a reference to the above,<sup>12</sup>, and a<sup>11</sup> citation.

Here is another type of citation: *citation*

## 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*.

## 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*.

Explicit external targets are interpolated into references such as “*Python*<sup>2</sup>”.

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

## Directives

### Contents

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

### Centered text

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

This is centered text!

## Images & Figures

### Images

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



### Figures

A figure directive with center alignment

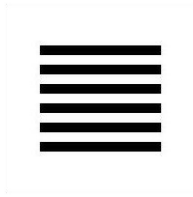


Fig. 2: A figure is an image with a caption and/or a legend:

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.

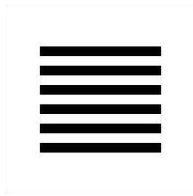


Fig. 3: This caption should be centered.

## 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.

---

## 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

Target Footnotes

Replacement Text

I recommend you try Python, *the* best language around<sup>2</sup>.

## 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.

## 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

## 4.2.5 Lists & Tables

### 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*

## Lists

### 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:
  3. Three
  4. Four
  - C. C
  - D. D
  - iii. iii
  - iv. iv
3. List items may also be auto-enumerated.

### Definition Lists

**Term** Definition

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

Definition paragraph 2.

**Term** Definition

### Option Lists

For listing command-line options:

- |                     |  |
|---------------------|--|
| <b>-a</b>           | command-line option “a”                          |
| <b>-b file</b>      | options can have arguments and long descriptions |
| <b>--long</b>       | options can be long also                         |
| <b>--input=file</b> | 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.

## Field list

- Author** David Goodger
- Address** 123 Example Street Example, EX Canada A1B 2C3
- Contact** [docutils-develop@lists.sourceforge.net](mailto: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.

## 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_
10     ↳generated from output
11     ↳`sphinx.ext.autodoc`. Which contain reST, but sphinx_
12     ↳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`

## Hlists

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

## Hlist with images

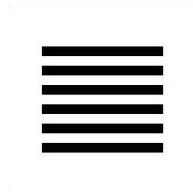


Fig. 4: This is a short caption for a figure.

•



Fig. 5: 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.

•

## 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.

- A) Using bullets and letters. (A)
- B) Using bullets and letters. (B)
- C) Using bullets and letters. (C)

## Tables

### 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.	<ul style="list-style-type: none"> <li>• Table cells</li> <li>• contain</li> <li>• body elements.</li> </ul>	
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	col-umn 2	col-umn 3	body row 1	col-umn 2	col-umn 3	body row 1	col-umn 2	col-umn 3	body row 1	col-umn 2	col-umn 3
body row 1	col-umn 2	col-umn 3	body row 1	col-umn 2	col-umn 3	body row 1	col-umn 2	col-umn 3	body row 1	col-umn 2	col-umn 3
body row 1	col-umn 2	col-umn 3	body row 1	col-umn 2	col-umn 3	body row 1	col-umn 2	col-umn 3	body row 1	col-umn 2	col-umn 3
body row 1	col-umn 2	col-umn 3	body row 1	col-umn 2	col-umn 3	body row 1	col-umn 2	col-umn 3	body row 1	col-umn 2	col-umn 3

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

 <p>Fig. 6: This is a short caption for a figure.</p>	 <p>Fig. 7: 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.</p>
---	--

4.2.6 test\_py\_module

<b>Table of Contents</b>
<ul style="list-style-type: none"><li>• <code>test_py_module</code><ul style="list-style-type: none"><li>– <i>Generated Index</i></li><li>– <i>Optional parameter args</i></li><li>– <i>Data</i></li></ul></li></ul>

Test Module for sphinx\_rtd\_theme.

```
class 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...

`__weakref__`

list of weak references to the object (if defined)

`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.

## Generated Index

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

## 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

## Data

test.[Data\\_item\\_1](#)

test.[Data\\_item\\_2](#)

test.[Data\\_item\\_3](#)

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

Some data link [Data\\_item\\_1](#).



**t**

test, [27](#)



## Symbols

`__init__()` (*test.Foo method*), 28  
`__weakref__` (*test.Foo attribute*), 28

## A

`add()` (*test.Foo method*), 28  
`another_function()` (*test.Foo method*), 28

## B

`bar` (*test.Foo attribute*), 28  
`baz` (*test.Foo attribute*), 28

## C

`capitalize()` (*test.Foo method*), 29

## D

`Data_item_1` (*in module test*), 29  
`Data_item_2` (*in module test*), 29  
`Data_item_3` (*in module test*), 29  
Documentation, 18

## F

`flox` (*test.Foo attribute*), 29  
`Foo` (*class in test*), 27

## P

`payments.dotpay.DotpayProvider`  
    (*class in test*), 29  
Python Enhancement Proposals  
    PEP 287, 13

## Q

`qux` (*test.Foo attribute*), 29

## R

Reading, 18  
RFC  
    RFC 2822, 13

## S

`spam` (*test.Foo attribute*), 29

## T

`test` (*module*), 27

## W

Writing, 18