

---

# **AcuGIS Suite**

***Release 0.4.3***

**acugis**

**Feb 21, 2020**



<b>1</b>	<b>Installation</b>	<b>1</b>
1.1	Via Git or Download . . . . .	1
<b>2</b>	<b>Configuration</b>	<b>3</b>
2.1	Theme options . . . . .	3
2.2	File-wide metadata . . . . .	5
2.3	Other configuration . . . . .	5
2.4	How the table of contents displays . . . . .	6
<b>3</b>	<b>Contributing</b>	<b>7</b>
3.1	Modifying the theme . . . . .	7
3.2	Translations . . . . .	8
3.3	Releasing the theme . . . . .	8
<b>4</b>	<b>Changelog</b>	<b>9</b>
4.1	master . . . . .	9
4.2	0.4.3 . . . . .	9
4.3	v0.4.2 . . . . .	10
4.4	v0.4.1 . . . . .	10
4.5	v0.4.0 . . . . .	10
4.6	v0.3.1 . . . . .	11
4.7	v0.3.0 . . . . .	11
4.8	v0.2.4 . . . . .	12
4.9	v0.2.3 . . . . .	12
4.10	v0.2.2 . . . . .	12
4.11	v0.2.1 . . . . .	12
4.12	v0.2.0 . . . . .	12
4.13	v0.1.10-alpha . . . . .	13
4.14	v0.1.9 . . . . .	13
4.15	v0.1.8 . . . . .	13
<b>5</b>	<b>Apache Tomcat</b>	<b>15</b>
5.1	Inline Markup . . . . .	16
5.2	Math . . . . .	17
5.3	Meta . . . . .	17
5.4	Blocks . . . . .	17
5.5	Sidebar . . . . .	19

5.6	References . . . . .	19
5.7	Directives . . . . .	20
5.8	Download Links . . . . .	23
<b>6</b>	<b>GeoServer</b>	<b>25</b>
6.1	Inline Markup . . . . .	26
6.2	Math . . . . .	27
6.3	Meta . . . . .	27
6.4	Blocks . . . . .	27
6.5	Sidebar . . . . .	29
6.6	References . . . . .	29
6.7	Directives . . . . .	30
6.8	Download Links . . . . .	33
<b>7</b>	<b>PostgreSQL</b>	<b>35</b>
7.1	Inline Markup . . . . .	36
7.2	Math . . . . .	37
7.3	Meta . . . . .	37
7.4	Blocks . . . . .	37
7.5	Sidebar . . . . .	39
7.6	References . . . . .	39
7.7	Directives . . . . .	40
7.8	Download Links . . . . .	43
<b>8</b>	<b>PostGIS</b>	<b>45</b>
8.1	Inline Markup . . . . .	46
8.2	Math . . . . .	47
8.3	Meta . . . . .	47
8.4	Blocks . . . . .	47
8.5	Sidebar . . . . .	49
8.6	References . . . . .	49
8.7	Directives . . . . .	50
8.8	Download Links . . . . .	53
<b>9</b>	<b>OpenLayers</b>	<b>55</b>
9.1	Inline Markup . . . . .	56
9.2	Math . . . . .	57
9.3	Meta . . . . .	57
9.4	Blocks . . . . .	57
9.5	Sidebar . . . . .	59
9.6	References . . . . .	59
9.7	Directives . . . . .	60
9.8	Download Links . . . . .	63
<b>10</b>	<b>Leaflet</b>	<b>65</b>
10.1	Inline Markup . . . . .	66
10.2	Math . . . . .	67
10.3	Meta . . . . .	67
10.4	Blocks . . . . .	67
10.5	Sidebar . . . . .	69
10.6	References . . . . .	69
10.7	Directives . . . . .	70
10.8	Download Links . . . . .	73
	<b>Index</b>	<b>75</b>

# CHAPTER 1

---

## Installation

---

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

```
$ pip install sphinx_rtd_theme
```

In your `conf.py` file:

```
import sphinx_rtd_theme

extensions = [
    ...
    'sphinx_rtd_theme',
]

html_theme = "sphinx_rtd_theme"
```

---

**Note:** Adding this theme as an extension is what enables localization of theme strings in your translated output. If these strings are not translated in your output, either we lack the localized strings for your locale, or you are using an old version of the theme.

---

## 1.1 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", ]
```



## 2.1 Theme options

The following options can be defined in your project's `conf.py` file, using the `html_theme_options` configuration option.

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': '',
    'style_nav_header_background': 'white',
    # Toc options
    'collapse_navigation': True,
    'sticky_navigation': True,
    'navigation_depth': 4,
    'includehidden': True,
    'titles_only': False
}
```

### 2.1.1 Table of contents options

The following options change how `toctree` directives generate documentation navigation.

**collapse\_navigation**

**Type** boolean

**Default** True

With this enabled, navigation entries are not expandable – the [ + ] icons next to each entry are removed.

**sticky\_navigation**

**Type** boolean

**Default** True

Scroll the navigation with the main page content as you scroll the page.

**navigation\_depth**

**Type** integer

**Default** 4

The maximum depth of the table of contents tree. Set this to -1 to allow unlimited depth.

**includehidden**

**Type** boolean

**Default** True

Specifies if the navigation includes hidden table(s) of contents – that is, any `toctree` directive that is marked with the `:hidden:` option.

**titles\_only**

**Type** boolean

**Default** False

When enabled, page subheadings are not included in the navigation.

---

**Note:** Setting `collapse_navigation` to `False` and using a high value for `navigation_depth` on projects with many files and a deep file structure can cause long compilation times and can result in HTML files that are significantly larger in file size.

---

## 2.1.2 Miscellaneous options

**analytics\_id**

**Type** string

If specified, Google Analytics' javascript is included in your pages. Set the value to the ID provided to you by google (like UA-XXXXXXX).

**canonical\_url**

**Type** URL

This will specify a `canonical URL` meta link element to tell search engines which URL should be ranked as the primary URL for your documentation. This is important if you have multiple URLs that your documentation is available through. The URL points to the root path of the documentation and requires a trailing slash.

**display\_version**

**Type** boolean

**Default** True

If `True`, the version number is shown at the top of the sidebar.

**logo\_only**



**Type** boolean

**Default** False

Only display the logo image, do not display the project name at the top of the sidebar

**prev\_next\_buttons\_location**

**Type** string

**Default** bottom

Location to display *Next* and *Previous* buttons. This can be either `bottom`, `top`, `both`, or `None`.

**style\_external\_links**

**Type** boolean

**Default** False

Add an icon next to external links.

**vcs\_pageview\_mode**

**Type** string

**Default** blob or view

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`.

**style\_nav\_header\_background**

**Type** string

**Default** #2980B9

Changes the background of the search area in the navigation bar. The value can be anything valid in a CSS *background* property.

## 2.2 File-wide metadata

The following options can be used as [file-wide metadata](#):

**github\_url**

Force the *Edit on GitHub* button to use the configured URL.

**bitbucket\_url**

Force the *Edit on Bitbucket* button to use the configured URL.

**gitlab\_url**

Force the *Edit on GitLab* button to use the configured URL.

## 2.3 Other configuration

### 2.3.1 Adding a logo

Using the Sphinx standard option `html_logo`, you can set an image file to be used as a logo at the top of the sidebar. The theme option `logo_only` also allows for *only* the logo to be shown at the top of the sidebar.

### 2.3.2 Adding custom CSS or Javascript

Adding custom CSS or Javascript can help you alter the look and feel of this theme without forking the theme for local use.

In order to add custom CSS or Javascript without disrupting the existing theme files, you can [add files to be included in your documentation output](#).

## 2.4 How the table of contents displays

Currently the left menu will build based upon any `toctree` directives defined in your source files. It outputs 4 levels of depth by default, to allow for quick navigation through topics. If no TOC trees are defined, Sphinx’s default behavior is to use the page headings instead.

It’s important to note that if you don’t follow the same styling for your reST headings across your documents, the TOC tree will build incorrectly, 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 not tall enough, it will revert to static positioning. To disable the sticky navigation altogether, change the `sticky_navigation` theme option.

This project follows the Read the Docs [code of conduct](#). If you are not familiar with our code of conduct policy, take a minute to read the policy before starting with your first contribution.

### 3.1 Modifying the theme

The styles for this theme use [SASS](#) and a custom CSS framework called [Wym](#). We use [Webpack](#) and [node-sass](#) to build the CSS. [Webpack](#) is used to watch for changes, rebuild the static assets, and rebuild the Sphinx demo documentation.

**Note:** The installation of Node is outside the scope of this documentation. You will need Node version 10+ in order to make changes to this theme.

#### 3.1.1 Set up your environment

1. Install [Sphinx](#) and documentation build dependencies.

```
$ pip install -e '.[dev]'
```

2. Install [Webpack](#), [node-sass](#), and theme dependencies locally.

```
$ npm install
```

#### 3.1.2 Making changes

Changes to the theme can be compiled and tested with [Webpack](#):

```
$ npm run dev
```

This script will do the following:

1. Install and update any dependencies.
2. Build the static CSS from SASS source files.
3. Build the demo documentation.
4. Watch for changes to the SASS files and documentation and rebuild everything on any detected changes.

Alternatively, if you don't need to watch the files, the release build script can be used to test built assets:

```
$ npm run build
```

## 3.2 Translations

Translations are managed using [Transifex](#). You can join any of the existing language teams or request a new language is added to the project. For more information on our translation standards, see our docs on [Internationalization](#)

Periodically, core team should update the translation files outside our normal releases. Someone from the core team, with write access to Transifex, should run the following:

```
$ python setup.py update_translations
```

This will extract new messages, upload the messages to Transifex, and will update our local translation files. Changes can be checked in to a branch and put up for review.

## 3.3 Releasing the theme

To release a new version of the theme, core team will take the following steps:

1. Bump the version in `sphinx_rtd_theme/__init__.py`, `setup.py` and `package.json`. We follow [semver](#) and [PEP440](#) (with regards to alpha release and development versions). The version increment should reflect these releases and any potentially breaking changes.
2. Update the changelog (`docs/changelog.rst`) with the version information.
3. Run `python setup.py update_translations` to compile new translation files and update Transifex.
4. Run `python setup.py build` to rebuild all the theme assets and the Python package.
5. Commit these changes.
6. Tag the release in git: `git tag $NEW_VERSION`.
7. Push the tag to GitHub: `git push --tags origin`.
8. Upload the package to PyPI:

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

### 4.1 master

#### 4.1.1 New Features

#### 4.1.2 Fixes

- Fix definition lists looking different with Sphinx 2.0+

#### 4.1.3 Other Changes

- Add the `navigation` template block around the navigation area.
- Added Spanish translation
- Added i18n support using Babel
- Moved build system from Grunt and friends to Webpack
- Remove Modernizr, but keep html5shiv (#724, #525)

### 4.2 0.4.3

**Date** Feb 12, 2019

#### 4.2.1 New Features

#### 4.2.2 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)
- Fix last\_updated notice appearing in same line as copyright notice (#704)

### 4.2.3 Other Changes

## 4.3 v0.4.2

**Date** Oct 5, 2018

### 4.3.1 New Features

### 4.3.2 Fixes

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

### 4.3.3 Other Changes

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

## 4.4 v0.4.1

**Date** July 27, 2018

### 4.4.1 New Features

### 4.4.2 Fixes

- Line height adjustments for Liberation Mono (#656)

### 4.4.3 Other Changes

- Add Sphinx as a dependency

## 4.5 v0.4.0

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

### 4.5.1 New Features

### 4.5.2 Fixes

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

### 4.5.3 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

## 4.6 v0.3.1

### 4.6.1 Fixes

- Revert part of #576 causing display issues with version selector menu
- Backwards compatibility fixes for pre-0.3.0 releases (#623)
- Fix mkdocs version selector (#622)
- Add open list spacing (#591)
- Fix table centering (#599)

## 4.7 v0.3.0

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

### 4.7.1 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

### 4.7.2 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

### 4.7.3 Other Changes

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

## 4.8 v0.2.4

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

## 4.9 v0.2.3

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

## 4.10 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.

## 4.11 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)

## 4.12 v0.2.0

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



## 4.13 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`

## 4.14 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

## 4.15 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



### Table of Contents

- *Apache Tomcat*
  - *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*
    - \* *Citations*
    - \* *Glossary*
    - \* *Targets*
  - *Directives*
    - \* *Contents*

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

## 5.1 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>1</sup>), internal cross-references ([example](#)), external hyperlinks with embedded URIs ([Python web site](#)), footnote references (manually numbered [\[1\]](#), anonymous auto-numbered [\[#\]](#), labeled auto-numbered [\[#label\]](#), or symbolic [\[\\*\]](#)), citation references (<sup>12</sup>), substitution references ([\[ComponentA/tomcat/static/tying-04-chien.jpg\]](#)), 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 to 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 <sub>subscript</sub>; a <sup>superscript</sup>; 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.

<sup>1</sup> <http://www.python.org/>

<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.

## 5.2 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:

$$\nabla^2 f = \frac{1}{r^2} \frac{\partial}{\partial r} \left( r^2 \frac{\partial f}{\partial r} \right) + \frac{1}{r^2 \sin \theta} \frac{\partial f}{\partial \theta} \left( \sin \theta \frac{\partial f}{\partial \theta} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2 f}{\partial \phi^2} \quad (5.1)$$

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

## 5.3 Meta

## 5.4 Blocks

### 5.4.1 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?
```

### 5.4.2 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...

### 5.4.3 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 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)

### 5.4.4 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)
```

### 5.4.5 Code Blocks

```
# parsed-literal test  
curl -O http://someurl/release-0.4.3.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

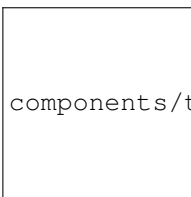
```

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.'

```

## 5.5 Sidebar

### Ch'ien / The Creative



components/tomcat/static/yi\_jing\_01\_chien.jpg

*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.

### 5.5.1 Code with Sidebar

#### A code example

With a sidebar on the right.

## 5.6 References

### 5.6.1 Citations

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

Here is another type of citation: *citation*

## 5.6.2 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.

## 5.6.3 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 “*Python*<sup>1</sup>”.

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.

## 5.7 Directives

### 5.7.1 Contents

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

### 5.7.2 Centered text

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

This is centered text!

### 5.7.3 Images & Figures

#### Images

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



#### Figures

A figure directive with center alignment



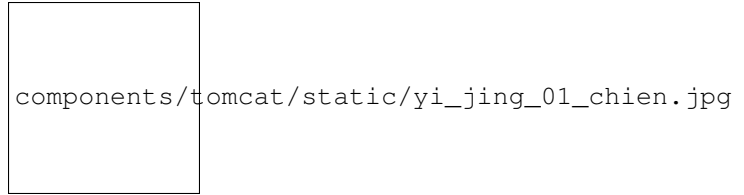


Fig. 1: A figure is an image with a caption and/or a legend:

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.

## 5.7.4 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.

---

## 5.7.5 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

### 5.7.6 Target Footnotes

### 5.7.7 Replacement Text

I recommend you try Python, *the* best language around<sup>1</sup>.

### 5.7.8 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.

## 5.8 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

- *GeoServer*
  - *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*
    - \* *Citations*
    - \* *Glossary*
    - \* *Targets*
  - *Directives*
    - \* *Contents*

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

## 6.1 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>1</sup>), internal cross-references ([example](#)), external hyperlinks with embedded URIs ([Python web site](#)), footnote references (manually numbered [\[1\]](#), anonymous auto-numbered [\[#\]](#), labeled auto-numbered [\[#label\]](#), or symbolic [\[\\*\]](#)), citation references (<sup>12</sup>), substitution references ([\[C\]](#)), 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 to 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 <sub>subscript</sub>; a <sup>superscript</sup>; 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.

<sup>1</sup> <http://www.python.org/>

<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.

## 6.2 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:

$$\nabla^2 f = \frac{1}{r^2} \frac{\partial}{\partial r} \left( r^2 \frac{\partial f}{\partial r} \right) + \frac{1}{r^2 \sin \theta} \frac{\partial f}{\partial \theta} \left( \sin \theta \frac{\partial f}{\partial \theta} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2 f}{\partial \phi^2} \quad (6.1)$$

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

## 6.3 Meta

## 6.4 Blocks

### 6.4.1 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?
```

### 6.4.2 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...

### 6.4.3 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 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)

### 6.4.4 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)
```

### 6.4.5 Code Blocks

```
# parsed-literal test  
curl -O http://someurl/release-0.4.3.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.'

```

## 6.5 Sidebar

### Ch'ien / The Creative



components/geoserver/static/yi\_jing\_01\_chien.jpg

*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.

### 6.5.1 Code with Sidebar

#### A code example

With a sidebar on the right.

## 6.6 References

### 6.6.1 Citations

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

Here is another type of citation: *citation*

## 6.6.2 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.

## 6.6.3 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 “*Python*<sup>1</sup>”.

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.

## 6.7 Directives

### 6.7.1 Contents

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

### 6.7.2 Centered text

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

This is centered text!

### 6.7.3 Images & Figures

#### Images

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



`components/geoserver/static/yi_jing_01_chien.jpg`

#### Figures

A figure directive with center alignment

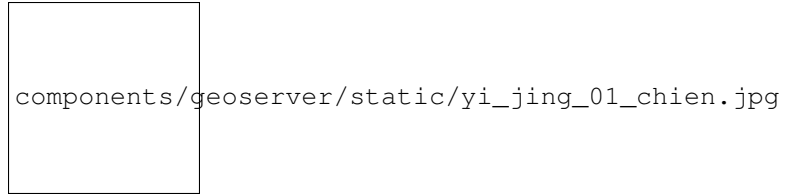


Fig. 1: A figure is an image with a caption and/or a legend:

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.

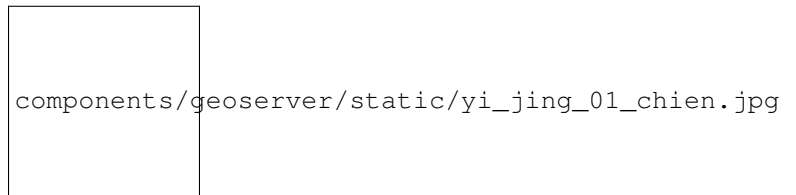


Fig. 2: This caption should be centered.

## 6.7.4 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.

---

## 6.7.5 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

### 6.7.6 Target Footnotes

### 6.7.7 Replacement Text

I recommend you try Python, *the* best language around<sup>1</sup>.

### 6.7.8 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.

## 6.8 Download Links

This 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

- *PostgreSQL*
  - *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*
    - \* *Citations*
    - \* *Glossary*
    - \* *Targets*
  - *Directives*
    - \* *Contents*

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

## 7.1 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>1</sup>), internal cross-references ([example](#)), external hyperlinks with embedded URIs ([Python web site](#)), footnote references (manually numbered [\[1\]](#), anonymous auto-numbered [\[#\]](#), labeled auto-numbered [\[#label\]](#), or symbolic [\[\\*\]](#)), citation references (<sup>12</sup>), substitution references ([components/postgresql/static/viewing\\_01\\_chaptering](#)), 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 to 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 <sub>subscript</sub>; a <sup>superscript</sup>; 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: `--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.

<sup>1</sup> <http://www.python.org/>

<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.



## 7.2 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:

$$\nabla^2 f = \frac{1}{r^2} \frac{\partial}{\partial r} \left( r^2 \frac{\partial f}{\partial r} \right) + \frac{1}{r^2 \sin \theta} \frac{\partial f}{\partial \theta} \left( \sin \theta \frac{\partial f}{\partial \theta} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2 f}{\partial \phi^2} \quad (7.1)$$

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

## 7.3 Meta

## 7.4 Blocks

### 7.4.1 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?
```

### 7.4.2 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...

### 7.4.3 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 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)

### 7.4.4 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)
```

### 7.4.5 Code Blocks

```
# parsed-literal test  
curl -O http://someurl/release-0.4.3.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

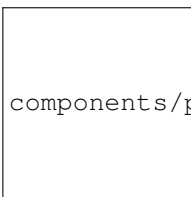
```

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.'

```

## 7.5 Sidebar

### Ch'ien / The Creative



components/postgresql/static/yi\_jing\_01\_chien.jpg

*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.

### 7.5.1 Code with Sidebar

#### A code example

With a sidebar on the right.

## 7.6 References

### 7.6.1 Citations

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

Here is another type of citation: *citation*

## 7.6.2 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.

## 7.6.3 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 “*Python*<sup>1</sup>”.

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.

## 7.7 Directives

### 7.7.1 Contents

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

### 7.7.2 Centered text

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

This is centered text!

### 7.7.3 Images & Figures

#### Images

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



#### Figures

A figure directive with center alignment

components/postgresql/static/yi\_jing\_01\_chien.jpg

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.

components/postgresql/static/yi\_jing\_01\_chien.jpg

Fig. 2: This caption should be centered.

## 7.7.4 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.

---

## 7.7.5 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

### 7.7.6 Target Footnotes

### 7.7.7 Replacement Text

I recommend you try Python, *the* best language around<sup>1</sup>.

### 7.7.8 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.

## 7.8 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

- *PostGIS*
  - *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*
    - \* *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 ([Python](#)<sup>1</sup>), internal cross-references ([example](#)), external hyperlinks with embedded URIs ([Python web site](#)), footnote references (manually numbered [\[1\]](#), anonymous auto-numbered [\[#\]](#), labeled auto-numbered [\[#label\]](#), or symbolic [\[\\*\]](#)), citation references (<sup>12</sup>), substitution references ([\[C\]](#)), 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 to 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 <sub>subscript</sub>; a <sup>superscript</sup>; 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.

<sup>1</sup> <http://www.python.org/>

<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.

## 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 (??) 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 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)

### 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-0.4.3.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

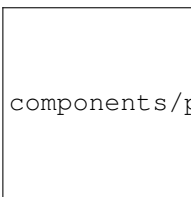
```

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.'

```

## 8.5 Sidebar

### Ch'ien / The Creative



components/postgis/static/yi\_jing\_01\_chien.jpg

*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.

## 8.6 References

### 8.6.1 Citations

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

Here is another type of citation: *citation*

## 8.6.2 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.

## 8.6.3 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 “*Python*<sup>1</sup>”.

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

components/postgis/static/yi\_jing\_01\_chien.jpg

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.

components/postgis/static/yi\_jing\_01\_chien.jpg

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 around<sup>1</sup>.

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



### Table of Contents

- *OpenLayers*
  - *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*
    - \* *Citations*
    - \* *Glossary*
    - \* *Targets*
  - *Directives*
    - \* *Contents*

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

## 9.1 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>1</sup>), internal cross-references ([example](#)), external hyperlinks with embedded URIs ([Python web site](#)), footnote references (manually numbered [\[1\]](#), anonymous auto-numbered [\[#\]](#), labeled auto-numbered [\[#label\]](#), or symbolic [\[\\*\]](#)), citation references (<sup>12</sup>), substitution references ([\[tutorials/openlayers/static/working-ol-chien-1\]](#)), 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 to 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 <sub>subscript</sub>; a <sup>superscript</sup>; 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.

<sup>1</sup> <http://www.python.org/>

<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.

## 9.2 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:

$$\nabla^2 f = \frac{1}{r^2} \frac{\partial}{\partial r} \left( r^2 \frac{\partial f}{\partial r} \right) + \frac{1}{r^2 \sin \theta} \frac{\partial f}{\partial \theta} \left( \sin \theta \frac{\partial f}{\partial \theta} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2 f}{\partial \phi^2} \quad (9.1)$$

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

## 9.3 Meta

## 9.4 Blocks

### 9.4.1 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?
```

### 9.4.2 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...

### 9.4.3 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 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)

### 9.4.4 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)
```

### 9.4.5 Code Blocks

```
# parsed-literal test  
curl -O http://someurl/release-0.4.3.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.'

```

## 9.5 Sidebar

### Ch'ien / The Creative



tutorials/openlayers/static/yi\_jing\_01\_chien.jpg

*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.

### 9.5.1 Code with Sidebar

#### A code example

With a sidebar on the right.

## 9.6 References

### 9.6.1 Citations

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

Here is another type of citation: *citation*

## 9.6.2 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.

## 9.6.3 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 “*Python*<sup>1</sup>”.

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.

## 9.7 Directives

### 9.7.1 Contents

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

### 9.7.2 Centered text

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

This is centered text!

### 9.7.3 Images & Figures

#### Images

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



`tutorials/openlayers/static/yi_jing_01_chien.jpg`

#### Figures

A figure directive with center alignment



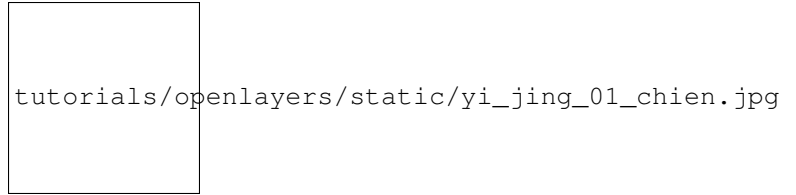


Fig. 1: A figure is an image with a caption and/or a legend:

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.

## 9.7.4 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.

---

## 9.7.5 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

### 9.7.6 Target Footnotes

### 9.7.7 Replacement Text

I recommend you try Python, *the* best language around<sup>1</sup>.

### 9.7.8 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.

## 9.8 Download Links

This 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

- *Leaflet*
  - *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*
    - \* *Citations*
    - \* *Glossary*
    - \* *Targets*
  - *Directives*
    - \* *Contents*

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

## 10.1 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>1</sup>), internal cross-references ([example](#)), external hyperlinks with embedded URIs ([Python web site](#)), footnote references (manually numbered [\[1\]](#), anonymous auto-numbered [\[#\]](#), labeled auto-numbered [\[#label\]](#), or symbolic [\[\\*\]](#)), 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*. 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 to 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 <sub>subscript</sub>; a <sup>superscript</sup>; 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.

<sup>1</sup> <http://www.python.org/>

<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.

## 10.2 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:

$$\nabla^2 f = \frac{1}{r^2} \frac{\partial}{\partial r} \left( r^2 \frac{\partial f}{\partial r} \right) + \frac{1}{r^2 \sin \theta} \frac{\partial f}{\partial \theta} \left( \sin \theta \frac{\partial f}{\partial \theta} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2 f}{\partial \phi^2} \quad (10.1)$$

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

## 10.3 Meta

## 10.4 Blocks

### 10.4.1 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?
```

### 10.4.2 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...

### 10.4.3 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 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)

### 10.4.4 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)
```

### 10.4.5 Code Blocks

```
# parsed-literal test  
curl -O http://someurl/release-0.4.3.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.'

```

## 10.5 Sidebar

### Ch'ien / The Creative



tutorials/leaflet/static/yi\_jing\_01\_chien.jpg

*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.

### 10.5.1 Code with Sidebar

#### A code example

With a sidebar on the right.

## 10.6 References

### 10.6.1 Citations

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

Here is another type of citation: *citation*

## 10.6.2 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.

## 10.6.3 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 “*Python*<sup>1</sup>”.

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.

## 10.7 Directives

### 10.7.1 Contents

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

### 10.7.2 Centered text

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

This is centered text!

### 10.7.3 Images & Figures

#### Images

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



#### Figures

A figure directive with center alignment

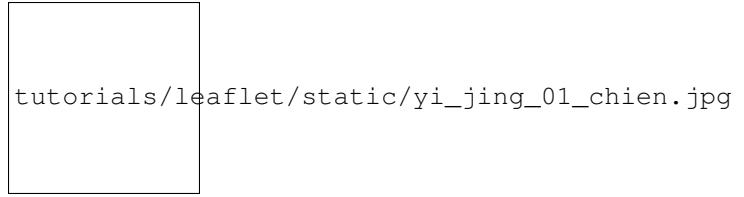


Fig. 1: A figure is an image with a caption and/or a legend:

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.

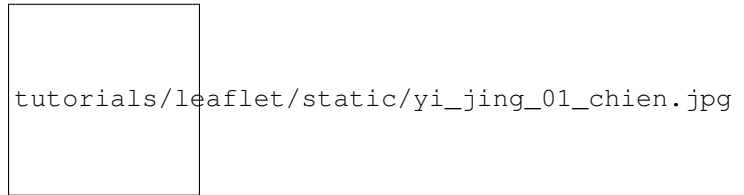


Fig. 2: This caption should be centered.

#### 10.7.4 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.

---

## 10.7.5 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

## 10.7.6 Target Footnotes

## 10.7.7 Replacement Text

I recommend you try Python, *the* best language around<sup>1</sup>.

## 10.7.8 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.

## 10.8 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



## A

analytics\_id  
configuration value, 4

## B

bitbucket\_url  
configuration value, 5

## C

canonical\_url  
configuration value, 4  
collapse\_navigation  
configuration value, 3  
configuration value  
analytics\_id, 4  
bitbucket\_url, 5  
canonical\_url, 4  
collapse\_navigation, 3  
display\_version, 4  
github\_url, 5  
gitlab\_url, 5  
includehidden, 4  
logo\_only, 4  
navigation\_depth, 4  
prev\_next\_buttons\_location, 5  
sticky\_navigation, 4  
style\_external\_links, 5  
style\_nav\_header\_background, 5  
titles\_only, 4  
vcs\_pageview\_mode, 5

## D

display\_version  
configuration value, 4  
Documentation, **20, 30, 40, 50, 60, 70**

## G

github\_url  
configuration value, 5

gitlab\_url  
configuration value, 5

## I

includehidden  
configuration value, 4

## L

logo\_only  
configuration value, 4

## N

navigation\_depth  
configuration value, 4

## P

prev\_next\_buttons\_location  
configuration value, 5  
Python Enhancement Proposals  
PEP 287, 16, 26, 36, 46, 56, 66

## R

Reading, **20, 30, 40, 50, 60, 70**  
RFC  
RFC 2822, 16, 26, 36, 46, 56, 66

## S

sticky\_navigation  
configuration value, 4  
style\_external\_links  
configuration value, 5  
style\_nav\_header\_background  
configuration value, 5

## T

titles\_only  
configuration value, 4

## V

vcs\_pageview\_mode

configuration value, 5

## **W**

Writing, [20](#), [30](#), [40](#), [50](#), [60](#), [70](#)