A ReStructuredText Primer

Author: Richard Jones
Version: 5801

Contents

The text below contains links that look like "(quickref)". These are relative links that point to the Quick reStructuredText user reference. If these links don't work, please refer to the master quick reference document.

> **Note**
>
> This document is an informal introduction to reStructuredText. The What Next? section below has links to further resources, including a formal reference.

Structure

From the outset, let me say that "Structured Text" is probably a bit of a misnomer. It's more like "Relaxed Text" that uses certain consistent patterns. These patterns are interpreted by a HTML converter to produce "Very Structured Text" that can be used by a web browser.

The most basic pattern recognised is a **paragraph** (quickref). That's a chunk of text that is separated by blank lines (one is enough). Paragraphs must have the same indentation -- that is, line up at their left edge. Paragraphs that start indented will result in indented quote paragraphs. For example:

```
This is a paragraph.  It's quite
```

```
short.

   This paragraph will result in an indented block of
   text, typically used for quoting other text.

This is another one.
```

Results in:

> This is a paragraph. It's quite short.
>
> > This paragraph will result in an indented block of text, typically used for quoting other text.
>
> This is another one.

Text styles

(quickref)

Inside paragraphs and other bodies of text, you may additionally mark text for *italics* with "`*italics*`" or **bold** with "`**bold**`". This is called "inline markup".

If you want something to appear as a fixed-space literal, use "``` ``double back-quotes`` ```". Note that no further fiddling is done inside the double back-quotes -- so asterisks "`*`" etc. are left alone.

If you find that you want to use one of the "special" characters in text, it will generally be OK -- reStructuredText is pretty smart. For example, this lone asterisk * is handled just fine, as is the asterisk in this equation: 5*6=30. If you actually want text *surrounded by asterisks* to **not** be italicised, then you need to indicate that the asterisk is not special. You do this by placing a backslash just before it, like so "`\*`" (quickref), or by enclosing it in double back-quotes (inline literals), like this:

```
``*``
```

> **Tip**
>
> Think of inline markup as a form of (parentheses) and use it the same way: immediately before and after the text being marked up. Inline markup by itself (surrounded by whitespace) or in the middle of a word won't be recognized. See the markup spec for full details.

Lists

Lists of items come in three main flavours: **enumerated**, **bulleted** and **definitions**. In all list cases, you may have as many paragraphs, sublists, etc. as you want, as long as the left-hand side of the paragraph or whatever aligns with the first line of text in the list item.

Lists must always start a new paragraph -- that is, they must appear after a blank line.

enumerated lists (numbers, letters or roman numerals; quickref)

Start a line off with a number or letter followed by a period ".", right bracket ")" or surrounded by brackets "( )" -- whatever you're comfortable with. All of the following forms are recognised:

```
1. numbers

A. upper-case letters
   and it goes over many lines

   with two paragraphs and all!

a. lower-case letters

   3. with a sub-list starting at a different number
   4. make sure the numbers are in the correct sequence though!

I. upper-case roman numerals

i. lower-case roman numerals

(1) numbers again

1) and again
```

Results in (note: the different enumerated list styles are not always supported by every web browser, so you may not get the full effect here):

1. numbers

A. upper-case letters and it goes over many lines

   with two paragraphs and all!

a. lower-case letters
   3. with a sub-list starting at a different number
   4. make sure the numbers are in the correct sequence though!

I. upper-case roman numerals

i. lower-case roman numerals

1. numbers again

1. and again

bulleted lists (quickref)

Just like enumerated lists, start the line off with a bullet point character - either "-", "+" or "*":

```
* a bullet point using "*"

  - a sub-list using "-"

    + yet another sub-list

  - another item
```

Results in:

- a bullet point using "*"
  - a sub-list using "-"
    - yet another sub-list
  - another item

definition lists (quickref)

Unlike the other two, the definition lists consist of a term, and the definition of that term. The format of a definition list is:

```
what
  Definition lists associate a term with a definition.

*how*
  The term is a one-line phrase, and the definition is one or more
  paragraphs or body elements, indented relative to the term.
  Blank lines are not allowed between term and definition.
```

Results in:

what
: Definition lists associate a term with a definition.

how
: The term is a one-line phrase, and the definition is one or more paragraphs or body elements, indented relative to the term. Blank lines are not allowed between term and definition.

Preformatting (code samples)

(quickref)

To just include a chunk of preformatted, never-to-be-fiddled-with text, finish the prior paragraph with "::". The preformatted block is finished when the text falls back to the same indentation level as a paragraph prior to the preformatted block. For example:

```
An example::

    Whitespace, newlines, blank lines, and all kinds of markup
      (like *this* or \this) is preserved by literal blocks.
  Lookie here, I've dropped an indentation level
  (but not far enough)

no more example
```

Results in:

An example:

```
    Whitespace, newlines, blank lines, and all kinds of markup
      (like *this* or \this) is preserved by literal blocks.
  Lookie here, I've dropped an indentation level
  (but not far enough)
```

no more example

Note that if a paragraph consists only of "::", then it's removed from the output:

```
::

    This is preformatted text, and the
    last "::" paragraph is removed
```

Results in:

```
This is preformatted text, and the
last "::" paragraph is removed
```

Sections

(quickref)

To break longer text up into sections, you use **section headers**. These are a single line of text (one or more words) with adornment: an underline alone, or an underline and an overline together, in dashes "-----", equals "======", tildes "~~~~~~" or any of the non-alphanumeric characters = - ` : ' " ~ ^ _ * + # < > that you feel comfortable with. An underline-only adornment is distinct from an overline-and-underline adornment using the same character. The underline/overline must be at least as long as the title text. Be consistent, since all sections marked with the same adornment style are deemed to be at the same level:

```
Chapter 1 Title
===============

Section 1.1 Title
-----------------

Subsection 1.1.1 Title
~~~~~~~~~~~~~~~~~~~~~~

Section 1.2 Title
-----------------

Chapter 2 Title
===============
```

This results in the following structure, illustrated by simplified pseudo-XML:

```
<section>
    <title>
        Chapter 1 Title
    <section>
        <title>
            Section 1.1 Title
        <section>
            <title>
                Subsection 1.1.1 Title
    <section>
        <title>
            Section 1.2 Title
<section>
```

```
        <title>
            Chapter 2 Title
```

(Pseudo-XML uses indentation for nesting and has no end-tags. It's not possible to show actual processed output, as in the other examples, because sections cannot exist inside block quotes. For a concrete example, compare the section structure of this document's source text and processed output.)

Note that section headers are available as link targets, just using their name. To link to the Lists heading, I write "`Lists_`". If the heading has a space in it like text styles, we need to quote the heading "`` `text styles`_ ``".

Document Title / Subtitle

The title of the whole document is distinct from section titles and may be formatted somewhat differently (e.g. the HTML writer by default shows it as a centered heading).

To indicate the document title in reStructuredText, use a unique adornment style at the beginning of the document. To indicate the document subtitle, use another unique adornment style immediately after the document title. For example:

```
================
 Document Title
================
----------
 Subtitle
----------

Section Title
=============

...
```

Note that "Document Title" and "Section Title" above both use equals signs, but are distict and unrelated styles. The text of overline-and-underlined titles (but not underlined-only) may be inset for aesthetics.

Images

(quickref)

To include an image in your document, you use the the `image` directive. For example:

```
.. image:: images/biohazard.png
```

results in:

The `images/biohazard.png` part indicates the filename of the image you wish to appear in the document. There's no restriction placed on the image (format, size etc). If the image is to appear in HTML and you wish to supply additional information, you may:

```
.. image:: images/biohazard.png
```

```
:height: 100
:width: 200
:scale: 50
:alt: alternate text
```

See the full image directive documentation for more info.

What Next?

This primer introduces the most common features of reStructuredText, but there are a lot more to explore. The Quick reStructuredText user reference is a good place to go next. For complete details, the reStructuredText Markup Specification is the place to go [1].

Users who have questions or need assistance with Docutils or reStructuredText should post a message to the Docutils-users mailing list.

> [1] If that relative link doesn't work, try the master document: http://docutils.sourceforge.net/docs/ref/rst/restructuredtext.html.

View document source. Generated on: 2008-12-08 07:46 UTC. Generated by Docutils from reStructuredText source.