

## Inline Markup

Inline markup allows words and phrases within text to have character styles (like italics and boldface) and functionality (like hyperlinks).	
<code>*emphasis*</code>	emphasis
<code>**strong emphasis**</code>	strong emphasis
<code>`interpreted text`</code>	The rendering and meaning of interpreted text is domain- or application-dependent.
<code>``inline literal``</code>	inline literal
<code>reference_</code>	<a href="#">phrase reference</a>
<code>`phrase reference`_</code>	<a href="#">phrase reference</a>
<code>anonymous_</code>	<a href="#">anonymous</a>
<code>_`inline internal target`</code>	inline internal target
<code>[substitution reference]</code>	The result is substituted in from the substitution definition.
<code>footnote reference [1]_</code>	footnote reference <sup>1</sup>
<code>citation reference [CIT2002]_</code>	citation reference [CIT2002]
<code>http://docutils.sf.net/</code>	<a href="http://docutils.sf.net/">http://docutils.sf.net/</a>

## Escaping with Backslashes

reStructuredText uses backslashes (") to override the special meaning given to markup characters and get the literal characters themselves. To get a literal backslash, use an escaped backslash ("). For example:

<code>*escape* ``with`` "\</code>	escape with ""
<code>\*escape* \``with`` "\</code>	*escape* ``with`` "\

## Lists

<ul style="list-style-type: none"><li>- This is item 1. A blank line before the first and last items is required.</li><li>- This is item 2</li></ul> <ul style="list-style-type: none"><li>- Item 3: blank lines between items are optional.</li><li>- Item 4: Bullets are "-", "*" or "+". Continuing text must be aligned after the bullet and whitespace.</li></ul>	<ul style="list-style-type: none"><li>• This is item 1. A blank line before the first and last items is required.</li><li>• This is item 2</li><li>• Item 3: blank lines between items are optional.</li><li>• Item 4: Bullets are "-", "*" or "+". Continuing text must be aligned after the bullet and whitespace.</li></ul>
<ol style="list-style-type: none"><li>3. This is the first item</li><li>4. This is the second item</li><li>5. Enumerators are arabic numbers, single letters, or roman numerals</li><li>6. List items should be sequentially numbered, but need not start at 1 (although not all formatters will honour the first index).</li><li>#. This item is auto-enumerated</li></ol>	<ol style="list-style-type: none"><li>3. This is the first item</li><li>4. This is the second item</li><li>5. Enumerators are arabic numbers, single letters, or roman numerals</li><li>6. List items should be sequentially numbered, but need not start at 1 (although not all formatters will honour the first index).</li><li>7. This item is auto-enumerated</li></ol>
<p><b>what</b></p> <p>Definition lists associate a term with a definition.</p>	<p><b>what</b></p> <p>Definition lists associate a term with a definition.</p>
<p><b>how</b></p> <p>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.</p>	<p><b>how</b></p> <p>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.</p>

<b>:Authors:</b> Tony J. [Tibs] Ibbs, David Goodger  (and sundry other good-natured folks)		<b>Authors:</b>	Tony J. (Tibs) Ibbs, David Goodger (and sundry other good-natured folks)
<b>:Version:</b> 1.0 of 2001/08/08 <b>:Dedication:</b> To my father.		<b>Version:</b>	1.0 of 2001/08/08
		<b>Dedication:</b>	To my father.
<code>-a</code>	command-line option "a"	<code>-a</code>	command-line option "a"
<code>-b file</code>	options can have arguments and long descriptions	<code>-b file</code>	options can have arguments and long descriptions
<code>--long</code>	options can be long also	<code>--long</code>	options can be long also
<code>--input=file</code>	long options can also have arguments	<code>--input=file</code>	long options can also have arguments
<code>/V</code>	DOS/VMS-style options too	<code>/V</code>	DOS/VMS-style options too

## Section Structure

<b>Title</b> =====	<b>Title</b>  Titles are underlined (or over- and underlined) with a nonalphanumeric character at least as long as the text.  A lone top-level section is lifted up to be the document's title
-----------------------	--

## Blocks

<p>This is a paragraph.</p>	<p>This is a paragraph.</p>
<p>Paragraphs line up at their left edges, and are normally separated by blank lines.</p>	<p>Paragraphs line up at their left edges, and are normally separated by blank lines.</p>
<p>A paragraph containing only two colons indicates the following indented or quoted text is a literal block or quoted text is a literal block.</p>	<p>A paragraph containing only two colons indicates that the following indented or quoted text is a literal block.</p>
<pre>::      Whitespace, newlines, blank lines, and all kinds of markup [like *this* or \this] is preserved here.</pre>	<p>Whitespace, newlines, blank lines, and all kinds of markup [like *this* or \this] is preserved by literal blocks.</p> <p>You can also tack the :: at the end of a paragraph:</p> <p>It's very convenient to use this form.</p>
<p>You can also tack the :: at the end of a paragraph::</p>	<p>Per-line quoting can also be used for unindented blocks:</p> <p>&gt; Useful for quotes from email and &gt; for Haskell literate programming.</p>

<p>Per-line quoting can also be used for unindented blocks::</p>	
<pre>&gt; Useful for quotes from email and &gt; for Haskell literate programming.    Line blocks are useful for addresses,   verse, and adornment-free lists.     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 spaces in place of vertical bars.</pre>	<p>Line blocks are useful for addresses, verse, and adornment-free lists.</p> <p>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 spaces in place of vertical bars.</p>
<p>Block quotes are just:</p>	<p>Block quotes are just:</p>
<pre>    Indented paragraphs,      and they may nest.</pre>	<pre>    Indented paragraphs,      and they may nest.</pre>
<p>Doctest blocks are interactive Python sessions. They begin with "<code>&gt;&gt;&gt;</code>" and end with a blank line.</p>	<p>Doctest blocks are interactive Python sessions. They begin with "<code>&gt;&gt;&gt;</code>" and end with a blank line.</p>
<pre>&gt;&gt;&gt; print "This is a doctest block." This is a doctest block.</pre>	<pre>&gt;&gt;&gt; print "This is a doctest block." This is a doctest block.</pre>

<p>A transition marker is a horizontal line of 4 or more repeated punctuation characters.</p> <p>-----</p>	<p>A transition marker is a horizontal line of 4 or more repeated punctuation characters.</p>
<p>A transition should not begin or end a section or document, nor should two transitions be immediately adjacent.</p>	<p>A transition should not begin or end a section or document, nor should two transitions be immediately adjacent.</p>

## Tables

There are two syntaxes for tables in reStructuredText. Grid tables are complete but cumbersome to create. Simple tables are easy to create but limited (no row spans, etc.).

```
+-----+-----+-----+
| Header 1 | Header 2 | Header 3 |
+-----+-----+-----+
| body row 1 | column 2 | column 3 |
+-----+-----+-----+
| body row 2 | Cells may span columns. |
+-----+-----+-----+
| body row 3 | Cells may | - Cells |
+-----+-----+-----+
| body row 4 | | - contain |
| body row 4 | | - blocks. |
+-----+-----+-----+
```

Header 1	Header 2	Header 3
body row 1	column 2	column 3
body row 2	Cells may span columns.	
body row 3	Cells may span rows.	<ul style="list-style-type: none"><li>Cells</li><li>contain</li><li>blocks.</li></ul>
body row 4		

```
====  =====  =====
      Inputs      Output
-----
      A          B      A or B
====  =====  =====
False  False      False
True   False      True
False  True        True
True   True        True
====  =====  =====
```

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

## Explicit Markup

Explicit markup blocks are used for constructs which float (footnotes), have no direct paper-document representation (hyperlink targets, comments), or require specialized processing (directives). They all begin with two periods and whitespace, the "explicit markup start".

Footnote references, like [5]\_. Note that footnotes may get rearranged, e.g., to the bottom of the "page".

Footnote references, like <sup>5</sup>. Note that footnotes may get rearranged, e.g., to the bottom of the "page".

```
.. [5] A numerical footnote. Note
   there's no colon after the `````.
```

5	A numerical footnote. Note there's no colon after the ].
---	--

Autonumbered footnotes are possible, like using [#]\_ and [#]\_.

Autonumbered footnotes are possible, like using <sup>1</sup> and <sup>2</sup>.

```
.. [#] This is the first one.
.. [#] This is the second one.
```

1	This is the first one.
---	------------------------

2	This is the second one.
---	-------------------------

They may be assigned 'autonumber labels' - for instance, [#fourth]\_ and [#third]\_.

They may be assigned 'autonumber labels' - for instance, <sup>4</sup> and <sup>3</sup>.

```
.. [#third] a.k.a. third_
```

3	a.k.a. <a href="#">third</a>
---	------------------------------

4	a.k.a. <a href="#">fourth</a>
---	-------------------------------

```
.. [#fourth] a.k.a. fourth_
```

Auto-symbol footnotes are also possible, like this: [\*]\_ and [\*]\_.

Auto-symbol footnotes are also possible, like this: <sup>\*</sup> and <sup>†</sup>.

```
.. [*] This is the first one.
.. [*] This is the second one.
```

<sup>*</sup>	This is the first one.
--------------	------------------------

<sup>†</sup>	This is the second one.
--------------	-------------------------

Citation references, like [CIT2002]\_. Note that citations may get rearranged, e.g., to the bottom of the "page".

Citation references, like [CIT2002]. Note that citations may get rearranged, e.g., to the bottom of the "page".

```
.. [CIT2002] A citation
   [as often used in journals].
```

CIT2002(1, 2)	A citation (as often used in journals).
---------------	---

Citation labels contain alphanumerics, underlines, hyphens and fullstops. Case is not significant.

Citation labels contain alphanumerics, underlines, hyphens and fullstops. Case is not significant.

Given a citation like [this]\_, one can also refer to it like this\_.

Given a citation like [this], one can also refer to it like [this](#).

```
.. [this] here.
```

<a href="#">this</a>	here.
----------------------	-------

External hyperlinks, like Python\_.

External hyperlinks, like [Python](#).

```
.. _Python: http://www.python.org/
```

External hyperlinks, like `Python`\_ <http://www.python.org/>`\_.

External hyperlinks, like [Python](#).

Internal crossreferences, like example\_.

Internal crossreferences, like [example](#).

```
.. _example:
```

This is an example crossreference target.

This is an example crossreference target.

```
Python_ is `my favourite
programming language`_.
```

[Python](#) is my favourite programming language.

```
.. _Python: http://www.python.org/
```

```
__ Python__
```

Titles are targets, too  
=====

Titles are targets, too

Implicit references, like `Titles are targets, too`\_.

Implicit references, like [Titles are targets, too](#).

Directives are a general-purpose extension mechanism, a way of adding support for new constructs without adding new syntax. For a description of all standard directives, see reStructuredText Directives (<http://is.gd/2Ecqh>).

For instance:

For instance:

```
.. image:: magnetic-balls.jpg
   :width: 48pt
```



Substitutions are like inline directives, allowing graphics and arbitrary constructs within text.

The |biohazard| symbol must be used on containers used to dispose of medical waste.

The \* symbol must be used on containers used to dispose of medical waste.

```
.. |biohazard| image:: biohazard.png
   :align: middle
   :width: 12
```

Any text which begins with an explicit markup start but doesn't use the syntax of any of the constructs above, is a comment.

```
.. This text will not be shown
   {but, for instance, in HTML might be
   rendered as an HTML comment}
```

An "empty comment" does not consume following blocks. [An empty comment is "..." with blank lines before and after.]

An "empty comment" does not consume following blocks. (An empty comment is "..." with blank lines before and after.)

So this block is not "lost", despite its indentation.

```
..
```

So this block is not "lost", despite its indentation.

## Credits

CP Font from LiquiType:

[http://www.liquitype.com/workshop/type\\_design/cp-mono](http://www.liquitype.com/workshop/type_design/cp-mono)

Magnetic Balls V2 image by fdecomite:

<http://www.flickr.com/photos/fdecomite/2926556794/>