## rst-mode: reStructuredText Mode

	Keystroke	Function	Note
Editing reStructuredText files	The reSructuredText files a	on related to reStructuredText that no re supported by the ret-mode which _, you must set the PEL <b>pel-use-rst</b>	is available in standard Emacs distribution.
Customize PEL reStructuredText Support (See also: ∑ Customize)	• <f11> <f1> SPC r • <f12> <f1></f1></f12></f1></f11>	(pel-cfg-pkg-reST &optional OTHER-WINDOW)	Customize PEL reStructuredText support.  If OTHER-WINDOW is non-nil (use C-u), display in another window.  The <f12> <f1> binding is available when point is in a buffer visiting a reStructuredText file.</f1></f12>
Activate reStructuredText mode	M-x rst-mode	(rst-mode)	Toggle the rst-mode used to edit reStructuredText markup.
Get version of rst-mode	C-h v rst-version		Shows the content of the variable rst-version.  • Works once the rst-mode is loaded only.
Display table of content	C-c C-t C-t	(rst-doc)	Display a table of contents for current buffer inside another buffer.  Displays all section titles found in the current buffer in a hierarchical list.  The resulting buffer can be navigated, and selecting a section title moves the cursor to that section.
Indent list item	<tab></tab>	(indent-for-tab-command &optional ARG)	When point is anywhere on a list item line (a line that starts with one if the supported bullet characters), this cycles the indentation through the possible indentations of the item.
(See ∑ Indentation)		(	
Comment (See ∑ Comments)	M-;	(comment-dwim ARG)	Comment line or region.  TODO: the uncommenting does not work. According to the comment-dwim description it should. Need to investigate.
Move to previous section title	• C-M-a • <f12> p</f12>	(rst-backward-section OFFSET)	Jump backward OFFSET section titles ending up at the start of the title line.  OFFSET defaults to 1 and may be negative to move backward.  An OFFSET of 0 does not move unless point is inside a title.  Go to end or beginning of buffer if no more section titles in the desired direction.
Move to next section title	• C-M-e • <f12> n</f12>	(rst-forward-section OFFSET)	Jump forward OFFSET section titles ending up at the start of the title line.  OFFSET defaults to 1 and may be negative to move backward.  An OFFSET of 0 does not move unless point is inside a title.  Go to end or beginning of buffer if no more section titles in the desired direction.
Mark complete current section	C-M-h	(rst-mark-section &optional COUNT ALLOW-EXTEND)	Select COUNT sections around point.  Mark following sections for positive COUNT or preceding sections for negative COUNT.
Section level adornment	The rst.el library provides the rst-adjust command to create section adornment of the current line. This command tries to infer the level required and unfortunately sometimes fails when market is used and not expected by its code.  PEL provides a set of very simple commands that use multiple key bindings to adorn the current line to a fixed section level: title level and up to 10 other levels, from 1 to 9 and then 0 for 10. It also provides commands to adorn a line to the same level as the previous section or a lower or higher level. And then to increase or decrease the section level of the adornment of the current line.  PEL provides 3 style of section adornments: default, Sphinx-Python and CRiSPer, which can be selected with commands.  PEL remembers the preferred style inside the customizable variable: pel-rst-adornment-style.  The rest.el provides the rst-preferred-adornment user option to select the adornment characters for the various sections. PEL code selects the value according to the adornment style you select. See section "Select Adornment Styles" below.		
Adjust section level	• C-= • C-c C-= • C-c C-a C-a	(rst-adjust PFXARG)	Auto-adjust the adornment around point.  Adjust/rotate the section adornment for the section title around point or promote/demote the adornments inside the region, depending on whether the region is active. This function is meant to be invoked possibly multiple times, and can vary its behavior with a positive PFXARG (toggle style), or with a negative PFXARG (alternate behavior).
			<ul> <li>This function is a bit of a swiss knife. It is meant to adjust the adornments of a section title in reStructuredText. It tries to deal with all the possible cases gracefully and to do "the right thing" in all cases.</li> </ul>
Adorn line at title level	• <f12> t • <f11> SPC r t</f11></f12>	(pel-rst-adorn-title)	section title in reStructuredText. It tries to deal with all the possible cases gracefully
Adorn line at title level  Adorn to specific level From level 1 to level 10		(pel-rst-adorn-1)  • (pel-rst-adorn-2)  • (pel-rst-adorn-3)  • (pel-rst-adorn-4)  • (pel-rst-adorn-5)  • (pel-rst-adorn-6)  • (pel-rst-adorn-7)  • (pel-rst-adorn-8)  • (pel-rst-adorn-9)  • (pel-rst-adorn-9)	section title in reStructuredText. It tries to deal with all the possible cases gracefully and to do "the right thing" in all cases.  Adorn current line with level-0 (title) reStructuredText section adornment.
Adorn to specific level	• <f11> SPC r t  • <f12> 1 • <f11> SPC r 1 • <f12> 2 • <f11> SPC r 2 • <f11> SPC r 3 • <f11> SPC r 3 • <f11> SPC r 4 • <f11> SPC r 4 • <f11> SPC r 5 • <f11> SPC r 5 • <f11> SPC r 5 • <f11> SPC r 6 • <f11> SPC r 6 • <f11> SPC r 7 • <f11> SPC r 8 • <f11> SPC r 8 • <f11> SPC r 9 • <f11> SPC r 9</f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f12></f11></f12></f11>	• (pel-rst-adorn-1) • (pel-rst-adorn-2) • (pel-rst-adorn-3) • (pel-rst-adorn-4) • (pel-rst-adorn-5) • (pel-rst-adorn-6) • (pel-rst-adorn-7) • (pel-rst-adorn-8) • (pel-rst-adorn-9)	section title in reStructuredText. It tries to deal with all the possible cases gracefully and to do "the right thing" in all cases.  Adorn current line with level-0 (title) reStructuredText section adornment.  If done at the top of the file, the first adorn line is placed on the first line of the file.  Adorn current line with level [1 to 10] reStructuredText section adornment.  The <f11> SPC 1 to <f11> SPC r 0 key sequences can be used inside any</f11></f11>
Adorn to specific level From level 1 to level 10  Adorn current line: same section level as previous	• <f11> SPC r t  • <f12> 1 • <f11> SPC r 1 • <f11> SPC r 1 • <f12> 2 • <f11> SPC r 2 • <f11> SPC r 3 • <f11> SPC r 3 • <f11> SPC r 4 • <f11> SPC r 5 • <f11> SPC r 5 • <f11> SPC r 5 • <f11> SPC r 6 • <f11> SPC r 6 • <f11> SPC r 7 • <f12> 7 • <f11> SPC r 8 • <f11> SPC r 8 • <f11> SPC r 9 • <f11> SPC r 0 • <f11> SPC r 0 • <f11> SPC r 0</f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f12></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f12></f11></f11></f12></f11>	• (pel-rst-adorn-1) • (pel-rst-adorn-2) • (pel-rst-adorn-3) • (pel-rst-adorn-4) • (pel-rst-adorn-6) • (pel-rst-adorn-7) • (pel-rst-adorn-7) • (pel-rst-adorn-8) • (pel-rst-adorn-9) • (pel-rst-adorn-0)	section title in reStructuredText. It tries to deal with all the possible cases gracefully and to do "the right thing" in all cases.  Adorn current line with level-0 (title) reStructuredText section adornment.  If done at the top of the file, the first adorn line is placed on the first line of the file.  Adorn current line with level [1 to 10] reStructuredText section adornment.  The <f11> SPC 1 to <f11> SPC r 0 key sequences can be used inside any buffer. The <f12> keys can only be used in inside the buffers in rst-mode.  Adorn current line with the same level as the previous section.</f12></f11></f11>
Adorn to specific level From level 1 to level 10  Adorn current line: same section level as previous section	• <f11> SPC r t  • <f12> 1 • <f11> SPC r 1 • <f11> SPC r 1 • <f12> 2 • <f11> SPC r 2 • <f11> SPC r 3 • <f12> 4 • <f11> SPC r 3 • <f12> 6 • <f11> SPC r 5 • <f11> SPC r 6 • <f11> SPC r 6 • <f11> SPC r 7 • <f12> 7 • <f11> SPC r 8 • <f11> SPC r 8 • <f11> SPC r 8 • <f12> 7 • <f11> SPC r 8 • <f11> SPC r 8 • <f11> SPC r 9 • <f11> SPC r 9 • <f11> SPC r 9 • <f11> SPC r 0 • <f11> SPC r 0 • <f11> SPC r 0</f11></f11></f11></f11></f11></f11></f11></f11></f12></f11></f11></f11></f11></f11></f11></f11></f11></f11></f12></f11></f11></f11></f11></f11></f11></f11></f12></f11></f12></f11></f11></f12></f11></f11></f12></f11>	(pel-rst-adorn-1)     (pel-rst-adorn-2)     (pel-rst-adorn-3)     (pel-rst-adorn-4)     (pel-rst-adorn-5)     (pel-rst-adorn-6)     (pel-rst-adorn-7)     (pel-rst-adorn-8)     (pel-rst-adorn-9)     (pel-rst-adorn-0)	section title in reStructuredText. It tries to deal with all the possible cases gracefully and to do "the right thing" in all cases.  Adorn current line with level-0 (title) reStructuredText section adornment.  If done at the top of the file, the first adorn line is placed on the first line of the file.  Adorn current line with level [1 to 10] reStructuredText section adornment.  The <f11> SPC 1 to <f11> SPC r 0 key sequences can be used inside any buffer. The <f12> keys can only be used in inside the buffers in rst-mode.  Adorn current line with the same level as the previous section.  If the line is already adorned, update the adornment: adjust to previous section level.  Adorn current line at a higher-level that current if already adorned.</f12></f11></f11>
Adorn to specific level From level 1 to level 10  Adorn current line: same section level as previous section Adorn to higher section level	• <f11> SPC r t  • <f12> 1 • <f12> 1 • <f11> SPC r 1 • <f12> 2 • <f11> SPC r 2 • <f11> SPC r 3 • <f12> 4 • <f11> SPC r 3 • <f12> 5 • <f11> SPC r 6 • <f12> 7 • <f11> SPC r 7 • <f12> 8 • <f11> SPC r 7 • <f12> 8 • <f11> SPC r 7 • <f12> 7 • <f12> 8 • <f11> SPC r 7 • <f12> 8 • <f11> SPC r 7 • <f12> 8 • <f11> SPC r 7 • <f12> 7 • <f12> 7 • <f12> 8 • <f11> SPC r 7 • <f12> 7 • <f12> 8 • <f11> SPC r 7 • <f12> 7 • <f12> 7 • <f11> SPC r 9 • <f12> 0 • <f11> SPC r 10 • <f12> - • <f11> SPC r 10 • <f12> - • <f11> SPC r 10</f11></f12></f11></f12></f11></f12></f11></f11></f11></f11></f11></f12></f12></f11></f12></f12></f11></f12></f12></f12></f11></f12></f11></f12></f11></f12></f12></f11></f12></f11></f12></f11></f12></f11></f11></f11></f11></f11></f12></f11></f12></f11></f11></f12></f11></f12></f12></f11>	(pel-rst-adorn-1)	section title in reStructuredText. It tries to deal with all the possible cases gracefully and to do "the right thing" in all cases.  Adorn current line with level-0 (title) reStructuredText section adornment.  • If done at the top of the file, the first adorn line is placed on the first line of the file.  Adorn current line with level [1 to 10] reStructuredText section adornment.  • The <f11> SPC 1 to <f11> SPC r 0 key sequences can be used inside any buffer. The <f12> keys can only be used in inside the buffers in rst-mode.  Adorn current line with the same level as the previous section.  • If the line is already adorned, update the adornment: adjust to previous section level.  • Adorn current line at a higher-level that current if already adorned.  • If the line is not already adorned, adorn it with a level higher than previous section.  Adorn current line at a lower-level than current if already adorned.  • If the line not already adorned, adorn it with a level lower than previous section.</f12></f11></f11>
Adorn to specific level From level 1 to level 10  Adorn current line: same section level as previous section Adorn to higher section level Adorn to lower section level Refresh current line	• <f11> SPC r t  • <f12> 1 • <f11> SPC r 1 • <f12> 2 • <f11> SPC r 2 • <f11> SPC r 3 • <f11> SPC r 3 • <f11> SPC r 4 • <f11> SPC r 4 • <f11> SPC r 5 • <f11> SPC r 5 • <f11> SPC r 6 • <f11> SPC r 7 • <f11> SPC r 8 • <f11> SPC r 9 • <f11> SPC r 9 • <f11> SPC r 0  • <f11> SPC r 10 • <f11< td=""><td>(pel-rst-adorn-1)     (pel-rst-adorn-2)     (pel-rst-adorn-3)     (pel-rst-adorn-4)     (pel-rst-adorn-5)     (pel-rst-adorn-6)     (pel-rst-adorn-7)     (pel-rst-adorn-8)     (pel-rst-adorn-9)     (pel-rst-adorn-o)  (pel-rst-adorn-o)  (pel-rst-adorn-o)  (pel-rst-adorn-o)</td><td>Adorn current line with the same level as the previous section.  If the line is already adorned, update the adornment: adjust to previous section level.  Aforn current line with level—0 (title) reStructuredText section adornment.  Adorn current line with level [1 to 10] reStructuredText section adornment.  The <f11> SPC 1 to <f11> SPC r 0 key sequences can be used inside any buffer. The <f12> keys can only be used in inside the buffers in rst-mode.  Adorn current line with the same level as the previous section.  If the line is already adorned, update the adornment: adjust to previous section level.  Adorn current line at a higher-level that current if already adorned.  If the line is not already adorned, adorn it with a level lower than previous section.  Adorn current line at a lower-level than current if already adorned.  If the line not already adorned, adorn it with a level lower than previous section.  Refresh the adornment of the current line, adjusting the underlining to the current length</f12></f11></f11></td></f11<></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f11></f12></f11></f12></f11>	(pel-rst-adorn-1)     (pel-rst-adorn-2)     (pel-rst-adorn-3)     (pel-rst-adorn-4)     (pel-rst-adorn-5)     (pel-rst-adorn-6)     (pel-rst-adorn-7)     (pel-rst-adorn-8)     (pel-rst-adorn-9)     (pel-rst-adorn-o)  (pel-rst-adorn-o)  (pel-rst-adorn-o)  (pel-rst-adorn-o)	Adorn current line with the same level as the previous section.  If the line is already adorned, update the adornment: adjust to previous section level.  Aforn current line with level—0 (title) reStructuredText section adornment.  Adorn current line with level [1 to 10] reStructuredText section adornment.  The <f11> SPC 1 to <f11> SPC r 0 key sequences can be used inside any buffer. The <f12> keys can only be used in inside the buffers in rst-mode.  Adorn current line with the same level as the previous section.  If the line is already adorned, update the adornment: adjust to previous section level.  Adorn current line at a higher-level that current if already adorned.  If the line is not already adorned, adorn it with a level lower than previous section.  Adorn current line at a lower-level than current if already adorned.  If the line not already adorned, adorn it with a level lower than previous section.  Refresh the adornment of the current line, adjusting the underlining to the current length</f12></f11></f11>

Season   S	Description	Keystroke	Function	Note	
### Casalina and Using Hyperfinitis  **Part Studies, 1998 commence tells professing for exemption of processing and programs and provided for events the position system of the processing and the processi			(pel-rst-adorn-Sphinx-Python)	This is what Sphinx supports: 6 levels:     parts,     chapters,     sections,     subsections,     subsubsections,	
## The previous common were hall it. Is stroke in the common of the create and in part of the common of the create and in part of the common of the create and interest the common of the common of the create and interest the common of the create and interest the common of the create and interest the common of the common o		1	(pel-rst-adorn-CRiSPer)		
see the neglocy (from a in prigrigated, or the word appear characters as that has for the last and crashes the last control of the bodowns of the second by explorates will be left. In exclusion of the properties of the last control of the properties of the last control of the last cont		<ul> <li>the embedded form where the URL is stored inside the text between angle brackets and</li> <li>the full named format where the link is located elsewhere in the file on its own line.</li> </ul>			
Set location of hyperfinis  - <11 > Set   Set   Fine   Set		uses the region (if one is highlighted) or the word at point otherwise as the title for the link and creates the link entry on a line identified by a dedicate bookmark: that bookmark is created by the <f12> s keystroke. That helps identify an area inside the file where the next (or several) hyperlinks will located.  In PEL, the <f12> key prefix is mode sensitive. If you want to use the same commands inside another mode, you can use the longer key chord that uses the <f11> SPC r prefix.</f11></f12></f12>			
See to hyperlink location   Control   Sec   Fig.   Sec   S	Set location of hyperlinks	• <f12> s</f12>		Set the reference bookmark for the currently edited file at point. Used to identify the location where the next invocation of M-x pel-rst-mekelink inserts fully expanded links. Ensures the bookmark is at the beginning of an empty line which is followed by	
** ** ** ** ** ** ** ** ** ** ** ** **	Go to hyperlink location	_	(pel-rst-goto-ref-bookmark)	Move point to the reference bookmark.  Useful to see where the bookmark for storing the hyperlink are currently located or add empty lines for future references.  Command pushes the mark on mark ring, type M-`to move back to previous	
* depart of the control of the con			(pel-rst-makelink &optional ARG)	Create a reStructuredText hyperlink prefix for the word at point or region's text.  If a region is active, use the text of the region to make the link, otherwise use the word at point.  If an argument (ARG, which can be a C-u) is specified, use the embedded URI format.  If no argument is specified, use the named hyperlink format:  if the region is a single word, just append an underscore to make the link  if the region is several words, surround the region with the "'" start string and the "'_" end string.  The named link is placed in the location of bookmark named "RST" if it exists and points to same file, otherwise the link is placed at the beginning of the next empty line.  The cursor is placed where the URL is to be written.  Command pushes the mark on mark ring, type M-`to move back to previous	
(See also: ∑ File Mngt)  active use C-c RET or the mouse to click on the button. If the URL is an email address a buffer to write an email to that address opens. If the URL is an email address a buffer to write an email to that address opens. If the URL is an email address a buffer to write an email to that address opens. If the URL is an email address a buffer to write an email to that address opens. If the URL is an email address a buffer to write an email to that address opens. If the URL is an email address a buffer to write an email to that address opens. If the URL is an email address a buffer to write an email to that address opens. If the URL is an email address a buffer to write an email to that address opens. If the URL is an email address a buffer to write an email to that address opens. If the URL is an email address a buffer to write an email to that address opens. If the URL is an email address a buffer to write an email to that address opens. If the URL is an email address a buffer to write an email to that address opens. If the URL is an email address a buffer to write an email to that address opens. If the URL is an email address a buffer to write an email to that address opens. If the URL is an email address a buffer to write an email to that address opens. If the URL is an email address a buffer to write an email to that address opens. If the URL is an email address a buffer to write an email to that address opens. If the URL is an email address a buffer to write an email address opens. If the URL is an email address a buffer to write an email address a buffer to write an email address opens. If the URL is an email address a buffer to write an email address.	whose name or markup link is at point  ★★  (See also ∑ Key-Chords,File-	• <f11> f .</f11>	(pel-open-at-point &optional N)	<ul> <li>If point is on a reStructuredText link in a rst-mode buffer, open the link target (that might be a local file or a URL on remote web site. In the latter case the page is opened in the systems' browser).</li> <li>If embedded space(s) are allowed in the filename, then point must be located at the first of the 2 delimiter characters. These delimiter character can be any of the following: "` '   () [] {}&lt;&gt; ' ' " "</li></ul>	
ARG)  With a prefix argument ARG, enable the mode if ARG is positive, and disable it otherwise.  Toggle goto-addr-prog- <f11> f U (goto-address-prog-mode Like 'goto-address-mode', but only for comments and strings.</f11>		active use C-c RET or the	mouse to click on the button. If the	e that turn URLs found in the current buffer into clickable buttons. Once the mode is URL is an email address a buffer to write an email to that address opens. If the URL is an	
	Toggle goto-addr-mode	<f11> f u</f11>		With a prefix argument ARG, enable the mode if ARG is positive, and disable it	
mode &optional ARG)		<f11> f U</f11>		Like 'goto-address-mode', but only for comments and strings.	

Description	Keystroke	Function	Note	
Open the URL (email or web page)	C-c RET	(goto-address-at-point &optional EVENT)	Send to the e-mail address or load the URL at point.  Send mail to address at point:  Find e-mail address around or before point. Then search backwards to beginning of line for the start of an e-mail address.  If no email address is found there, then load the URL at or before point.	
<b>Editing Content</b>	The following generic commands are useful when editing reStructuredText content.			
Fill current paragraph (See also: ∑ Filling/ Justification)	• M-q • <f11> t f p</f11>	(fill-paragraph &optional JUSTIFY REGION)	To justify as well: C-u M-q  • Notes: in refill mode this is done automatically. In auto fill mode the filling is done at the end of the line.  • d Refill also properly refill a multi-line comment.	
Align a set of lines on some text	<f11> t w a</f11>	(align-regexp BEG END REGEXP &optional GROUP SPACING REPEAT)	Align the current region using an ad-hoc rule read from the minibuffer. BEG and END mark the limits of the region. Interactively, this function prompts for the regular expression REGEXP to align with.  • First select a region, then issue the command. For example, to align assignment of variables over the equal sign use = as the regexp.  • The PEL package creates the ar alias for align-regexp, so it's also possible to invoke it with M-x ar <ret>  — Useful command to align the hyperlink references on their URL: select all hyperlink lines and then issue the command, specifying http as the regexp to line them all vertically.</ret>	
Text Emphasis	The PEL commands empl	The PEL commands emphasize the current word or marked region, then move point to the character right after the emphasized text.		
Bold	• <f12> b • <f11> SPC r b</f11></f12>	(pel-rst-bold)	Mark current word or marked region bold.  • Leave point after to the next character.	
Italic	• <f12> i • <f11> SPC r i</f11></f12>	(pel-rst-italic)	Mark current word or marked region italic.  • Leave point after to the next character.	
Literal	• <f12> 1 • <f11> SPC r 1</f11></f12>	(pel-rst-literal)	Mark current word or marked region with the literal markup.  • Leave point after to the next character.	
Interpreted	• <f12> ` • <f11> SPC r `</f11></f12>	(pel-rst-interpreted)	Mark current word or marked region with the interpreted markup.  • Leave point after to the next character.	

## rst-mode - References

Description & URL	Notes
Emacs Support for reStructuredText	
How to get the table of content with section numbers?	
reStructuredText	Main page for all reStructuredText documents.
reStructuredText markup Specifications	Formal markup specifications.
Sphinx Python Documentation Generator	
Sphinx - Documentation Contents	
Sphinx — Documentation — Sections	