rst-mode: reStructuredText Mode

Falities	<u>Keystroke</u>	Function	<u>Note</u>	
Editing reStructuredText files	There's more information related to reStructuredText that need to be documented. The reSructuredText files are supported by the ret-mode which is available in standard Emacs distribution. To activate it under PEL, you must set the PEL pel-use-rst-mode customization variable to t. Pel-rst-tab-width: The width of a tab used for reStructuredText files. Defaults to 2. This concept differs from indentation: you can have an indentation of 3 and tab width of 8: M-i will move point to columns that are multiple of 8 <tab> will indent to a column that is a multiple of 3. PEL stores this value inside the tab-width user option variable for rst-mode buffers. See Indentation.</tab>			
Customize PEL reStructuredText Support See also:	• <f11> <f2> SPC r • <f12> <f2></f2></f12></f2></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> <f2> binding is available when point is in a buffer visiting a reStructuredText file.</f2></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	
(See <u>∑ Indentation</u>		,	item.	
Comment See also: <u>See Comments</u>	M-;	(comment-dwim ARG)	Comment line or region. TODO: the uncommenting does not work. According to the comment-dwim	
		(b b	description it should. Need to investigate.	
Move to previous section title	• C-M-a • <f12> p • <f12> <up></up></f12></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> <down></down></f12></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.			
	The rest.el provides the	rst-preferred-adornment user of	ole variable: pel-rst-adornment-style . option to select the adornment characters for the various sections. PEL code selects the	
Adjust section level	The rest.el provides the	rst-preferred-adornment user of	ple variable: pel-rst-adornment-style. pption to select the adornment characters for the various sections. PEL code selects the ion "Select Adornment Styles" below. 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). • This function is a bit of a swiss knife. It is meant to adjust the adornments of a section	
Adjust section level Adorn line at title level	The rest.el provides the value according to the ador • C-= • C-c C-=	rst-preferred-adornment user onment style you select. See sect	ple variable: pel-rst-adornment-style. pption to select the adornment characters for the various sections. PEL code selects the ion "Select Adornment Styles" below. 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). 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 rest.el provides the value according to the ador C-= C-c C-= C-c C-a C-a	rst-preferred-adornment user on ment style you select. See sect (rst-adjust PFXARG)	ple variable: pel-rst-adornment-style. pption to select the adornment characters for the various sections. PEL code selects the ion "Select Adornment Styles" below. 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). • 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. Adorn current line with level-0 (title) reStructuredText section adornment.	
Adorn line at title level Adorn to specific level	• <f12> t • <f12> t • <f12> t • <f11> SPC r 1 • <f12> 2 • <f11> SPC r 2 • <f11> SPC r 3 • <f11> SPC r 3 • <f11> SPC r 4 • <f12> 5 • <f11> SPC r 6 • <f11> SPC r 5 • <f11> SPC r 5 • <f11> SPC r 8 • <f11> SPC r 9 • <f11> SPC r 9</f11></f11></f11></f11></f11></f11></f12></f11></f11></f11></f11></f12></f11></f12></f12></f12>	rst-preferred-adornment user of nament style you select. See sect (rst-adjust PFXARG) (pel-rst-adorn-title) (pel-rst-adorn-1) (pel-rst-adorn-2) (pel-rst-adorn-3) (pel-rst-adorn-5) (pel-rst-adorn-6) (pel-rst-adorn-7) (pel-rst-adorn-8) (pel-rst-adorn-9)	ple variable: pel-rst-adornment-style. pption to select the adornment characters for the various sections. PEL code selects the ion "Select Adornment Styles" below. 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). • 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. 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.	
Adorn line at title level Adorn to specific level From level 1 to level 10 Adorn current line: same section level as previous	**The rest.el provides the value according to the ador* **C-= **C-c C-= **C-c C-= **C-c C-a C-a	rst-preferred-adornment user on ment style you select. See sect (rst-adjust PFXARG) (pel-rst-adorn-title) (pel-rst-adorn-1) (pel-rst-adorn-2) (pel-rst-adorn-3) (pel-rst-adorn-5) (pel-rst-adorn-6) (pel-rst-adorn-7) (pel-rst-adorn-8) (pel-rst-adorn-9) (pel-rst-adorn-0)	potion to select the adornment characters for the various sections. PEL code selects the ion "Select Adornment Styles" below. 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). • 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. 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.</f12></f11></f11>	
Adorn line at title level Adorn to specific level From level 1 to level 10 Adorn current line: same section level as previous section	**The rest.el provides the value according to the ador* **C-= **C-c C-= **C-c C-= **C-c C-a C-a **Cf12> t **Cf11> SPC r t **Cf12> 2 **Cf11> SPC r 2 **Cf12> 3 **Cf11> SPC r 3 **Cf12> 4 **Cf12> 5 **Cf11> SPC r 5 **Cf12> 6 **Cf12> 6 **Cf12> 7 **Cf12> 7 **Cf12> 8 **Cf12> 7 **Cf12> 8 **Cf12> 9 **Cf12> 9 **Cf12> 0 **Cf12> 1 **Cf12	rst-preferred-adornment user on ment style you select. See sect (rst-adjust PFXARG) (pel-rst-adorn-title) (pel-rst-adorn-1) (pel-rst-adorn-2) (pel-rst-adorn-3) (pel-rst-adorn-5) (pel-rst-adorn-6) (pel-rst-adorn-7) (pel-rst-adorn-8) (pel-rst-adorn-9) (pel-rst-adorn-0)	potion to select the adornment characters for the various sections. PEL code selects the ion "Select Adornment Styles" below. 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). • 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. 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 line at title level Adorn to specific level From level 1 to level 10 Adorn current line: same section level as previous section Adorn to higher section level	**The rest.el provides the value according to the ador* **C-= **C-C C-= **C-C C-= **C-C C-a C-a	rst-preferred-adornment user on ment style you select. See sect (rst-adjust PFXARG) (pel-rst-adorn-title) (pel-rst-adorn-1) (pel-rst-adorn-2) (pel-rst-adorn-3) (pel-rst-adorn-5) (pel-rst-adorn-7) (pel-rst-adorn-7) (pel-rst-adorn-9) (pel-rst-adorn-0) (pel-rst-adorn-o) (pel-rst-adorn-o) (pel-rst-adorn-o)	ption to select the adornment characters for the various sections. PEL code selects the ion "Select Adornment Styles" below. 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). 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. 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.</f12></f11></f11>	

<u>Description</u>	<u>Keystroke</u>	Function	<u>Note</u>
Select default adornment style	• <f12> A d • <f11> SPC r A d</f11></f12>	(pel-rst-adorn-default)	Set the default section adornment style. This is Emacs rst-mode default: a title with 7 levels.
Select Sphinx-Python adornment style	• <f12> A S • <f11> SPC r A S</f11></f12>	(pel-rst-adorn-Sphinx- Python)	Set the Sphinx-Python section adornment style. This is what Sphinx supports: 6 levels: parts, chapters, sections, subsections, subsubsections, paragraphs.
Select CRiSPer adornment style	• <f12> A C • <f11> SPC r A C</f11></f12>	(pel-rst-adorn-CRiSPer)	Set the CRiSPer section adornment style. A title level with another 12 levels. Use <f12> + to create those levels.</f12>
Creating and Using Hyperlinks	the embedded form where the full named format when When editing a buffer using t uses the region (if one is high	lighted) or the word at point oth	kt between angle brackets and
	uses the <f11> SPC r pr</f11>	efix.	to use the same commands inside another mode, you can use the longer key chord that st-mode customization variable to t.
Set location of hyperlinks	• <f12> s • <f11> SPC r s</f11></f12>	(pel-rst-set-ref-bookmark)	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 another empty line, by inserting 2 lines and placing the point at the beginning of the first of the 2 lines.
Go to hyperlink location	• <f12> g • <f11> SPC r g</f11></f12>	(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 location.
Add an hyperlink for text at point	• <f12> . • <f11> SPC r .</f11></f12>	(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 location.
Open file or web-page whose name or markup link is at point ** See also: Key-Chords File mngt	• C-^ • <f11> f . • 6y</f11>	(pel-open-at-point &optional N)	Open the file, library or the URL, named at point, with potential line & column #s. 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). 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: "^' ()[]{}<>'"" 「J () ⟨ ⟩ ⟨ ⟩ () [] () ⟨ ⟩ ⟨ ⟩ () [] () ⟨ ⟩ ⟨ ⟩ () [] () ⟨ ⟩ ⟨ ⟩ () ⟨ ⟩ () () ⟨ ⟩ ⟨ ⟩ () ⟨ ⟩ ⟨ ⟩ () ⟨ ⟩ ⟨ ⟩ () ⟨ ⟩ ⟨ ⟩ () ⟨ ⟩ ⟨ ⟩ ⟨ ⟩ () ⟨ ⟩ ⟨ ⟩ ⟨ ⟩ ⟨ ⟩ ⟨ ⟩ ⟨ ⟩ ⟨ ⟩ ⟨ ⟩ ⟨ ⟩ ⟨
Activating URLs See also: File mngt	active use C-c RET or the mo an web or FTP address the s	ouse to click on the button. If the ystem browser is invoked to ope	
Toggle goto-addr-mode	<f11> f u</f11>	(goto-address-mode &optional ARG)	Minor mode to buttonize URLs and e-mail addresses in the current buffer. With a prefix argument ARG, enable the mode if ARG is positive, and disable it otherwise.
Toggle goto-addr-prog- mode	<f11> f U</f11>	(goto-address-prog-mode &optional ARG)	Like 'goto-address-mode', but only for comments and strings.

<u>Description</u>	<u>Keystroke</u>	Function	<u>Note</u>		
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.		
Editing Content	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. 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 emphas	ize the current word or marked r	egion, 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.		
Tempo skeletons for reStructuredText See also: Inserting Text	PEL provides support for flexible text template insertion through the Emacs built-in tempo skeleton mechanism. • PEL creates key bindings to invoke the skeletons in the supported major modes, using the same key prefix sequence for each mode: f12> , with the same key bindings for equivalent concepts (such as file header block) as much as possible. See also: <a "case",="" <f12="" and="" completely="" etc)="" hit="" href="mailto:single-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-state-sta</td></tr><tr><td>Insert a file header</td><td><f12> <f12> h</td><td>(pel-rst-large-header)</td><td>Insert a large header includes all normal header fields plus separators. • Prompts for title and insert title, automatically updated timestamp, attributes for home page and license, markup for table of contents using the tempo skeleton mechanism. • Automatically activates the PEL tempo skeleton mode so you can move to the target points where extra text must be entered to complete the template.</td></tr><tr><td>Toggle pel-tempo-mode</td><td><f12> <f12> SPC</td><td>(pel-tempo-mode &optional ARG)</td><td>Toggle PEL tempo mode on/off. PEL tempo mode activates C-c . and C-c , as well as to C-c C and C-c C-, key bindings to navigate across tempo mark hot-spots. When pel-tempo-mode is active the pel-tempo-mode lighter (‡) is shown on the status bar. The second set are only available when Emacs runs in graphics mode. When a skeleton is inserted via the execution of one of the pel-rst commands, the pel-tempo-mode is automatically activated.</td></tr><tr><td>Jump to next tempo mark</td><td>• C-c M-f
• C-c .
• C-c C</td><td>(tempo-forward-mark)</td><td>Jump to the next mark in 'tempo-back-mark-list': the location where code must be updated inside the inserted skeleton. • These key key bindings are only available when pel-tempo-mode is active.</td></tr><tr><td>Jump to previous tempo mark</td><td>• C-c M-b
• C-c ,
• C-c C-,</td><td>(tempo-backward-mark)</td><td>Jump to the previous mark in 'tempo-back-mark-list': the location where code must be updated inside the inserted skeleton. • These key binding are only available when pel-tempo-mode is active.</td></tr><tr><td>Tempo Template Tag
Insertion</td><td><f12> <f12> <f12></td><td>(tempo-complete-tag
&optional SILENT)</td><td>Look for a tag and expand it. Instead of using the <f12> <f12> key bindings above, you can type the template name (shown in the title column like " if",="" or="" partially="" then=""> <f12> <f12> < £12>. A completion buffer opens up if the template name is incomplete (or empty in which case the buffer lists all available template names). Select the template name and hit RET. Emacs expands the template. • All the tags in the tag lists in 'tempo-local-tags' (this includes 'tempo-tags') are searched for a match for the text before the point. The way the string to match for is determined can be altered with the variable 'tempo-match-finder'. If 'tempo-match-finder' returns nil, then the results are the same as no match at all. • If a single match is found, the corresponding template is expanded in place of the matching string. • If a partial completion or no match at all is found, and SILENT is non-nil, the function will give a signal. • If a partial completion is found and 'tempo-show-completion-buffer' is non-nil, a buffer containing possible completions is displayed. ■ Since only one template is available in rst-mode, the usefulness of this command is limited for reStructuredText.</f12></f12>				

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	