rst-mode: reStructuredText Mode

Description	Keystroke (bold:= global map)	Function	Note
Editing reStructuredText files	🚧 There's more information related to reStructuredText that need to be documented. The reStructuredText 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.		
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.
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). • 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.
Comment	`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`	**(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`	**(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.
Writing Hyperlinks	The following 3 PEL commands help write hyperlink of various forms: • the embedded form where the URL is stored inside the text between angle brackets and • the full named format where the link is located elsewhere in the file on its own line. The `<f11> m .` keystroke runs the command that creates a hyperlink, the long named format by default: it 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 dedicated bookmark: that bookmark is created by the `<f11> m s` keystroke. That helps identify an area inside the file where the next (or several) hyperlinks will be located.		
Set location of hyperlinks	`<f11> m s`	**(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	`<f11> m g`	**(pel-rst-goto-ref-bookmark)**	Move point to the reference bookmark. • Useful to see where the bookmark for storing the hyperlink are currently located.
Add an hyperlink for text at point	`<f11> m .`	**(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. • You can return to the location of the link by typing `M-\``.
rst-preferred-adornment			Identifies the list of section adornments, how to identify each section level. • Set this variable to **nil** if you want to limit the section adornment to what is currently used inside the file.
Editing Content	The following generic commands are useful when editing reStructuredText content.		
Fill current paragraph	• `M-q` • `<f11> t f p`	**(fill-paragraph** &optional JUSTIFY REGION**)**	To justify as well: **C-u M-q** Note: in refill mode this is done automatically. In auto fill mode the filling is done at the end of the line.
Align a set of lines on some text	`<f11> t w a`	**(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.

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	

2

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	