## rst-mode: reStructuredText Mode

1st-illoue. restructured text wode				
<u>Description</u>	<u>Keystroke</u>	Function	<u>Note</u>	
reStructuredText  O Help & customize  rst-mode, syntax-control  text-filling, text emphasis  itemize list of lines  Format reStucturedText table  View/Navigate Table of Content  Insert file header  Adorn Section Level  Creating/Using hyperlinks  Using goto-url-mode  Copy URL target in file & visit it  Open File at point  Supports reStructured text hyperlinks	This page describes Emacs support for reStructuredText (abbreviated sometimes as 'rst' and sometimes as 'reSt').  • The reSructuredText files are supported by Emacs rst-mode from rst.el which is available in standard Emacs distribution.  • Supported file extensions: .rst, .rest, .stxt and .rst.txt. The .rst.txt extension allows rendering by tools supporting .txt files.  • 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 <a href="tab"><a href="t&lt;/th&gt;&lt;/tr&gt;&lt;tr&gt;&lt;th&gt;compile/render file     More Info on reSTructuredText:     Last updated on:&lt;/th&gt;&lt;td colspan=2&gt;Emacs Support for reSTructuredText     Basic Intro to rst&lt;/td&gt;&lt;td&gt;&lt;ul&gt; &lt;li&gt;reStructuredText Directives&lt;/li&gt; &lt;li&gt;Quick reference to rst&lt;/li&gt; &lt;li&gt;rst-cheatsheet (pdf)&lt;/li&gt; &lt;/ul&gt;&lt;/td&gt;&lt;td&gt;&lt;ul&gt;     &lt;li&gt;Docutils Front-end tools&lt;/li&gt;     &lt;li&gt;Sphinx @ Wikipedia, Sphinx home&lt;/li&gt;     &lt;li&gt;Sphinx &amp; rst syntax guide&lt;/li&gt; &lt;/ul&gt;&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;th&gt;Open this PDF file. See also: &lt;u&gt;∑ Help/Info&lt;/u&gt;&lt;/th&gt;&lt;td&gt;&lt;f11&gt; SPC M-r &lt;f1&gt; &lt;f12&gt; &lt;f1&gt;&lt;/td&gt;&lt;td&gt;(pel-help-pdf &amp;optional OPEN-&lt;br&gt;WEB-PAGE)&lt;/td&gt;&lt;td&gt;Open the &lt;u&gt;M reStructuredText&lt;/u&gt; local PDF. If the p&lt;br&gt;then it opens the remote GitHub hosted raw PDI&lt;br&gt;user-option is set it's the other way around.&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;th&gt;© Customize PEL reStructuredText support&lt;/th&gt;&lt;td&gt;&lt;f11&gt; SPC M-r &lt;f2&gt; &lt;f12&gt; &lt;f2&gt;&lt;/td&gt;&lt;td&gt;(pel-customize-pel &amp;optional OTHER-WINDOW)&lt;/td&gt;&lt;td&gt;Customize PEL reStructuredText support: pel-pl- • If OTHER-WINDOW is non-nil (use &lt;b&gt;C-u&lt;/b&gt;), disp&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;th&gt;© Customize Emacs reStructuredText support&lt;/th&gt;&lt;td&gt;&lt;f11&gt; SPC M-r &lt;f3&gt; &lt;f12&gt; &lt;f3&gt;&lt;/td&gt;&lt;td&gt;(pel-customize-library &amp;optional OTHER-WINDOW)&lt;/td&gt;&lt;td&gt;Customize Emacs reStructuredText support: rst • If OTHER-WINDOW is non-nil (use &lt;b&gt;C-u&lt;/b&gt;), disp&lt;/td&gt;&lt;td&gt;lay in another window.&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;th&gt;&lt;u&gt;rst-mode&lt;/u&gt;&lt;/th&gt;&lt;td&gt;Emacs provides the rst-mode The following file extensions&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;td&gt;.rst, .rest. PEL adds the .stxt extension.&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;th&gt;Activate reStructuredText mode&lt;/th&gt;&lt;td&gt;M-x rst-mode&lt;/td&gt;&lt;td&gt;(rst-mode)&lt;/td&gt;&lt;td&gt;Toggle the rst-mode used to edit reStructuredTe  • Automatically invoked when visiting .rst, .rest&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;th&gt;Get version of rst-mode&lt;/th&gt;&lt;td&gt;C-h v rst-version&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;td&gt;Shows the content of the variable rst-version. C&lt;/td&gt;&lt;td&gt;only works once the rst-mode is loaded.&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;th&gt;Syntax Control  • superword-mode&lt;/th&gt;&lt;td colspan=3&gt;reStructuredText files often benefit from the &lt;a href=" superword-mode"="">superword-mode</a> and treating underscores as part of words. This helps searching for symbols and opening file that have names that include underscore characters.  To force activation of various modes in reStructuredText: first open PEL customization buffer for reStructuredText with <f12> <f2>, then  To force <a href="superword-mode">superword-mode</a> for reStructuredText: add the <a href="superword-mode">superword-mode</a> to the <a href="pel-rst-activates-minor-modes">pel-rst-activates-minor-modes</a> user-option.  To force treating underscore as symbol during superword-mode: add <a href="pel-rst-set-underscore-syntax">pel-rst-set-underscore-syntax</a> to the <a href="after-superword-mode">after-superword-mode</a> in the <a href="pel-rst-set-underscore-syntax">pel-rst-set-underscore-syntax</a> to the <a href="pel-rst-set-underscore-syntax">after-superword-mode</a> in the <a href="pel-rst-set-underscore-syntax">after-superword-mode</a> in the <a href<="" td=""></a></f2></f12></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a>			
Control syntax of the underscore character	<f12> _</f12>	(pel-rst-set-underscore-syntax &optional ACTION)	Control syntax of underscore to punctuation or sactive.	symbol when the superword-mode is
Requires superword-mode on.     See:      Text Modes	<ul> <li>If superword-mode is off the function issues a user error. Otherwise the function operates according to the value of the optional argument:</li> <li>If the argument is not specified or nil, the function toggles the syntax of the underscore character between punctuation (the default) and symbol.</li> <li>If the argument is positive, it sets the syntax of underscore to symbol.</li> <li>If the argument is 0 or negative it sets the syntax of the underscore back to punctuation.</li> </ul>			
<b>Editing Content</b>	The following generic commands are useful when editing reStructuredText content.			
• Text filling • Extilling/Justification	Although text filling will be handled for the generated rendering, you may decide to fill the reStructuredText file itself, after all you're using a markup that's made to allow reading the original text. You can turn the auto fill mode on and identify the fill column.  Force the auto-fill-mode when a reStructuredText file is visited by adding the auto-fill-mode to the pel-rst-activates-minor-modes user-option.			
Toggle auto-fill mode	• <f11> t f a • <f11> RET</f11></f11>	(auto-fill-mode &optional ARG)	Toggle automatic line breaking (Auto Fill mode).  With a prefix argument, enable Auto Fill mode disable it otherwise.  When Auto Fill mode is enabled, inserting a speculumn' automatically breaks the line at a previous column.	pace at a column beyond 'current-fill-
Set Fill Column	• C-x f • <f11> t f c</f11>	(set-fill-column ARG)	When no prefix value: prompts for column unles  If with <b>C-u</b> prefix: use current column.  If with prefix value: use that value.	s a prefix argument was used.
Fill current paragraph	• M-q • <f11> t f p</f11>	(fill-paragraph & optional JUSTIFY REGION)	To justify as well: <b>C-u M-q</b> • In refill mode this is done automatically. In aut the line.	o fill mode the filling is done at the end of
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 rea mark the limits of the region. Interactively, this for expression REGEXP to align with.	
	<ul> <li>First select a region, then issue the command. For example, to align assignment of variables over the equal sign use = as the regexp.</li> <li>The PEL package creates the ar alias for align-regexp, so it's also possible to invoke it with M-x ar RET</li> <li>Use it to align hyperlink references URL: select all hyperlink lines and then issue the command, specifying http as the regexp to line them vertically.</li> </ul>			
Text Emphasis	The PEL commands emphasi	ize the current word or marked region	n, then move point to the character right after the	emphasized text or inside if empty.
Bold	<f12> b <f11> SPC M-r b</f11></f12>	(pel-rst-bold)	Mark current word or marked region bold. If poi  Leave point after to the next character.  Inserts required escaped spaces when the em	
Italic	<f12> i <f11> SPC M-r i</f11></f12>	(pel-rst-italic)	Mark current word or marked region italic. If poir  Leave point after to the next character.  Inserts required escaped spaces when the em	nt after word, use previous word.
Literal	<f12> 1 <f11> SPC M-r 1</f11></f12>	(pel-rst-literal)	Mark current word or marked region with the lite word.  • Leave point after to the next character.	
Interpreted	<f12> ` <f11> SPC M-r `</f11></f12>	(pel-rst-interpreted)	<ul> <li>Inserts required escaped spaces when the em</li> <li>Mark current word or marked region with the interpretable to the next character.</li> <li>Inserts required escaped spaces when the em</li> </ul>	erpreted markup.
Indent list item See also: ∑ Indentation	<tab></tab>	(indent-for-tab-command &optional ARG)	Inserts required escaped spaces when the em  With point anywhere on a list item line (a line that characters), this cycles the indentation through the second control of th	t starts with one if the supported bullet
Comment	M-;	(comment-dwim ARG)	Comment line or region.	

<u>Description</u>	<u>Keystroke</u>	Function	Note	
Itemize all previous lines same indention level	• <f12> M • M-<f12> M</f12></f12>	(pel-itemize-lines &optional ITEM- PREFIX-STRING)	<ul> <li>Prepend each of the previous lines with a ITEM-PREFIX-STRING ("- " by default).</li> <li>Indents all lines above current line that are at the same indentation level.</li> <li>Use it to put a "- " prefix on each line instead of typing manually. Put point at empty line after list. Type the command to itemize all lines above. No need to mark.</li> </ul>	
Duplicate current table underlining  This is not a perfect helper for creating restructured Toxt table but	• <f12> M-t • M-<f12> M-t</f12></f12>	(pel-rst-table-dup-separator- lines &optional UPDATE)	Duplicate the current table underlining separator, adding it to the top and the bottom of the table. This command helps you creating a <u>reStructuredText simple table layout</u> , assuming you have only 1 title row and point is at the written separator line.  • With the <b>C-u</b> option updates the top and bottom line to the line at point.	
it helps creating simple ones.	Assuming you have the following text and point is on the second line (in blue)			
For example, to create the top and bottom line from the line under the	After executing the	3 3	ocdef. ghijkl. 12345	
title:	command, the top and bottom lines are inserted. The point does not move.		2345	
File's Table of Content	<ul> <li>Use the <u>contents markup directive</u> to generate a table of contents for your reStructuredText file based on its sections.</li> <li>Insert the table of content text inside the file with the <u>rst-toc-insert</u> command (although you may want to use the <u>contents markup directive</u> instead).</li> <li>Generate a table of content in a buffer to view the file sections and navigate inside them:</li> <li>with C-c C-t C-t to invoke the <u>rst-doc</u> command: it opens a *table of Content* buffer, moves point inside it, move to the section title, hit RET to select that section inside the original reStructuredText buffer.</li> </ul>			
See also: <u>∑ Speedbar</u>		ppen a buffer that lists the sections. S		
Insert a table content at point  Alternative: use the	C-c C-t TAB	(rst-toc-insert &optional MAX- LEVEL)	Insert the table of contents of the current section at the current column.  • By default the top level is ignored if there is only one.	
contents markup directive			-max-level'. Text in the line beyond column is deleted.	
Display table of content	C-c C-t C-t	(rst-doc)	Display a table of contents for current buffer inside the *Table of Contents* buffer.  • Displays all section titles found in the current buffer in a hierarchical list.	
Navigate to specific section			y navigating to it, then hit <b>RET</b> on that section to that section in the file.	
Moving across sections		g commands to move to the next or		
Move to previous section title	• C-M-a • <f12> p • <f12> <up></up></f12></f12>	(rst-backward-section OFFSET)	<ul> <li>Jump backward OFFSET section titles ending up at the start of the title line.</li> <li>OFFSET defaults to 1 and may be negative to move backward.</li> <li>An OFFSET of 0 does not move unless point is inside a title.</li> <li>Go to end or beginning of buffer if no more section titles in the desired direction.</li> </ul>	
	• <f11> SPC M-r p • <f11> SPC M-r <up></up></f11></f11>			
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 may a unless point is inside a title.	
	• <f12> <down>  • <f11> SPC M-r n • <f11> SPC M-r <down></down></f11></f11></down></f12>		<ul> <li>An OFFSET of 0 does not move unless point is inside a title.</li> <li>Go to end or beginning of buffer if no more section titles in the desired</li> <li>direction.</li> </ul>	
Mark complete current section	С-М-Һ	(rst-mark-section &optional COUNT ALLOW-EXTEND)	Select COUNT sections around point.  • Mark following sections for positive COUNT or preceding sections for negative COUNT.	
Insert elements based on Tempo skeletons for reStructuredText	PEL provides support for flexible text template insertion through the Emacs built-in tempo skeleton mechanism. See also: Inserting Text  • PEL creates key bindings to invoke the skeletons in the supported major modes, using the same key prefix sequence for each mode: <f12> <f12>, with the same key bindings for equivalent concepts (such as file header block) as much as possible.  See also: Inserting Text for more info and information about tempo skeleton and yasnippet template-based text insertion).</f12></f12>			
∑ Customize PEL Text Insertions	<f6> <f2></f2></f6>	(pel-customize-pel &optional	Open the customization that control some of the features of the reStructuredText tempo	
control for reStructuredText skeletons.	<f12> <f12> <f2></f2></f12></f12>	OTHER-WINDOW)  (pel-customize-generic-skels &optional OTHER-WINDOW)	skeletons text inserted by the skeleton text insertion commands for reStructuredText.  • If OTHER-WINDOW is non-nil (use <b>C-u</b> ), display in other window.	
Insert a file header	<f12> <f12> h</f12></f12>	(pel-rst-large-header)	Insert a large header includes all normal header fields plus separators.	
			nome page & license, markup for table of contents using the tempo skeleton mechanism. an move to the target points where extra text must be entered to complete the template.	
Toggle pel-tempo-mode	<f12> <f12> SPC</f12></f12>	(pel-tempo-mode &optional ARG)	Toggle <b>pel-tempo-mode</b> on/off. That mode activates key bindings to move to predefined hot-spots where template text must be added.	
See also: <u>▼ Mode Line</u>	The keys are: C-c . and. C-c , as well as (in graphics mode only): C-c C and C-c C-, When pel-tempo-mode is active the <b>pel-tempo-mode lighter</b> (‡) is shown on the status bar.  Solution when a skeleton is inserted via the execution of one of the pel-rst commands, the <b>pel-tempo-mode</b> is automatically activated.		s shown on the status bar.	
Jump to next tempo mark	• C-c M-f • C-c . • C-c C	(tempo-forward-mark)	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.	
Jump to previous tempo mark	• C-c M-b • C-c , • C-c C-,	(tempo-backward-mark)	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.	
Tempo Template Tag Insertion	<f12> <f12> <f12></f12></f12></f12>	(tempo-complete-tag &optional SILENT)	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", "case", etc) completely or partially and then hit <f12> <f12> <f12> 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.  A Since, at the moment, only one template is available in rst-mode, the usefulness of this command is limited for reStructuredText.</f12></f12></f12></f12></f12>			
Select Section Title Adornment Styles	The underlying character used for section line adornment is customizable. The number of available levels and whether the line is indented, has a line over and under the title line is selected by the adornment style. PEL supports 3 styles. The following commands can be used to select a style.			
Select default adornment style	<f12> A d <f11> SPC M-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 M-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 M-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>	
	1			

<u>Description</u>	<u>Keystroke</u>	Function	Note	
Section Title level	The rst.el library provides the	e rst-adjust command to create section	on adornment of the current line.	
adornment	The rst.el library provides the <b>rst-adjust</b> 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:			
<ul> <li>commands that insert section titles</li> </ul>	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			
	<ul> <li>It also provides commands to adorn a line to the same level as the previous section or a lower or nighter level. And then to increase or decrease the section level of the adornment of the current line.</li> <li>PEL provides 3 style of section adornments: default, Sphinx-Python and CRiSPer, which can be selected with commands.</li> </ul>			
	<ul> <li>PEL provides 3 style of section adornments: default, Sphinx-Python and CRISPer, which can be selected with commands.</li> <li>PEL remembers the preferred style inside the customizable variable: pel-rst-adornment-style.</li> <li>The rest.el provides the rst-preferred-adornment user option to select the adornment characters for the various sections.</li> <li>PEL code selects the value according to the adornment style you select.</li> <li>See section "Select Adornment Styles" above.</li> </ul>			
Adjust section level	• C-= • C-c C-=	(rst-adjust PFXARG)	Auto-adjust the adornment around point.  • Adjust/rotate the section adornment for the section title around point or promote/	
	• C-c C-a C-a		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 line at title level	<f12> t</f12>	(pel-rst-adorn-title)	Adorn current line with level-0 (title) reStructuredText section adornment.	
	<f11> SPC M-r t</f11>		If done at the top of the file, the first adorn line is placed on the first line of the file, a mark is left at the end of the title line and point is moved 2 lines below.	
			To return to the end of the title line, type M-     or <f6><f6>.</f6></f6>	
Adorn to specific level From level 1 to level 10	• <f12> 1</f12>	<ul><li>(pel-rst-adorn-1)</li><li>(pel-rst-adorn-2)</li></ul>	Adorn current line with level [1 to 10] reStructuredText section adornment.	
	• <f12> 9 • <f12> 0</f12></f12>	• (pel-rst-adorn-3)	The <f11> SPC M-r 1 to <f11> SPC M-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>	
	• <f11> SPC M-r 1</f11>	<ul><li>(pel-rst-adorn-7)</li><li>(pel-rst-adorn-8)</li></ul>	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	
	 • <f11> SPC M-r 0</f11>	• (pel-rst-adorn-9) • (pel-rst-adorn-0)		
Adorn current line: same section	• <f12> =</f12>	(pel-rst-adorn-same-level)	Adorn current line with the same level as the previous section.	
level as previous section	<f11> SPC M-r =</f11>		If the line is already adorned, update the adornment: adjust to previous section level.	
Adorn to higher section level	<f12> +</f12>	(pel-rst-adorn-increase-level)	Adorn current line at a higher-level that current if already adorned.	
	<f11> SPC M-r +</f11>		If the line is not already adorned, adorn it with a level higher than previous section.	
Adorn to lower section level	<f12> -</f12>	(pel-rst-adorn-decrease-level)	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 current line adornment	<f11> SPC M-r -</f11>	(nel-ret-adorn-refresh)	Refresh the adornment of the current line adjusting the underlines to the current length	
neirean current line adornment	<f12> r <f11> SPC M-r r</f11></f12>	(pel-rst-adorn-refresh)	Refresh the adornment of the current line, adjusting the underlining to the current length of the line.  This can be useful when changing the text on the line.	
Creating and Using	• This can be useful when changing the text on the line.  The following 3 PEL commands help write hyperlink of various forms:			
Hyperlinks	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.			
	When editing a buffer using the rst-mode, type the <f12> . keystroke to create a hyperlink.</f12>			
	• It uses the selected 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 I dedicated bookmark: that bookmark is created by the <f12> s keystroke. That helps identify an area inside the file where the next (or several)</f12>			
	1		t to use the same commands inside another mode, you can use the longer key chord that	
Set location of hyperlinks	uses the <f11> SPC M-</f11>	-r prefix (assuming that pel-use-rst- (pel-rst-set-ref-bookmark)	Set the reference bookmark for the currently edited file at point.	
oct location of hypermines	<f11> SPC M-r s</f11>	(por-ist-set-rei-bookmark)	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 place.			
Add an hyperlink for text at point	beginning of the first of the	e 2 lines.  (pel-rst-makelink &optional ARG)	Create a reStructuredText hyperlink prefix for the word at point or region's text.	
And all hypermix for text at point	<f11> SPC M-r .</f11>	(per-ist-makelink doptional / tild)	<ul> <li>If region active, use text of the region for the link, otherwise use the word at point.</li> <li>If an argument (which can be a C-u) is specified, use the embedded URI format.</li> </ul>	
4		I. use the named hyperlink format:	in an argument (which can be a <b>c-u</b> ) is specified, use the embedded on format.	
It's better to use the enclosing syntax ( <word>_) as it allows</word>	<ul> <li>If no argument is specified, use the named hyperlink format:</li> <li>if the region is a single word, and pel-rst-use-single-underscore-for-single-word-ref user-option is t, just append an underscore to make the lii</li> <li>if the region is several words, surround it with the "'" and the "'_" strings.</li> <li>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 onext empty line.</li> </ul>			
navigation to referenced text with pel-open-at-point (M- <f6>).</f6>				
Therefore you keep the user-option set to nil (the default).	The cursor is placed where		£6> to move back to previous location.	
Go to hyperlink location	<f12> g</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.	
	<f11> SPC M-r g</f11>		<ul> <li>Command pushes the mark on mark ring, type M-     or <f6><f6> to move back to previous location.</f6></f6></li> </ul>	
Activating URLs to	Emacs provides the <b>goto-ur</b> l	I-mode and the goto-url-prog-mode	e that turn URLs found in the current buffer into clickable buttons.	
browse and open files		ne following key sequences are availa	ble wheel point is over a URL button:	
See also:	If the URL is an email	I address a buffer to write an email to r FTP address the system browser is		
• ∑ File mngt • ∑ Navigation	• C-c C-n: move point	to the end of the next URL in the buf		
<u></u>	• C-c C-f : download		cal temporary file and visit the file. See (pel-open-url-at-point) above.	
Total Control of the		to-address . Mostly control the regex		
Toggle goto-address-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-addrress-prog- mode	<f11> f U</f11>	(goto-address-prog-mode &optional ARG)	Like 'goto-address-mode', but only for comments and strings.	
Open the URL (email or web	C-c RET	(goto-address-at-point &optional	Open the URL at point:	
page)		EVENT)	If URL is a web page: open it in a browser     If URL is a mail address:     Sept mill to address 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.	
Move to end of next URL in buffer	C-c C-n	(pel-goto-next-url)	Move point forward to the end of the next URL located in the current buffer.  • The global < \$6> Can key binding activates the goto-address-mode if it is not already.	
See also: ∑ Navigation	<f6> C-n</f6>		<ul> <li>The global <f6> C-n key binding activates the goto-address-mode if it is not already active.</f6></li> </ul>	
Move to beginning of previous URL in buffer	С-с С-р	(pel-goto-previous-url)	Move point backward to the beginning of the previous URL located in the current buffer.  • The global <f6> C-p key binding activates the goto-address-mode if it is not already</f6>	
See also: <u>Navigation</u>	<f11> C-p</f11>		active.	

<u>Description</u>	Keystroke	Function	Note	
Copy URL at point in temporary	<f11> f M-u</f11>	(pel-open-url-at-point)	Copy the URL at point to a local temporary file and visit that file.	
file and visit the file  See also:  File mngt			<ul> <li>A The download copy of the file does not have the same name and may not open with the proper mode because it won't have an extension. The HTML formatted files will be recognized by Emacs but most of the files won't be.</li> <li>Save the file somewhere else using the C-x C-w key sequence and identify the proper extension to activate the required major mode.</li> </ul>	
	C-c C-f		description of the URL and the goto-address-mode minor mode is active. Use <f11> f u or <f11> f U to activate this mode.</f11></f11>	
Open file or web-page whose name is at point  ** Command is generic and is also	• M-* • <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.  Supports glob characters, partial directory path. When multiple files are found it prompts using the method selected by pel-prompt-read-method user-option.  The 6y key-chord is available if pel-use-key-chord is non-nil. See Key-	
specialized for: • PI - C	This commond works control	socially by this class appaidled for you	Chords.	
• <u>क्रा - C++</u> • <u>क्रा - Erlang</u>	This command works generically but is also specialized for reStructuredText: Inside a rst-mode buffer, when the point is over a reStructuredText external hyperlink target reference, the command locates the reference and opens the file identified by the reference (unless N >= 100):  If the reference is a web URL, it opens the identified web using the system browser.  If the reference is a HTML file name that corresponds to the rendering of a local restructuredText file, it opens that reStructuredText file.			
• ֆլ - UNIX Shell				
Jump to referenced link (unless N >= 100) ►	<ul> <li>If the reference is a HTML file that does not correspond to a reStructuredText source, it opens that HTML file.</li> <li>If the reference is another type of file it opens that file.</li> <li>If the reference URL identifies a # <u>URI fragment</u> that identifies the name of a target file section, the command moves point to the section</li> <li>If the hyperlink refers to a title inside the document move point to that title instead of trying to open a file.</li> </ul>			
Delimiting characters 🖛	In general the command extracts the file or directory name, and possibly line and column numbers, from text at point and tries to open the file or directory.  • The generic mode extraction works by identifying the beginning & end of the file/directory/library/URL name string by delimiter characters, one of: tab, newline and: "`' ()[1}<-> ''"			
	<ul> <li>If embedded space(s) are allowed in the name, point must be located at the first of the 2 delimiter chars. Otherwise point can be anywhere in the name.</li> <li>The name may include glob characters</li> <li>The command uses a URL unchanged but uses the following heuristic to identify the exact location of the file/directory:         <ul> <li>In the file/dir name is an absolute path it uses that. Otherwise</li> <li>it builds a absolute path using the extracted relative path name inside the directory identified by the pel-open-file-at-point-dir user-option, which can be</li> </ul> </li> <li>The command uses a URL unchanged but uses the following heuristic to identify the exact location of the file/directory:         <ul> <li>It builds a absolute path using the extracted relative path name inside the directory (2) use current working directory or (3) use user-specified directory. It uses</li> </ul> </li> </ul>			
File identification heuristic				
<f11> f <f2> • &lt;</f2></f11>	<ul> <li>1) use parent directory of currently visited file, or use current working directory, 2) use current working directory, or 3) use user-specified directory. It uses the found file/dir name if it exists. Otherwise</li> <li>it searches for the relative file/dir name in directory tree under the root marker file identified by the pel-project-root-identifiers user-option which is something like .git, .hg, .project, .pel-project (the default). If it can find such a file in the above directories it searches the tree under the found root.</li> <li>If it finds several files it prompts using the current completion mode to allow selection of the appropriate name (see below) and opens the selected one.</li> <li>If it finds only one it opens that file. Otherwise,</li> <li>it prompts showing the name searched and provide the following choices: 1) create the file with specified name, 2) edit the name to search again, 3) use</li> </ul>			
	• It prompts showing the name searched and provide the following choices: 1) create the file with specified name, 2) edit the name to search again, 3) use the name found and search for an Emacs library file with that name, or 4) quit.			
	The command opens the extracted name according to this heuristic:  • If the string is a properly formatted <b>URL</b> , it opens it using the OS default browser (even if a optional numeric argument specified otherwise), otherwise  • if the string is a file or directory name it opens it.  • If the file name is followed by <b>line and column numbers</b> the point is moved to that position in the buffer.			
	When finding several file names, the command lists them and prompts using the method selected by <b>pel-prompt-read-method</b> user-option.			
	1	· _ ·	You can select a more powerful ivy prompting instead.  The second is a select a more powerful ivy prompting instead.  The second is a select a more powerful ivy prompting instead.	
Onland world: Gland and and	Note that the command	shows all files found by the specified	d search method, it does not only use the first one found.	
Select multi-file selection method 🖛		potential duplication in header file na		
	The command opens the file in the window selected by the following logic controlled by presence or absence of typed numerical prefix arguments:  • Select target window:  • With the temperature.			
Select target window 🖛	If no window holds the window, if 2: use the     With prefix numeric arguments.	nat file, select the target window acco other window, if 3 or more, use the co ument N:	int to that window and to the line column coordinates if specified. rding to the number of editable windows in frame: if 1, split that window and use the new urrent window.	
	<ul> <li>N &lt; 0 : create a new v</li> <li>(abs N) &gt; 20: then op</li> <li>N = 0: use the 'other'</li> </ul>	en the <b>directory</b> instead of the file. In	nterpret the window position from the N value adjusted: N-20 (or N+20 if N is negative)	
			et window based on the number of editable windows in frame:	
N>20 : open the directory	if 2 windows:	use the other window, ws: use the current window.	······································	
	• N is: 8: up, 2: down,	4:left, 5:current, 6:right.	lication (with N=29 or N=-29, open the file's directory with the OS associated application	
	(eg. macOS Finder, W	Vindows Explorer). If this is a URL, or	the description of the state of	
See function docstring for more info.	_ ·	xistent or dedicated window is not al	·	
Compile/render			using a command identified by the <b>pel-rst-compiler</b> user-option.  In the restructuredText using the <b>Docutils front-end tool</b> : <b>rst2html</b> .	
Compile the file: generate	<f12> c</f12>	(pel-rst-compile)	"Compile" the reStructuredText file into the rendered format (html or something else).	
rendered file  See also:  Compilation Mode	<ul> <li>The compiler command is selected by the pel-rst-compiler user-option. It defaults to pel-rst2html, which is treated specially: if it is not found in PATH, PEL automatically adjusts the command line to use the script provided in its bin directory. That script uses rst2html to generate an html file.</li> <li>You can specify any other command line with or without arguments, as long as the name of the processed reStructuredText file is the last argument of that command (because PEL appends the name of that file to the command).</li> <li>All errors are shown in a *compilation* buffer. See <u>© Compilation Mode</u> for more info.</li> </ul>			
Move point to next compilation/	• C-x `		Move to the next error message and visit the corresponding source code.	
rendering error	• M-g n • M-g M-n		<ul> <li>If all the error messages parsed so far have been processed already, the message buffer is checked for new ones.</li> <li>A prefix ARG specifies how many error messages to move; negative means move back to previous error messages.</li> <li>Just C-u as a prefix means reparse the error message buffer and start at the first error.</li> </ul>	
Move point to previous	• M-g p	(previous-error &optional N)	Visit previous previous error message and corresponding source code.	
compilation/rendering error	• м-д м-р		Prefix arg N says how many error messages to move backwards (or forwards, if negative).	

## rst-mode - References

Description & URL	Notes
Emacs Support for reStructuredText	
reStructuredText	Main page for all reStructuredText documents.
reStructuredText markup Specifications	Formal markup specifications.
Sphinx Python Documentation Generator	Sphinx — Documentation Contents     Sphinx — Documentation — Sections