


























Emacs support for the Seed7 Programming Language



Description	Keystroke	Function	Note
Seed7 Editing	PEL supports for the Seed7 programming language is experimental and not yet documented except for what you see here. <div> The seed7-mode external package is installed when  the pel-use-seed7 user-option is set to t.</div> <ul style="list-style-type: none">Seed7 files are files with .sd7 and .s7i extensions. The seed7-mode supports some navigation commands. It also supports imenu (see ℹ Menus) and ℹ Speedbar to identify syntactic elements. <div>  I am currently writing seed7-mode and it's in early state. It supports code highlighting and some comments control and navigation commands.</div> <ul style="list-style-type: none">I will develop more later, as time permits.		
Last updated on:	2025-04-26		
Open this PDF file. See also: ℹ Help/Info	<f11> SPC 7 <f1>	(pel-help-pdf &optional OPEN-WEB-PAGE)	Open the ℹ1 - Seed7 local PDF. If the prefix argument (like C-u or M--) is used, then it opens the remote GitHub hosted raw PDF instead. If the pel-flip-help-pdf-arg user-option is set it's the other way around.
	<f12> <f1>		
ℹ Customize PEL Seed7 support	<f11> SPC 7 <f2>	(pel-customize-pel &optional OTHER-WINDOW)	Customize PEL Seed7 support. <ul style="list-style-type: none">If OTHER-WINDOW is non-nil (use C-u), display in another window.
	<f12> <f2>		
ℹ Customize Emacs Seed7 support	<f11> SPC 7 <f3>	(pel-customize-library &optional OTHER-WINDOW)	Customize Emacs Seed7 support: seed7 <ul style="list-style-type: none">If OTHER-WINDOW is non-nil (use C-u), display in another window.
	<f12> <f3>		
Comments	Several commands are unavailable to insert and manipulate commands, listed in ℹ Comments . Some are duplicated here for convenience. The ones specific to seed7-mode are listed first.		
Toggle between Seed7 (* block *) and # line style	C-c ;	(seed7-toggle-comment-style &optional ARG)	Toggle the Seed7 comment style between block and line comments. <ul style="list-style-type: none">Optional numeric ARG, if supplied, switches to block comment style when positive, to line comment style when negative, and just toggles it when zero or left out. Note: the default style for all Seed7 buffers is controlled by the ' seed7-uses-block-comment ' customizable user-option. The default is line style comments.
Insert, realign, comment/uncomment region	M- ;	(comment-dwim ARG)	Insert or realign comment on current line (or region if a region is active). <ul style="list-style-type: none">On a single line, the comment is placed <i>after</i> the code.C-u M- ; executes comment-kill
Toggle display of comments in buffer or active region See also: ℹ Comments	<f11> ; ;	(hide/show-comments-toggle &optional START END)	Toggle hiding/showing of comments in the active region or whole buffer. <ul style="list-style-type: none">If the region is active then toggle in the region. Otherwise, in the whole buffer. <div> This requires the hide-comnt.el package (see ℹ Comments).  PEL activates it when the pel-use-hide-comnt user option is t.</div>
Change comment style for buffer	<f11> ; s	(pel-comment-style &optional CUSTOMIZE)	Select a comment style for the buffer: prompts with the list of available styles, showing the currently used one. Apply the choice to the current buffer. <ul style="list-style-type: none">With C-u prefix, open the customize buffer to control selection of the default comment style for all buffers (the comment-style user option).
As of Emacs 30, Emacs supports 8 different comment styles, listed here: 	Emacs supports several comment styles, as specified by the comment-styles user-option (which can be modified). Some of these styles only take effect when a region of several lines is comments. By changing the style you can create the boxed comments, for instance and also uncomment the box comment with comment-swim (bound to M- ;) and then change for another comment style in the same buffer. <ul style="list-style-type: none">The style selected by the command only affects the current buffer. It is not persistent. The persistent setting is the comment-style user option. <div><ul style="list-style-type: none">0 = plain: Start in column 0 (do not indent), as in Emacs-201 = indent-or-triple: Start in column 0, but only for single-char starters2 = indent: Full comment per line, ends not aligned3 = aligned: Full comment per line, ends aligned4 = box: Full comment per line, ends aligned, + top and bottom5 = extra-line: One comment for all lines, end on a line by itself6 = multi-line: One comment for all lines, end on last commented line7 = box-multi: One comment for all lines, + top and bottom</div>		
Compilation	The Seed7 source code is either interpreted or compiled. In both cases you can verify it's validity by performing a static check of the code, an operation that does not generate any binary file but perform the same language checking that the compiler will do.		
Static check or compile Seed7 file	<f12> c	(seed7-compile &optional COMPILE)	Static check current Seed7 file, show errors in compilation-mode buffer. <ul style="list-style-type: none">If optional COMPILE argument set, compile the file to executable instead.For example: type C-u <f12> c for compiling the file. Without the C-u prefix it just static checks the file, an operation that is much faster.The static analysis is performed by the command identified by the seed7-checker user-option, which defaults to s7-check. You can specify any command with or without its path.The compilation is performed by the command identified by the seed7-compiler user-option, which defaults to s7c. You can specify any command with or without its path.Any detected error is shown in a *compilation* compilation-mode buffer. Use it to navigate to the line of the code in error.

Description	Keystroke	Function	Note
Code Navigation	 Code navigation under Seed7 with seed7-mode is still under construction but several commands are already available. <ul style="list-style-type: none"> PEL provides some extra key bindings. 		
Shift-Selection	If you press and hold the shift key while typing a movement command, that sets the mark before moving point (Emacs name for cursor) so that the region extends from the original point to its new position. This Shift-Selection is called “ <i>Shift-Marking</i> ” in this document. <ul style="list-style-type: none"> Shift selection is supported by some navigation commands, not all. The following symbols are used to identify whether the command supports shifts selection: <ul style="list-style-type: none">  This command supports shift selection in GUI and terminal mode.  This command supports shift selection only in GUI mode.  This command supports shift selection in GUI mode and also in terminal mode under some conditions (described in the description cell for the command).  This command does not support shift selection. Sometimes for this you can first set the mark before moving. Pressing the Shift key when using the key binding for commands that do not show any of these 3 arrows have no impact on the shift selection (and may be inappropriate for the command). 		
Move Point	The following sub-sections describe how to navigate across various types of textual and syntactical entities.		
<ul style="list-style-type: none"> by defun 	The commands move point by Seed7 function and procedure definitions. <div>  In PEL: <ul style="list-style-type: none"> The <f12> cursor key mappings use <up> and <down> to move to the beginning of the function, procedure or other blocks. The <f6> cursor key mapping use <up> and <down> to move to the beginning of the function, procedure </div>		
Backward to beginning of defun  	<ul style="list-style-type: none"> <f6> <up> 	(beginning-of-defun &optional ARG)	Move backward to the beginning of a defun. <ul style="list-style-type: none"> With ARG, do it that many times. Negative ARG means move forward to the ARGth following beginning of defun. Prints the name of the function or procedure in the message area. On successful move, you can move back to original position by typing M-` , <f6> <f6> or <f11> . ` Supports shift selection
	<ul style="list-style-type: none"> C-M-a C-M-<home> C-[C-a Esc C-a 		
Forward to end of defun  	<ul style="list-style-type: none"> <f6> <down> 	(end-of-defun &optional ARG)	Move forward to next end of defun. <ul style="list-style-type: none"> With argument, do it that many times. Negative argument -N means move back to Nth preceding end of defun. Prints the name of the function or procedure in the message area. On successful move, you can move back to original position by typing M-` , <f6> <f6> or <f11> . ` Supports shift selection
	<ul style="list-style-type: none"> C-M-e C-M-<end> C-[C-e Esc C-e 		
Forward to start of next defun 	<f6> M-<down>	(seed7-beg-of-next-defun &optional N SILENT DONT-PUSH-MARK)	Move forward to the beginning of the next function or procedure. <ul style="list-style-type: none"> With optional argument N, repeat the search that many times. Supports shift selection. Move back to previous position with M-` , <f6> <f6> or <f11> . `
Backward to end of previous define   will be replaced	<f6> <left>	(pel-end-of-previous-defun &optional SILENT DONT-PUSH_MARK)	Move backwards to the end of the previous function definition. <ul style="list-style-type: none"> Issue user error not find end of previous function unless SILENT is non-nil. If the end of previous function is found, push the start location to the mark ring unless DONT-PUSH_MARK is non-nil. <ul style="list-style-type: none"> Move back to previous position with M-` , <f6> <f6> or <f11> . `
Forward to end of current block statement 	<f12> <down>	(seed7-to-block-forward)	Move forward from the beginning of a Seed7 block to its end. <ul style="list-style-type: none"> supports the Seed7: if/end if, block/end block, case/end case, enum/end enum, for/end for, repeat/until, struct/end struct, while/end while. It also supports moving to the end of a function or a procedure. Supports shift selection. Move back to previous position with M-` , <f6> <f6> or <f11> . `
Backward to beginning of current block statement 	<f12> <up>	(seed7-to-block-backward)	Move backward from the end of a Seed7 block to its beginning. <ul style="list-style-type: none"> supports the Seed7: if/end if, block/end block, case/end case, enum/end enum, for/end for, repeat/until, struct/end struct, while/end while. It also supports moving to the end of a function or a procedure. Supports shift selection. Move back to previous position with M-` , <f6> <f6> or <f11> . `

Emacs & Seed7 — References

Document	Notes	
The Seed7 Programming Language	<ul style="list-style-type: none"> Seed7 @ Wikipedia Seed7 Home Seed7 @ Github <ul style="list-style-type: none"> Seed7 Manual Seed7 Language Reference 	
	<ul style="list-style-type: none"> Seed7 @ reddit Seed7 @ Rosetta code 	
Presentations	<ul style="list-style-type: none"> The Seed7 Programming Language @ Youtube The Seed7 Programming Langage Presentation at CPP Vienna @ Youtube Another speech about the Seed7 Programming Language 	
	Modern Extensible Languages. Daniel Zingaro, McMaster U. April 11, 2007 (pdf)	
Emacs support  is partial, not yet completed.	seed7-mode @ Github	
Other tools that support Seed7	<ul style="list-style-type: none"> ripgrep a very fast grep replacement - supports seed7 file types with this pull request accepted April 7 2025 <ul style="list-style-type: none"> With this version of ripgrep, you can use deadgrep to identify Seed7 files by name in Emacs. See § Grep ugrep another very fast grep replacement - supports seed7 files with this pull request . 	