

















## Emacs support for Forth

Description	Keystroke	Function	Note
<b>Forth programming Language Support</b>	Support for the <b>Forth programming language</b> is minimal: you can activate the forth-mode package by setting the <b>pel-use-forth</b> user option. <ul style="list-style-type: none"> <li>Files with the .f, .fs, .fth and .4th are recognized as Forth source files and will automatically activate forth-mode if the package has been activated via that user-option.</li> <li>Generic programming language features like template text insertion handle Forth comment style. See <a href="#">↗ Inserting Text</a> .</li> <li> Forth support is provided by <a href="#">forth-mode</a> external package  automatically downloaded and installed by PEL when the <b>pel-use-forth</b> user option is <b>t</b>.</li> <li>This mode supports Emacs <a href="#">↗ Speedbar</a> : it shows Forth variables and words.</li> </ul>		
<b>Open this PDF file.</b> See also: <a href="#">↗ Help/Info</a>	<div>&lt;f11&gt; SPC f &lt;f1&gt;</div> <div>&lt;f12&gt; &lt;f1&gt;</div>	(pel-help-pdf &optional OPEN-WEB-PAGE)	Open the  <b>Forth</b> local PDF. If the prefix argument (like <b>C-u</b> or <b>M--</b> ) is used, then it opens the remote GitHub hosted raw PDF instead. If the <b>pel-flip-help-pdf-arg</b> user-option is set it's the other way around.
<a href="#">↗ Customize</a> PEL Forth support	<div>&lt;f11&gt; SPC f &lt;f2&gt;</div> <div>&lt;f12&gt; &lt;f2&gt;</div>	(pel-customize-pel &optional OTHER-WINDOW)	Customize PEL Forth support. <ul style="list-style-type: none"> <li>If OTHER-WINDOW is non-nil (use <b>C-u</b>), display in another window.</li> </ul>
<b>Forth Language Reference Access</b>	The following commands connect to a web site to retrieve specification information about Forth words.		
<b>Forth '94 word specification</b>	<b>C-c C-d 1</b>	(forth-spec-lookup-1994 NAME)	View the documentation on NAME from the ANS'94 Forth Standard <ul style="list-style-type: none"> <li>Prompts for Forth word. Supports completion.</li> <li>Connects to <a href="http://lars.nocrew.org/dpans/dpans.htm">http://lars.nocrew.org/dpans/dpans.htm</a>, a draft of ANSI Forth '94.</li> </ul>
<b>Forth 2012 word specification</b>  	<b>C-c C-d 2</b>	(forth-spec-lookup-2012 NAME)	View the documentation on NAME from the Forth 2012 Standard. <ul style="list-style-type: none"> <li>Attempts to access <a href="http://www.forth200x.org/documents/html/alpha.html">http://www.forth200x.org/documents/html/alpha.html</a>  which, at the time of this writing (early 2021) does not exists. There are other files on this web site that contain information about Forth 2012 such as <a href="http://www.forth200x.org/documents/forth-2012.pdf">http://www.forth200x.org/documents/forth-2012.pdf</a>. </li> </ul>
<b>Navigation inside Forth Code</b> See also: <a href="#">↗ Navigation</a>	Basic navigation commands are available: both standard Emacs navigation and extra commands provided by PEL, as listed below. <ul style="list-style-type: none"> <li>This mode supports Emacs <a href="#">↗ Speedbar</a> : it shows Forth variables and words.</li> </ul>		
<b>Forward to start of next word definition</b>	<div>&lt;f6&gt; &lt;down&gt;</div>	(pel-beginning-of-next-defun &optional SILENT DONT-PUSH_MARK)	Move forward to the beginning of the next word definition. <ul style="list-style-type: none"> <li>Beeps if does not find beginning of next function unless SILENT is non-nil.</li> <li>If the beginning of next function is found, push the start location to the mark ring unless DONT-PUSH_MARK is non-nil.                             <ul style="list-style-type: none"> <li>Move back to previous position with <b>M-`</b>.  this should but does not work in Forth.</li> </ul> </li> </ul>  Shift marking is available : hold Shift after typing <b>&lt;f6&gt;</b> .
<b>Backward to end of previous word definition</b>	<div>&lt;f6&gt; &lt;left&gt;</div>	(pel-end-of-previous-defun &optional SILENT DONT-PUSH_MARK)	Move backwards to the end of the previous word definition. <ul style="list-style-type: none"> <li>Beeps if does not find end of previous function unless SILENT is non-nil.</li> <li>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"> <li>Move back to previous position with <b>M-`</b>.  this should but does not work in Forth.</li> </ul> </li> </ul>  Shift marking is available.
<b><a href="#">Backward to beginning of function definition</a></b>	<ul style="list-style-type: none"> <li><b>C-M-a</b></li> <li><b>C-M-&lt;home&gt;</b></li> <li><b>&lt;f6&gt; &lt;up&gt;</b></li> <li><b>C-[ C-a</b></li> <li><b>Esc C-a</b></li> </ul>	(beginning-of-defun &optional ARG)	Move backward to the beginning of a word. <ul style="list-style-type: none"> <li>With ARG, do it that many times. Negative ARG means move forward to the ARGth following beginning of word.</li> </ul>  Shift marking is available in graphics mode, <b>not in terminal mode</b> (for <b>C-M-a</b> and <b>C-M-&lt;home&gt;</b> ). It's always available for <b>&lt;f6&gt; &lt;up&gt;</b> : hold Shift after typing <b>&lt;f6&gt;</b> .
<b><a href="#">Forward to end of function and class definition</a></b>	<ul style="list-style-type: none"> <li><b>C-M-e</b></li> <li><b>C-M-&lt;end&gt;</b></li> <li><b>&lt;f6&gt; &lt;right&gt;</b></li> <li><b>C-[ C-e</b></li> <li><b>Esc C-e</b></li> </ul>	(end-of-defun &optional ARG)	Move forward to next end of word. With argument, do it that many times. Negative argument -N means move back to Nth preceding end of word  Shift marking is available in graphics mode, <b>not in terminal mode</b> (both keys).
<b>Marking Forth Words</b>			
<b>Mark current Forth word</b>  See also: <a href="#">↗ Marking</a>	<b>C-M-h</b>	(mark-defun &optional ALLOW-EXTEND)	Put mark at end of this word definition, point at beginning. <ul style="list-style-type: none"> <li>The Foth word marked is the one that contains point or follows point.</li> <li>With positive ARG, mark this and that many next words; with negative ARG, change the direction of marking.</li> <li>If the mark is active, it marks the next or previous word(s) after the one(s) already marked.</li> </ul>
<b>Forth Evaluation</b>	When a Forth interpreter is available, the following commands use it to provide interactive evaluation within Emacs. <ul style="list-style-type: none"> <li>The first time a command requiring a Forth executable is require it prompts for one then uses it.</li> </ul>		
<b>Open a Forth shell</b>	<div>&lt;f11&gt; z f</div>	(run-forth)	Start an interactive forth session. <ul style="list-style-type: none"> <li>Prompt for a Forth executable.                             <ul style="list-style-type: none"> <li><a href="#">gforth</a> is a good free implementation.                                     <ul style="list-style-type: none"> <li>On macOS, you can install it with <b>brew install gforth</b> in a terminal shell.</li> </ul> </li> <li> Notice that it is integrated with the Home-brew Emacs installation and it will upgrade your Homebre-based Emacs unless its pinned (in which case Homebrew won't install gforth).</li> </ul> </li> </ul>  Requires the <b>forth-mode</b> external package  PEL installs and activates when the <b>pel-use-forth</b> user option is <b>t</b> . It also requires a Forth interpreter (which must be installed separately)
<b>Evaluate Forth Expression</b>	<b>C-c C-e</b>	(forth-eval-last-expression)	Evaluate Forth expression at point.
<b>Kill Forth process</b>	<b>C-c C-k</b>	(forth-kill &optional BUFFER)	Kill Forth process associated with current buffer.
<b>Load Forth File</b>	<b>C-c C-l</b>	(forth-load-file FILE)	Load specified file in the Forth interpreter.
<b>Evaluate region of Forth code</b>	<b>C-c C-r</b>	(forth-eval-region START END)	Evaluate marked region of Forth code.
<b>SEE code of current word</b>	<b>C-c C-s</b>	(forth-see WORD)	Execute the Forth <b>SEE</b> command on the current word, accessing the code of that word. <ul style="list-style-type: none"> <li>Uses the word at point , showing the result inside the *see* buffer.</li> </ul>
<b>Opens the Forth process output buffer</b>	<b>C-c C-z</b>	(forth-switch-to-output-buffer)	Opens the Forth process output buffer
<b>Evaluate Forth expression.</b>	<b>C-c :</b>	(forth-eval STRING)	Prompts for a Forth expression. On RET uses the Forth interpreter to evaluate it and display the result in the minibuffer.
<b>Evaluate the current Forth word.</b>	<b>C-M-x</b>	(forth-eval-defun)	Evaluate the current word.
<b>Evaluate current Forth expression</b>	<b>C-x M-e</b>	(forth-eval-last-expression-display-output)	Evaluate current Forth expression

Description	Keystroke	Function	Note
Indenting Forth Code			
Indent expression	C–M–q	(prog-indent-sexp &optional DEFUN)	Indent the expression after point. When interactively called with prefix, indent the enclosing defun instead.
Commenting Forth Code			
Comment/uncomment  See also: <a href="#">Σ Comments</a>	M– ;	(comment-dwim ARG)	Comment line or region. If line/region is already commented, uncomment it. <ul style="list-style-type: none"> <li>Forth comments are quite flexible. This command uses uses the \ word at the beginning of a line and ( ) after now-whitespace.</li> <li>With PEL, use the &lt;f11&gt; ; ? command to get a full list of the variables used to control Forth comments.</li> </ul>

## Forth— References

Document	Notes
<a href="#">Forth Programming Language</a>	Forth is a stack-based programming language designed Designed by <b>Charles H. Moore</b>
<a href="#">The Forth Programming Language - Wikipedia</a>	
<a href="#">Forth Books @ forth.com</a>	Links to several good Forth Books, including <b>Starting Forth</b> the original 1981 Forth book by <a href="#">Leo Brodie</a> .
<a href="#">forth-standard.org</a>	Get the latest information about Forth on this web site.
<a href="#">ANSI Forth '94 Draft Specification</a>	
<a href="#">Forth 200x Extension Proposals</a>	
<a href="#">Forth Implementations</a>	
<a href="#">GForth @ Wikipedia</a>	GNU Forth, a free implementation
<a href="#">GForth @ GNU</a>	
<a href="#">SwiftForth</a>	A commercial implementation.