
















Emacs support for Forth

Description	Keystroke	Function	Note
<b>Forth programming Language Support</b>	Support for the <a href="#">Forth programming language</a> 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 t.</li><li>This mode supports Emacs <a href="#">§ Speedbar</a>: it shows Forth variables and words.</li></ul>		
Open this PDF file. 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 local copy of the <a href="#">§1 - Forth</a> PDF file unless a command prefix (like <b>C-u</b> ) was used. In that case it opens the Github-hosted file instead.
<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.		
Forth '94 word specification	C-c C-d 1	(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>
Forth 2012 word specification  	C-c C-d 2	(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>		
Forward to start of next word definition	<f6> <down>	(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.
Backward to end of previous word definition	<f6> <left>	(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.
Backward to beginning of function definition	<ul style="list-style-type: none"><li>C-M-a</li><li>C-M-&lt;home&gt;</li><li>&lt;f6&gt; p</li><li>&lt;f6&gt; &lt;up&gt;</li><li>C-[ C-a</li><li>Esc C-a</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> ). However <b>&lt;f6&gt; p</b> handles Shift-marking fine in terminal mode.
Forward to end of function and class definition	<ul style="list-style-type: none"><li>C-M-e</li><li>C-M-&lt;end&gt;</li><li>&lt;f6&gt; &lt;right&gt;</li><li>C-[ C-e</li><li>Esc C-e</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>			
Mark current Forth word  See also: <a href="#">§ Marking</a>	C-M-h	(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>		
Open a Forth shell	<f11> z f	(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>gforth 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 <a href="#">forth-mode</a> external package  PEL installs and activates when the <b>pel-use-forth</b> user option is t. It also requires a Forth interpreter (which must be installed separately)
Evaluate Forth Expression	C-c C-e	(forth-eval-last-expression)	Evaluate Forth expression at point.
Kill Forth process	C-c C-k	(forth-kill &optional BUFFER)	Kill Forth process associated with current buffer.
Load Forth File	C-c C-l	(forth-load-file FILE)	Load specified file in the Forth interpreter.
Evaluate region of Forth code	C-c C-r	(forth-eval-region START END)	Evaluate marked region of Forth code.
SEE code of current word	C-c C-s	(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>
Opens the Forth process output buffer	C-c C-z	(forth-switch-to-output-buffer)	Opens the Forth process output buffer
Evaluate Forth expression.	C-c :	(forth-eval STRING)	Prompts for a Forth expression. On RET uses the Forth interpreter to evaluate it and display the result in the minibuffer.
Evaluate the current Forth word.	C-M-x	(forth-eval-defun)	Evaluate the current word.
Evaluate current Forth expression	C-x M-e	(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="#">⌘ Comments</a>	M– ;	(comment-dwim ARG)	Comment line or region. If line/region is already commented, uncomment it. • Forth comments are quite flexible. This command uses uses the \ word at the beginning of a line and ( ) after now-whitespace. • With PEL, use the <f11> ; ? command to get a full list of the variables used to control Forth comments.

### Forth— References

Document	Notes
<a href="#">Forth Programming Language</a>	Forth is a stack-based programming language designed Designed by <a href="#">Charles H. Moore</a>
<a href="#">The Forth Programming Language - Wikipedia</a>	
<a href="#">Forth Books @ forth.com</a>	Links to several good Forth Books, including <a href="#">Starting Forth</a> 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.