

🚧 Emacs support for Dart 🚧

Description	Keystroke	Function	Note
Dart Editing	Emacs does not provide any built-in mode for Dart programming language programming language. PEL provides support for Dart via the following external packages activated by the <code>pel-use-dart</code> user-option set to the following values: 📦 dart-mode  when <code>pel-use-dart</code> is set to <code>t</code>. See  Tree Sitter and  Tree-sitter. 📦 dart-ts-mode  when <code>pel-use-dart</code> is set to <code>with-tree-sitter</code>, Emacs >= 30 and <code>pel-use-tree-sitter</code> is <code>t</code>. Best choice. PEL: - Promotes the use of dart-ts-mode for Emacs >= 30. - Associates the <code>.dart</code> file extensions with the Dart major mode currently used.  iMenu is supported by dart-ts-mode but not by dart-mode. - Adds Dart support for  Speedbar for dart-ts-mode when the <code>pel-use-speedbar</code> user-option is turned on. - Activates PEL functions keys under the <code><f12></code> prefix inside Dart buffers. The official Dart indentation scheme strictly imposes a 2-space indentation scheme despite people requesting more flexibility. 😓 If this is a problem for you, PEL can help 😊  By setting <code>pel-indent-with-tabs-mode-for-dart</code> to a value between 2 and 8, PEL will automatically activates the <code>pel-indent-with-tabs-mode</code> minor mode for Dart buffers using the selected value as the visual indentation rendering width. You will be able to edit the buffer with the tab-based indentation scheme, with further ability to dynamically change the indentation rendered width with the <code>pel-set-tab-width</code> command (bound to <code><f11></code> <code><tab></code> <code>w</code>). The file is saved using the original 2-space indentation scheme! See img alt="tree-sitter icon" data-bbox="288 208 303 223"/> Indentation for more details.		
 Edit Dart code using wider indentation in buffer, keep 2-space indent in files.	Last updated on: 2025-11-23		
Open this PDF file. See also: img alt="tree-sitter icon" data-bbox="288 246 303 261"/> Help/Info 	<f11> SPC d <f1> <f12> <f1>	(pel-help-pdf & optional OPEN-WEB-PAGE)	Open the  - Dart local PDF. If the prefix argument (like <code>C-u</code> or <code>M--</code>) is used, then it opens the remote GitHub hosted raw PDF instead. If the <code>pel-flip-help-pdf-arg</code> user-option is set it's the other way around.
img alt="tree-sitter icon" data-bbox="288 269 303 284"/> Customize PEL Dart support 	<f11> SPC d <f2> <f12> <f2>	(pel-customize-pel & optional OTHER-WINDOW)	Customize PEL Dart support. If OTHER-WINDOW is non-nil (use <code>C-u</code>), display in another window.
img alt="tree-sitter icon" data-bbox="288 299 303 314"/> Customize Emacs Dart support 	<f11> SPC d <f3> <f12> <f3>	(pel-customize-library & optional OTHER-WINDOW)	Customize Emacs Dart support: <code>dart</code>, <code>dart-ts</code> If OTHER-WINDOW is non-nil (use <code>C-u</code>), display in another window.
Toggle between classic and Tree-Sitter major mode	PEL supports switching from classic to tree-sitter based mode with the following command. See  Tree Sitter		
	<f11> C-t C-t	(pel-treesit-toggle-mode)	Toggle the major mode between the classic mode and the Tree-Sitter based mode. If the other major mode is not available the command signals a user error. 👉 Use the repeat command (bound to <code><f5></code> under PEL) to quickly toggle from the classic to the Tree-Sitter major mode and compare the impact of syntax highlighting.
Show PEL setup information for the major mode.	<f11> ? / <f11> SPC d ? <f12> ?	(pel-mode-setup-info & optional APPEND) (pel-dart-setup-info & optional APPEND)	Display Dart setup information inside a "pel-dart-info" buffer with buttons providing quick access to the customization buffer of each variable shown. The information shown includes the value and interpretation of: <code>pel-use-dart</code> (whether the classic or tree-sitter based major mode is used). - indentation and hard tab control customizable user-options To append information in the buffer instead of clearing the previous content type any prefix argument (such as <code>C-u</code>) before the command keystroke.
Comments			
Toggle display of comments in buffer or active region See also: img alt="tree-sitter icon" data-bbox="288 481 303 496"/> Comments 	<f11> ; ;	(hide/show-comments-toggle & optional START END)	Toggle hiding/showing of comments in the active region or whole buffer. If the region is active then toggle in the region. Otherwise, in the whole buffer. 📦 This requires the hide-comnt.el package (see img alt="tree-sitter icon" data-bbox="288 498 303 513"/> Comments).  PEL activates it when the <code>pel-use-hide-comnt</code> user option is <code>t</code>.
Insert, realign, comment/uncomment region See also: img alt="tree-sitter icon" data-bbox="288 523 303 538"/> Comments With PEL: Comment the current line with <code>M-0 M-;</code> 	M-;	(comment-dwim ARG) (pel-comment-dwim ARG)	Insert or realign comment on current line (or region if a region is active). - On a single line, the comment is placed <i>after</i> the code. <code>C-u M-;</code> executes comment-kill Same as <code>comment-dwim</code> but comments the current line with a numeric ARG or 0.
Indentation	See also: img alt="tree-sitter icon" data-bbox="288 553 303 568"/> Indentation	If you suffer from this issue, you can use PEL or the tab-based-indent to work around the problem.	
Toggle indent-with-tabs-mode 	<f11> <tab> i m	(pel-indent-with-tabs-mode & optional ARG)	Toggle minor mode that automatically changes to tab-based indentation allowing wider rendering via larger tab widths. Save files without the tabs, with original indent scheme.
A PEL independent minor mode exists: tab-based-indent package; it provides the same functionality but outside of PEL.			

Emacs & Dart — References

Document	Notes
The Dart programming language	Dart @ Wikipedia Dart home Dart @ Github
Dart Documentation	Introduction to Dart Effective Dart Dart Libraries Dart SDK Asynchronous programming: futures, async, await Packages
Dart Projects	Awesome Dart