

# Emacs support for Dart

Description	Keystroke	Function	Note
<b>Dart Editing</b>  See also: <a href="#">Help/Info</a>	Emacs does not provide any built-in mode for <a href="#">Dart programming language</a> 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:  📦 <a href="#">dart-mode</a> ✅ when <code>pel-use-dart</code> is set to <b>t</b> . 📦 <a href="#">dart-ts-mode</a> ✅ when <code>pel-use-dart</code> is set to <b>with-tree-sitter</b> , <code>Emacs &gt;= 30</code> and <code>pel-use-tree-sitter</code> is <b>t</b> . See <a href="#">Tree Sitter</a> and <a href="#">Tree-sitter</a> . Best choice. PEL: <ul style="list-style-type: none"><li>Promotes the use of <a href="#">dart-ts-mode</a> for Emacs <math>\geq 30</math>.</li><li>Associates the <code>.dart</code> file extensions with the Dart major mode currently used.</li><li><a href="#">iMenu</a> is supported by <a href="#">dart-ts-mode</a> but not by <a href="#">dart-mode</a>.</li><li>Adds Dart support for <a href="#">Speedbar</a> for <a href="#">dart-ts-mode</a> when the <code>pel-use-speedbar</code> user-option is turned on.</li><li>Activates PEL functions keys under the <code>&lt;f12&gt;</code> prefix inside Dart buffers.</li></ul>		
Last updated on:	2025-11-05		
<b>Open this PDF file.</b> See also: <a href="#">Help/Info</a>	<code>&lt;f11&gt; SPC d &lt;f1&gt;</code> <code>&lt;f12&gt; &lt;f1&gt;</code>	( <code>pel-help-pdf</code> &optional OPEN-WEB-PAGE)	Open the <a href="#">Dart</a> 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 <code>pel-flip-help-pdf-arg</code> user-option is set it's the other way around.
<a href="#">Customize PEL</a> Dart support	<code>&lt;f11&gt; SPC d &lt;f2&gt;</code> <code>&lt;f12&gt; &lt;f2&gt;</code>	( <code>pel-customize-pel</code> &optional OTHER-WINDOW)	Customize PEL Dart 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>
<a href="#">Customize</a> Emacs Dart support	<code>&lt;f11&gt; SPC d &lt;f3&gt;</code> <code>&lt;f12&gt; &lt;f3&gt;</code>	( <code>pel-customize-library</code> &optional OTHER-WINDOW)	Customize Emacs Dart support: dart, dart-ts <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>Toggle between classic and Tree-Sitter major mode</b>	PEL supports switching from classic to tree-sitter based mode with the following command. See <a href="#">Tree Sitter</a>	<code>&lt;f11&gt; C-t C-t</code>	( <code>pel-treesit-toggle-mode</code> ) Toggle the major mode between the classic mode and the Tree-Sitter based mode. <ul style="list-style-type: none"><li>If the other major mode is not available the command signals a user error.</li></ul> 👉 Use the repeat command (bound to <code>&lt;f5&gt;</code> under PEL) to quickly toggle from the classic to the Tree-Sitter major mode and compare the impact of syntax highlighting.
<b>Show PEL setup information for the major mode.</b>	<code>&lt;f11&gt; ? /</code> <code>&lt;f11&gt; SPC d ?</code> <code>&lt;f12&gt; ?</code>	( <code>pel-mode-setup-info</code> &optional APPEND) ( <code>pel-dart-setup-info</code> &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: <ul style="list-style-type: none"><li><code>pel-use-dart</code> (whether the classic or tree-sitter based major mode is used).</li><li>indentation and hard tab control customizable user-options</li></ul> To append information in the buffer instead of clearing the previous content type any prefix argument (such as <b>C-u</b> ) before the command keystroke.
<b>Comments</b>  See also: <a href="#">Comments</a>	<code>&lt;f11&gt; ; ;</code>	( <code>hide/show-comments-toggle</code> &optional START END)	Toggle hiding/showing of comments in the active region or whole buffer. <ul style="list-style-type: none"><li>If the region is active then toggle in the region. Otherwise, in the whole buffer.</li></ul> 👉 This requires the <a href="#">hide-commt.el</a> package (see <a href="#">Comments</a> ). ✅ PEL activates it when the <code>pel-use-hide-commt</code> user option is <b>t</b> .
<b>Insert, realign, comment/uncomment region</b> See also: <a href="#">Comments With PEL</a> : Comment the current line with <b>M-0 M-;</b>	<b>M- ;</b>	( <code>comment-dwim</code> ARG) ( <code>pel-comment-dwim</code> ARG)	Insert or realign comment on current line (or region if a region is active). <ul style="list-style-type: none"><li>On a single line, the comment is placed <b>after</b> the code.</li><li><b>C-u M- ;</b> executes comment-kill</li></ul> Same as <code>comment-dwim</code> but comments the current line with a numeric ARG or 0.

## Emacs & Dart – References

Document	Notes
<a href="#">The Dart programming language</a>	<ul style="list-style-type: none"> <li><a href="#">Dart @ Wikipedia</a></li> <li><a href="#">Dart home</a></li> <li><a href="#">Dart @ Github</a></li> <li>•</li> </ul>
<a href="#">Dart Documentation</a>	<ul style="list-style-type: none"> <li><a href="#">Introduction to Dart</a></li> <li><a href="#">Effective Dart</a></li> <li><a href="#">Dart Libraries</a></li> <li><a href="#">Dart SDK</a></li> <li><a href="#">Asynchronous programming: futures, async, await</a></li> <li><a href="#">Packages</a></li> </ul>
<a href="#">Dart Projects</a>	<ul style="list-style-type: none"> <li><a href="#">Awesome Dart</a></li> </ul>