






























Emacs Support for Javascript 🚧

Description	Keystroke	Function	Note
Javascript Support <ul style="list-style-type: none">◦ Help & Customization<ul style="list-style-type: none">• Major mode selection• Minor mode activation• Comments◦ Indentation Control• Navigation• Outline/Fold/Show• Compilation/Static Analysis• Javascript REPL• Debugging mode◦ References <div>Last updated on: 2025-10-23</div>			Emacs has built-in support for Javascript via the js-mode and js-ts-mode (when tree-sitter is activated).  PEL supports Javascript when pel-use-js user options is turned on (non-nil), depending on its value: See  Tree Sitter and  Tree-sitter <ul style="list-style-type: none">• t: PEL activates the built-in classic js-mode• with-tree-sitter: PEL activates the Tree-sitter based internal js-ts-mode. Requires: Emacs >= 30, and when pel-use-tree-sitter is set to t.• js2-mode: PEL activates the classic  external js2-mode . It provides a js2-minor-mode that provides js2 features to the other modes.• with-js2-minor: PEL activates the built-in classic js-mode and also activates the js2-minor-mode• with-ts-js2-minor: PEL activates the Tree-sitter based internal js-ts-mode and also activates the js2-minor-mode.  Probably the most useful for most.• js3-mode: PEL activates the  external js3-mode. A package that has not been maintained for some time. PEL also provides: <ul style="list-style-type: none">•  Speedbar support for Javascript files listing functions and types when pel-use-speedbar is on.•  iMenu is supported by all javascript modes•  js-comint support provided when  pel-use-js-comint user-option is turned on (set to t).•  js2-closure support provided when  pel-use-js2-closure user-option is turned on (set to t).•  flow-js2-mode support provided when  pel-use-flow-js2-mode user-option is turned on (set to t).•  xref-js2 support provided when  pel-use-xref-js2 user-option is turned on (set to t).  Not yet integrated seamlessly in PEL.
Open this PDF file. See also:  Help/Info	<f11> SPC i <f1> <f12> <f1>	(pel-help-pdf &optional OPEN-WEB-PAGE)	Open the  Java 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.
 Customize PEL Javascript support	<f11> SPC i <f2> <f12> <f2>	(pel-customize-pel &optional OTHER-WINDOW)	Customize PEL Javascript support: pel-use-javascript <ul style="list-style-type: none">• If OTHER-WINDOW is non-nil (use C-u), display in another window.
 Customize Emacs Javascript support	<f11> SPC i <f3> <f12> <f3>	(pel-customize-library &optional OTHER-WINDOW)	Customize Emacs Java support: js, js2 <ul style="list-style-type: none">• If OTHER-WINDOW is non-nil (use C-u), display in another window.
Environment Help	Use the following command to verify your Javascript environment.		
Show PEL setup for Javascript	<f11> SPC i ? <f12> ?	(pel-js-setup-info &optional APPEND)	Display Javascript setup information inside a *pel-js-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">• pel-use-js (whether the classic or tree-sitter based major mode is used),• activation of minor modes for the 3 set of major modes supported,• the user options controlling indentation and hard tab width rendering. To append information in the buffer instead of clearing the previous content type any prefix argument (such as C-u) before the command keystroke.
Toggle between classic-mode and tree-sitter based mode	PEL supports the ability to switch between a classic and the Tree-Sitter based mode. <ul style="list-style-type: none">• For Javascript, the switch can be done between the js-mode and the js-ts-mode. There is no Tree-Sitter-based mode for the js2-mode or the js3-mode.• However, you can use js-mode and js-ts-mode with the js2-minor-mode which gives you the ability to use A Tree-Sitter based mode with the extra functionality provided by the js2-mode package. The selection of the major mode used by default is controlled by the pel-use-js user-option as described at the top of the page. <ul style="list-style-type: none">• That selection can be changed without the following command without impacting the mode used when you open new JavaScript buffers.		
Toggle between classic and Tree-Sitter major mode 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. <ul style="list-style-type: none">• If the other major mode is not available the command signals a user error.  Use the repeat command (bound to <f5> under PEL) to quickly toggle from the classic to the Tree-Sitter major mode and compare the impact of syntax highlighting.
Minor Mode activation	PEL provides 3 user-options you can use to identify minor modes that are activated when the mode, selected by pel-use-js , starts. These user-options are: <ul style="list-style-type: none">• pel-js-activates-minor-modes : identify the functions that activate minor modes in the js-mode and js-ts-mode buffers.• pel-js3-activates-minor-modes : identify the functions that activate minor modes in the js2-mode buffers (when pel-use-js is set to js2-mode).• pel-js3-activates-minor-modes : identify the functions that activate minor modes in the js3-mode buffers (when pel-use-js is set to js3-mode). It is also possible to manually activate or deactivate the minor mode. The following are minor modes related to the js2-mode functionality. Other minor modes can be quite useful for Javascript editing, such as the smart-dash mode, various modes for completion, search, cross reference, etc. Refer to other PEL PDF for more information on those. For example: <div><div><ul style="list-style-type: none">• Anzu-mode See  Search/Replace• smart-dash-mode See  Text Modes• which-function-mode See  Mode Line</div><div><ul style="list-style-type: none">• flyspell-prog-mode See  Spell Checking</div></div>		
Toggle js2-minor-mode	<f11> SPC i 2 <f12> 2	(js2-minor-mode &optional ARG)	Minor mode for running js2 as a background linter. <ul style="list-style-type: none">• This allows you to use a different major mode for JavaScript editing, such as 'js-mode', while retaining the asynchronous error/warning highlighting features of 'js2-mode'.• This is a minor mode. If called interactively, toggle the 'Js2 minor mode' mode.• If the prefix argument is positive, enable the mode, if it is zero or negative, disable the mode.  Requires external js2-mode . Activated by pel-use-js . See above.
Toggle editing Javascript with flow type annotation.	<f11> SPC i M-f <f12> M-f	(flow-js2-mode &optional ARG)	Toggle minor mode for editing JS files with flow type annotations. <ul style="list-style-type: none">• If called interactively, toggle the 'Flow-Js2 mode' mode.• If the prefix argument is positive, enable the mode, if it is zero or negative, disable the mode.
Toggle Js2 refactor mode	<f11> SPC i M-r <f12> M-r	(js2-refactor-mode &optional ARG)	Toggle minor mode providing JavaScript refactorings. <ul style="list-style-type: none">• If called interactively, toggle the 'Js2-Refactor mode' mode.• If the prefix argument is positive, enable the mode, if it is zero or negative, disable the mode.
Comments	Global comment commands can be used in markdown buffers to comment or un-comment lines and regions. See also:  Comments <ul style="list-style-type: none">• Also see the outline commands provided by js2-mode below; the C-c C-t folds comment blocks.		
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
With PEL: Comment the current line with M-0 M-;		(pel-comment-dwim ARG)	Same as comment-dwim but comments the current line with a numeric ARG or 0.
Toggle display of comments in buffer or active region	<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.

Javascript & Emacs Reference

Javascript Language	<ul style="list-style-type: none">• Javascript @ Wikipedia• ECMAScript home . Has links to the standard.
JSX : Javascript XML	<ul style="list-style-type: none">• JSX @ Wikipedia<ul style="list-style-type: none">• React @ Wikipedia•