













































































🚦 Tree-Sitter parsers for Emacs 🚧

TreeSitter parsers	Supported by PEL	User-option control	tree-sitter mode	Language grammar	With ⓘ iMenu support	With ⓘ Speedbar support	Status of the Tree-Sitter aware major mode	Features working in the tree-sitter mode	Language Server	
<div>Last updated on: 2026-01-17</div> <div>See Also:</div> <ul style="list-style-type: none">📖 Tree SitterUsing tree-sitter with Emacs and PELList of Tree-Sitter parsers	Indicates yes only when explicitly supported by PEL code.	The name and value of PEL user option that control whether Tree-Sitter aware mode is used.	The name of the major mode command that are tree-sitter based. Modes names in black are built-in Emacs. The others have a link	Name and link to the project providing the language grammar. If an entry is required in treesit-load-name-override-list it is identified here.	Whether all commands based on imenu work in tree-sitter mode.	Whether Speedbar support works for the tree-sitter based mode.	Identify any known problem here. Later this will be expanded to several features			
📄 - Ada 🚧 ⚡ Ⓢ	Yes	pel-use-ada	ada-ts-mode	tree-sitter-langs ➡ briot/tree-sitter-ada	Yes	Yes	The ada-ts-mode is a great implementation!	<ul style="list-style-type: none">outline-minor-mode. See ⓘ Outlinesyntax-highlighting: 4 levels: controlled by treesit-font-lock-levelflexible/precise indentation control with several indentation back-ends:<ul style="list-style-type: none">tree-sitter based (the default), very flexible with 9 customizable user-options.LSP back-end (a formatter instead of indentation engine)		
📄 - Algol	No									
📄🍏 - AppleScript				No grammar found yet.						
APL 🚧				No grammar found yet.						
📄 - Arc ⓘ ⓘ				No grammar found yet, but perhaps a scheme syntax could be used.						
📄 - awk ⓘ				Beaglefoot/tree-sitter-awk						
📄 - C Ⓢ	Yes	pel-use-c		tree-sitter-langs ➡ tree-sitter/tree-sitter-c						
C# 🚧 future										
📄 - C++ ⓘ Ⓢ	Yes	pel-use-c++		tree-sitter-langs ➡ tree-sitter/tree-sitter-cpp						
📄 - C3 Ⓢ	Yes	pel-use-c3	c3-ts-mode	github.com/c3lang/tree-sitter-c3	Yes	Yes	Basic: highlighting, iMenu and Speedbar work fine. No support for navigation commands as of December 2025.			
Carbon 🚧 future Ⓢ				No grammar found yet.						
📄 - Chez ⓘ ⓘ				No grammar found yet, but perhaps a scheme syntax could be used.						
📄 - Chibi ⓘ ⓘ				No grammar found yet, but perhaps a scheme syntax could be used.						
📄 - Chicken ⓘ ⓘ				No grammar found yet, but perhaps a scheme syntax could be used.						
📄 - Clojure ⓘ ⓘ				tree-sitter-langs ➡ sogaiu/tree-sitter-clojure						
Common Lisp ⓘ ⓘ				tree-sitter-langs ➡ tree-sitter-grammars/tree-sitter-commonlisp						
Crystal 🚧				crystal-lang-tools/tree-sitter-crystal						

TreeSitter parsers	Supported by PEL	User-option control	tree-sitter mode	Language grammar	With  iMenu support	With  Speedbar support	Status of the Tree-Sitter aware major mode	Features working in the tree-sitter mode	Language Server	
 - D  				tree-sitter-langs ➡ CyberShadow/tree-sitter-d						
 - Dart	Yes	pel-use-dart	dart-ts-mode	tree-sitter-langs ➡ UserNobody14/tree-sitter-dart	Yes, only with dart-ts-mode	Yes, only with dart-ts-mode	OK			
 - Eiffel   				No grammar found yet.						
 - Elm   				tree-sitter-langs ➡ elm-tooling/tree-sitter-elm						
 - Elixir   	Yes	pel-use-elixir	elixir-ts-mode	tree-sitter-langs ➡ elixir-lang/tree-sitter-elixir	Yes	Yes	OK			
  - Emacs Lisp				tree-sitter-langs ➡ Wilfred/tree-sitter-elisp						
 - Erlang  	Yes	pel-use-erlang	erlang-ts-mode	tree-sitter-langs ➡ WhatsApp/tree-sitter-erlang	Yes	Yes	As of this writing, this is an early version. Fontification only works for comments. Maintainers of erlang-ts-mode would appreciate help.			
 - Factor   	No: As of Emacs 30.2 there is no factor-ts-mode			No grammar found yet.			Nothing found yet.			
 - Forth 	No: As of Emacs 30.2 there is no forth-ts-mode			AlexanderBrevig/tree-sitter-forth	Yes	Yes	Nothing found yet.			
Fortran 				tree-sitter-langs ➡ stadelmanma/tree-sitter-fortran						
 - Gambit  				No grammar found yet, but perhaps a scheme syntax could be used.						
 - Gerbil   				No grammar found yet, but perhaps a scheme syntax could be used.						
 - GNU Guile  				No grammar found yet, but perhaps a scheme syntax could be used.						
 - Gleam	Yes	See note ➡	gleam-ts-mode	tree-sitter-langs ➡ tree-sitter-gleam	Yes	Yes	OK	Note: Gleam is only supported by a Tree-Sitter aware mode. There's no classic mode for Gleam.		
 - Go 	Yes	pel-use-go	go-ts-mode	tree-sitter-langs ➡ tree-sitter-go	Yes	Yes	OK			
 - Go go.mod	Yes	pel-use-go	go-mod-ts-mode	tree-sitter-go-mod	Yes	Yes	OK			
Groovy 				tree-sitter-langs ➡ Decodetalkers/tree-sitter-groovy						
 - Haskell 				tree-sitter-langs ➡ tree-sitter/tree-sitter-haskell						
Haxe 				tree-sitter-langs ➡ vantroeseba/tree-sitter-haxe						
 - Hy (python) 				No grammar found yet.						
 - Janet   				<ul style="list-style-type: none">tree-sitter-langs ➡ sogaiu/tree-sitter-janet-simple ,GrayJack/tree-sitter-janet						
 - Java 				tree-sitter-langs ➡ tree-sitter/tree-sitter-java						

TreeSitter parsers	Supported by PEL	User-option control	tree-sitter mode	Language grammar	With  iMenu support	With  Speedbar support	Status of the Tree-Sitter aware major mode	Features working in the tree-sitter mode	Language Server	
 - Javascript	Yes	pel-use-js	js-ts-mode	tree-sitter-langs ➡ tree-sitter/tree-sitter-javascript	Yes	Yes	OK			
 - Julia				tree-sitter-langs ➡ tree-sitter/tree-sitter-julia						
 Kotlin				tree-sitter-langs ➡ fwcd/tree-sitter-kotlin						
 - LFE				No grammar found yet.						
 - Lua	Yes	pel-use-lua	lua-ts-mode	tree-sitter-grammars/tree-sitter-lua	Yes	Yes	OK			
				tree-sitter-langs ➡ MunifTanjim/tree-sitter-lua ⚠	Yes	Yes	<ul style="list-style-type: none">fortification does not workThe tree-sitter-lua project used by tree-sitter-langs seems unmaintained.			
 - Modula	No: As of Emacs 30.2 there is no modula-ts-mode			No grammar found yet.						
 - NetRexx	No: As of Emacs 30.2 there is no netrexx-ts-mode			No grammar found yet.			Nothing found yet.			
 - Nim	No: As of Emacs 30.2 there is no nim-ts-mode implemented yet.			alviss/tree-sitter-nim	No	Yes, but since iMenu is not supported, nothing shows.	Nothing found yet.			
 - Objective-C	No: As of Emacs 30.2 there is no known objc-ts-mode implemented yet.			<ul style="list-style-type: none">tree-sitter-grammars/tree-sitter-objcmerico-dev/tree-sitter-objc						
 - OCaml	No: there seems to be several tree-sitter aware major modes for OCaml but PEL does not yet support any as there does not seem to have a clear winner.			tree-sitter-langs ➡ tree-sitter/tree-sitter-ocaml			<div>There seems to have several incomplete implementations:</div> <ul style="list-style-type: none">terrateamio/ocaml-ts-modedmitrig/ocaml-ts-mode			
 - Odin			Sampie159/odin-ts-mode	tree-sitter-grammars/tree-sitter-odin					DanielGavin/ols	
 - Pascal	No: As of Emacs 30.2 there is no pascal-ts-mode			tree-sitter-langs ➡ isopod/tree-sitter-pascal			Nothing found yet.			
 - Perl	No: As of Emacs 30.2 there is no perl-ts-mode implementation that has reached good enough stability.			tree-sitter-langs ➡ tree-sitter-perl/tree-sitter-perl	Yes	Yes	<div>Under development, not yet ready:</div> <ul style="list-style-type: none">sourcehut/pranshu/perl-ts-modeHaraldJoerg/emacs-perl-ts-mode			
 - Pike	No: As of Emacs 30.2 there is no pike-ts-mode			No grammar found yet.			Nothing found yet.			
 - Python				tree-sitter-langs ➡ tree-sitter/tree-sitter-python						
 - Purescript				tree-sitter-langs ➡ postsolar/tree-sitter-purescript						
 - R				tree-sitter-langs ➡ r-lib/tree-sitter-r						
 - Racket				tree-sitter-langs ➡ 6cdh/tree-sitter-racket						
 - ReasonML				reasonml-editor/tree-sitter-reason			Nothing found yet.			

TreeSitter parsers	Supported by PEL	User-option control	tree-sitter mode	Language grammar	With  iMenu support	With  Speedbar support	Status of the Tree-Sitter aware major mode	Features working in the tree-sitter mode	Language Server	
 - REXX	No: As of Emacs 30.2 there is no rexx-ts-mode			No grammar found yet.			Nothing found yet.			
 - Ruby	Yes	pel-use-ruby	ruby-ts-mode	tree-sitter-langs ➡ tree-sitter/tree-sitter-ruby	Yes	Yes	OK			
 - Rust 	Yes	pel-use-rust	rust-ts-mode	tree-sitter-langs ➡ tree-sitter/tree-sitter-rust	Yes	Yes	OK			
Scala 				tree-sitter-langs ➡ tree-sitter/tree-sitter-scala						
 - Scheme  				tree-sitter-langs ➡ 6cdh/tree-sitter-scheme						
 -Seed7    	No: As of Emacs 30.2 there is no seed7-ts-mode implemented yet.			No grammar found yet.	Yes, for seed7-mode	Yes, for seed7-mode	Nothing found yet.			
 -Smalltalk  				No grammar found yet.						
 -Swift				tree-sitter-langs ➡ alex-pinkus/tree-sitter-swift						
 - Tcl  	No: As of Emacs 30.2 there is tcl-ts-mode implemented yet, even though the Tree-Siter grammar exists.			tree-sitter-langs ➡ tree-sitter/tree-sitter-tcl	Yes, for tcl-mode	Yes, for tcl-mode	Nothing found yet.			
 - Typescript 				tree-sitter-langs ➡ tree-sitter/tree-sitter-typescript						
 - UNIX Shell										
 - V	No: As of Emacs 30.2 there is v-ts-mode implemented yet, even though the Tree-Siter grammar exists.			No grammar found yet.			Nothing complete found yet. There is nedpals/tree-sitter-v but that does not seemed maintained.			
 - Verilog				tree-sitter-langs ➡ tree-sitter/tree-sitter-verilog						
 - VHDL				tree-sitter-langs ➡ alemuller/tree-sitter-vhdl						
 -Zig 	Yes	pel-use-zig	zig-ts-mode	tree-sitter-langs ➡ tree-sitter-grammars/tree-sitter-zig	Yes	Yes	With language grammar of 2025-10-13: • fortification does not work • incomplete indentation control • no format on save like zig-mode			
				maxxino/tree-sitter-zig	Yes	Yes	OK			