

































































🚦 Tree-Sitter parsers for Emacs 🚧🚧🚧

TreeSitter parsers	Supported by PEL	tree-sitter mode	With  iMenu support	With  Speedbar support								
Last updated on: 2025-10-14 See Also:  Tree Sitter	Indicates yes only when explicitly supported by PEL code.	The name of the major mode command that supports the tree-sitter based control.	Whether all commands based on imenu work in tree-sitter mode.	Whether Speedbar support works for the tree-sitter based mode.	 As PEL introduces explicit support for more major mode, new class will be filled. Once enough tree-sitter support is explicitly implemented, I will add explicit support for LSP and then check the support of various features like completion, navigation based on LSP and tree-sitter. I will then add more columns related to these features here and in the  Language Servers table.							
 - Ada   												
 - AppleScript												
APL 												
 - Arc  												
 - awk 												
 - C 												
 - C++  												
Carbon  future 												
 - Chez  												
 - Chibi  												
 - Chicken  												
 - Clojure  												
Common Lisp  												
Crystal 												
 - D   												
Dart 												
 - Eiffel   												
 - Elm  												
 - Elixir    												
  - Emacs Lisp												
 - Erlang   												
 - Factor    												
 - Forth 												
Fortran 												
 - Gambit  												
 - Gerbil   												
 - GNU Guile  												
 - Gleam	Yes	gleam-ts-mode	Yes	Yes	Note: Gleam is only supported by a Tree-Sitter aware mode. There's no classic mode for Gleam.							
 - Go 	Yes	go-ts-mode	Yes	Yes								
 - Go go.mod	Yes	go-mod-ts-mode	Yes	Yes								

TreeSitter parsers	Supported by PEL	tree-sitter mode	With  iMenu support	With  Speedbar support							
Groovy 											
<u>PL - Haskell</u> 											
Haxe 											
<u>PL - Hy</u> <i>(python)</i> 											
<u>PL - Janet</u>   											
Java 											
<u>PL - Javascript</u> 											
<u>PL - Julia</u> 											
Kotlin 											
<u>PL - LFE</u>    											
<u>PL - Lua</u>   											
<u>PL-Modula</u>											
<u>PL - NetRexx</u>											
<u>PL - Nim</u>  											
<u>PL-Objective-C</u> 											
<u>PL - OCaml</u>  											
<u>PL - Odin</u> 											
<u>PL-Pascal</u>											
<u>PL - Perl</u> <i>(perl5)</i>											
<u>PL - Pike</u> <i>α</i>  											
<u>PL - Python</u> <i>α</i>   											
<u>PL - Purescript</u>  											
R     											
<u>PL - Racket</u>  											
<u>PL - ReasonML</u> 											
<u>PL - REXX</u>											
<u>PL - Ruby</u>											
<u>PL - Rust</u> 	Yes	rust-ts-mode	Yes	Yes							
Scala 											
<u>PL - Scheme</u>  											
<u>PL-Seed7</u>    											
<u>PL-Smalltalk</u>  											
<u>PL-Swift</u>											
<u>PL - Tcl</u>  											
<u>PL - Typescript</u> 											

<u>TreeSitter parsers</u>	Supported by PEL	tree-sitter mode	With  iMenu support	With  Speedbar support								
 - UNIX Shell												
 - V												
 -Zig 