↘ Topic Index

🚦 Tree-Sitter parsers for Emacs 🚧🚧

TreeSitter parsers	Supported by PEL	User-option control	tree-sitter mode	Language grammar	With Σ iMenu support	With Σ Speedbar support	Status					
Last updated on: 2025-10-15 See Also: Σ Tree Sitter	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 supports the tree-sitter based control. Modes names in black are built-in Emacs.	Name and link to the project providing the language grammar.	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	🚧 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.				
𝔓ℓ - Ada 🚧 ↘Ⓢ												
𝔓ℓ - AppleScript												
APL 🚧												
𝔓ℓ - Arc ⓕⓜ												
𝔓ℓ - awk ⓓ												
𝔓ℓ - C Ⓢ												
𝔓ℓ - C++ ⓄⓈ												
Carbon 🚧 future Ⓢ												
𝔓ℓ - Chez ⓕⓜ												
𝔓ℓ - Chibi ⓕⓜ												
𝔓ℓ - Chicken ⓕⓜ												
𝔓ℓ - Clojure ⓕⓜ												
Common Lisp ⓕⓜ												
Crystal 🚧												
𝔓ℓ - D ⓘⓕⒶ												
Dart 🚧												
𝔓ℓ - Eiffel 🚧 ⓄⓈ												
𝔓ℓ - Elm 🚧 Ⓕ												
𝔓ℓ - Elixir ⓒⓜⓕⒶ	Yes	pel-use-elixir	elixir-ts-mode	tree-sitter-langs ➡ tree-sitter-elixir	Yes	Yes	OK					
𝔈𝔓ℓ - Emacs Lisp												
𝔓ℓ - Erlang ⓒⓕⒶ												
𝔓ℓ - Factor ⓀⓕⓄⓜ												
𝔓ℓ - Forth Ⓚ												
Fortran 🚧												
𝔓ℓ - Gambit ⓕⓜ												
𝔓ℓ - Gerbil ⓕⓜⒶ												
𝔓ℓ - GNU Guile ⓕⓜ												
𝔓ℓ - 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.				

TreeSitter parsers	Supported by PEL	User-option control	tree-sitter mode	Language grammar	With ∑iMenu support	With ∑ Speedbar support	Status					
𝔓𝔈 - Go Ⓢ	Yes	pel-use-go	go-ts-mode	tree-sitter-langs ➡ tree-sitter-go	Yes	Yes	OK					
𝔓𝔈 - Go go.mod	Yes	pel-use-go	go-mod-ts-mode	tree-sitter-go-mod	Yes	Yes	OK					
Groovy												
𝔓𝔈 - Haskell Ⓕ												
Haxe												
𝔓𝔈 - Hy (python) ⓜ												
𝔓𝔈 - Janet ⓘⓕⓜ												
Java												
𝔓𝔈 - Javascript												
𝔓𝔈 - Julia ⓜ												
Kotlin												
𝔓𝔈 - LFE ⓒⓜⓕⒶ												
𝔓𝔈 -Lua ⓕ ⓞ ⓟ	Yes	pel-use-lua	lua-ts-mode	tree-sitter-langs ➡ tree-sitter-lua ⚠️	Yes	Yes	• fortification does not work • The tree-sitter-lua project used by tree-sitter-langs seems unmaintained. It should probably use tree-sitter-grammars/tree-sitter-lua					
𝔓𝔈-Modula												
𝔓𝔈 - NetRexx												
𝔓𝔈 - Nim ⓜⓈ												
𝔓𝔈-Objective-C												
𝔓𝔈 - OCaml ⓘⓕ												
𝔓𝔈 - Odin Ⓢ												
𝔓𝔈-Pascal												
𝔓𝔈 - Perl (perl5)												
𝔓𝔈 - Pike 𝒹 ⓘ ⓞ												
𝔓𝔈 - *Python* 𝒹ⓟⓞⓕ												
𝔓𝔈 - Purescript Ⓕ												
R ⓞ ⓟ ⓕ ⓧ |												
𝔓𝔈 - Racket ⓕⓜ												
𝔓𝔈 - ReasonML												
𝔓𝔈 - REXX												
𝔓𝔈 - Ruby	Yes	pel-use-ruby	ruby-ts-mode	tree-sitter-langs ➡ tree-sitter/tree-sitter-ruby	Yes	Yes	OK					
𝔓𝔈 - Rust Ⓢ	Yes	pel-use-rust	rust-ts-mode	tree-sitter-langs ➡ tree-sitter-rust	Yes	Yes	OK					

TreeSitter parsers	Supported by PEL	User-option control	tree-sitter mode	Language grammar	With 𝛴iMenu support	With 𝛴 Speedbar support	Status					
Scala												
𝔓𝔈 - Scheme ⓕⓜ												
𝔓𝔈-Seed7 ⓔⓖ												
𝔓𝔈-Smalltalk ⓞ												
𝔓𝔈-Swift												
𝔓𝔈 - Tcl ⓕⓘ												
𝔓𝔈 - Typescript												
𝔓𝔈 - UNIX Shell												
𝔓𝔈 - V												
𝔓𝔈 -Zig Ⓢ	Yes	pel-use-zig	zig-ts-mode	tree-sitter-langs ➡ tree-sitter-zig	Yes	Yes	• fortification does not work • incomplete indentation control • no format on save like zig-mode					