PEL Topics Index

Last updated on: 2025-03-30 — Note: with PEL, type `<f11> <f1>` to open this PDF index.

Emacs Reference Cards

☝With PEL, access these cards from Emacs with the `<f11> ? e r` key sequence.
See Σ Help/Info for more info.

These are links to the PDF version of official English version of the quick reference cards for GNU Emacs and popular external packages. PEL documents Emacs key bindings as well, these cards provide useful complement to what PEL provides.

Emacs	Calc	Gnus	Magit Cheatsheet	Org	Viper
Emacs survival card	Dired	Gnus booklet	Magit Ref-card		VIP

➢ PEL Overview

- PEL repo
- PEL Readme
- PEL Manual
- PEL NEWS
- Discussions

PEL license

Last updated on: 2025-03-30

Emacs Mailing Lists

This table holds links to the PEL file tables (hosted on Github as raw PDF files).

☝For the best user experience, use a browser that can render PDF directly instead of downloading.
- Mozilla Firefox (version > 78) does that perfectly. You may need to activate a plug-in for other browsers.
- With that in place, you can browse through all the PDFs and reach a vast amount of information quickly.

☝From within Emacs open this topic index PDF by typing the `<f11> ? <f1>` key sequence. More help topics with `<f11> ? p` keys.

☝The symbols, **colour coding** and various other conventions are described in the ➢Legend PDF.

Terminal Multiplexers: GNU screen , Tmux
Command Line Scripting Languages: bash, sh, zsh
Cmdline: GNU readline, ls -l

General Info ➢	➢Legend	➢Recommended Emacs User Option		➢Themes	Migrate from CRiSP
Startup ➢		Run Emacs daemon & clients		iMenu/Speedbar support	
PEL Code ➢	How to do it with PEL	PEL Naming Conventions		PEL Environment Variables	PEL utilities

OS Desktop Key Bindings

(Bindings that don't clash with PEL)

macOS Fct Keys	macOS Keys	Mint 20 Desktop Keys		Ubuntu 16.04 Desktop Keys
	terminal settings	Rocky Linux 8 Desktop Keys		

Feature Comparisons

Completion Modes Compatibility | Speedbar/iMenu Mode Compatibility | Shells/Terminals Comparisons

Key Prefixes & Suffixes

Σ Modifier Keys | Σ Numkeypad | Keys - Fn | Keys - F11 | Keys - F12 | ➢PEL

Σ Emacs Features

- A Guided Tour of Emacs.
- Awesome-Emacs
- MELPA and GNU ELPA

The tables listed at right describe Emacs commands & key bindings for concepts & features. The cell is light-blue for major mode, light-red for minor mode specific concepts. Grey cells are links into other pages for important concepts.
Emacs commands can be executed by name or bound to key sequences. They describe the commands, their arguments and the key sequences bound to them.

- Emacs Keys
- Numeric Arguments

You can also:
- Run Command by Name

Emacs uses a concept of modes:
- Emacs Major and Minor Modes
  - Major Modes
  - Minor Modes
  - Choosing Modes

PEL provides several key sequences to toggle minor modes.

Cells link titles starting with only Σ are Emacs generic features, blue links are external packages. The green links are mostly PEL extensions.

Σ Abbreviations	Σ Diff & Merge	Σ Grep	Σ Marking	Σ Scrolling	Σ Tab Bar
Σ Align	Σ Dired	Σ Help/Info	Σ Menus	Σ Search/Replace	T Templates
Σ Auto-Completion	Σ Display - Lines	Σ Hide/Show	Σ Mode Line	Σ Sessions	Σ Text Modes
Σ Autosave/Backup	Σ Drawing	Σ Highlight (colors)	Σ Mouse	Σ start Shells/REPLs	Σ Time Tracking
Σ Bookmarks	Σ Enriched Text	Σ ibuffer-mode	Σ Narrowing	Σ shell-mode	Σ Tramp
Σ Buffers	Σ Execute Cmds	Σ Indentation	Σ Navigation	Σ term-mode	Σ Transpose text
Σ Case Conversions	Σ Exec Shell Cmds	Σ Input Method	Σ Object Files	Σ eat-mode	Σ𝑥 Treemacs
Σ Close/Suspend	Σ Faces/Fonts	Σ Inserting Text	Σ Outline	Σ vterm-mode	Σ Undo/Redo/Repeat
Σ Comments	ΣP Fast Startup	Σ Key-Chords	Σ Packages	Σ𝑥 Smartparens	Σ VCS-Git 𝑥Magit
Σ Completion/Input	Σ File Encoding	Σ Keyboard Macros	Σ𝑥 Projectile	Σ Sorting	Σ VCS-Mercurial
Σ Counting	Σ File-mngt	𝔓ℓ𝑥- Lispy	Σ Rectangles	Σ Speedbar	Σ VCS-Subversion
ΣM CUA	Σ File/Dir Variables		Σ Registers	Σ Spell Checking	Σ Web
Σ Cursor	Σ Fill/Justify			Σ SyntaxCheck	Σ Whitespace
Σ Customize	Σ Frames				Σ Windows
Σ Cut & Paste					Σ Xref - Cross Refs
𝑥𝔓ℓ - Emacs Lisp concepts & tools	𝑥 display-buffer	𝑥※ - ELisp Types	𝑥 ERT (regr-testing)	𝑥 Hooks	

XRef - Cross Reference Tools

See also: Σ Xref

Emacs supports various cross reference mechanisms described in the Σ Xref table. These mechanisms take advantage of various external tools and integrate with them. Notes about those tools are available in the tables listed in this section. Also describes indentation.

Xref-Support	Xref-Frontend	Xref-Backend			Indentation Styles

PEL supports installation and partial setup of the following tools:

Build Tools & Preprocessor

PEL has support for several build tools but they are not all documented in a page.
- Nix — Requires nix-mode external package — activated when pel-use-nix-mode user-option is tuned on.
- Tup — Requires tup-mode external package — activated when pel-use-tup user-option is tuned on.

𝔓ℓ - CMake	𝔓ℓ - M4	𝔓ℓ - Make gmake			
Ⓓ CWL	Ⓓ YAML				
Ⓢ ASN.1 asn1-mode	Ⓢ MIB snmp-mode	Ⓢ YANG			
Config files	RFC (RFC @ Wikipedia)	RPM Files (spec file format)		Ⓜ X.509 Certificates	
Verilog	VHDL				
Ⓜ AsciiDoc	Ⓜ Markdown	Ⓜ Org-Mode	Ⓜ reStructuredText		OS App Control Scripting Languages
Ⓜ Graphviz Dot	Ⓜ MscGen	Ⓜ PlantUML			𝔓ℓ - AppleScript

Data Serialization

Data Modelling/ Specification

Other File Formats

Hardware Description Languages

Lightweight Markup Languages

- Graphics Markup

Programming Languages
Main Paradigm of Programming Languages
- *Actor Model:* Ⓐ *Concatenative* Ⓚ
- *Concurrent:* ⓒ *Domain Specific* ⓓ
- *Dynamic* 𝒹 *Extensible* ⓔ
- *Functional:* ⓕ *Pure:* Ⓕ
- *Generic* ⓖ
- *Imperative:* ⓘ *or no token*
- *Object Oriented* ⓞ *Procedural* Ⓟ
- *Has Syntactic Macros:* ⓜ
- *Multi-paradigm* *Reflective* |
- *System Level* Ⓢ

- The programming languages supported by PEL are listed here in alphabetical order.
- Emacs (and PEL) also provides basic support for some of the one PEL does not support and for other programming languages not listed here.

Future support for Crystal, Elm, Kotlin, Lua, Purescript, ReasonML, Seed7, Typescript, Zig and documentation of support for Ada, Fortran, Javascript, Java, Modula, Pascal (based on my need for them or requests).

Emacs has major mode support for several programming languages. PEL extends Emacs support for some of them (others are marked).

BEAM Programming Languages	*Functional Languages*	Javascript target	Pascal-style syntax	Lisp-like Languages	Stack Based Languages
Curly Bracket Languages	Java Virtual Machine Languages	ML Family Languages	Lisp Family Languages	Scheme Language Dialects	

Cell colours identifies the programming language family(ies).

Ada	𝔓ℓ - D ⓘⓕⒶ	𝔓ℓ - Gambit ⓕⓜ	𝔓ℓ - Janet ⓘⓕⓜ	Objective-C	𝔓ℓ - Ruby
𝔓ℓ - Arc ⓕⓜ	Dart	𝔓ℓ - Gerbil ⓕⓜⒶ	Java	𝔓ℓ - OCaml ⓘⓕ	𝔓ℓ - Rust Ⓢ
𝔓ℓ - awk ⓓ	Eiffel Ⓢ	𝔓ℓ - GNU Guile ⓕⓜ	𝔓ℓ - Javascript	𝔓ℓ - Odin	Scala
𝔓ℓ - C Ⓢ	𝔓ℓ - Elm Ⓕ	𝔓ℓ - Gleam	𝔓ℓ - Julia ⓜ	𝔓ℓ-Pascal	𝔓ℓ - Scheme ⓕⓜ
𝔓ℓ - C++ ⓞⓈ	𝔓ℓ - Elixir ⓒⓜⓕⒶ	𝔓ℓ - Go Ⓢ	Kotlin	𝔓ℓ - Perl (perl5)	𝔓ℓ-Seed7 ⓔⓖ
𝔓ℓ - Chez ⓕⓜ	𝑥𝔓ℓ - Emacs Lisp	Groovy	𝔓ℓ - LFE ⓒⓜⓕⒶ	𝔓ℓ - Pike 𝒹ⓘⓞ	𝔓ℓ-Swift
𝔓ℓ - Chibi ⓕⓜ	𝔓ℓ - Erlang ⓒⓕⒶ	𝔓ℓ - Haskell Ⓕ	𝔓ℓ -Lua ⓕⓞⓟ	𝔓ℓ - *Python* 𝒹ⓟⓞⓕ	𝔓ℓ - Tcl ⓕⓘ
𝔓ℓ - Chicken ⓕⓜ	Factor Ⓚⓕⓞⓜ	Haxe	Modula	𝔓ℓ - Purescript Ⓕ	𝔓ℓ - Typescript
𝔓ℓ - Clojure ⓕⓜ	𝔓ℓ - Forth Ⓚ	𝔓ℓ - Hy (python) ⓜ	𝔓ℓ - NetRexx	𝔓ℓ - Racket ⓕⓜ	𝔓ℓ - UNIX Shell
Common Lisp ⓕⓜ	Fortran		𝔓ℓ - Nim ⓜⓈ	𝔓ℓ - ReasonML	𝔓ℓ - V
Crystal				𝔓ℓ - REXX	𝔓ℓ -Zig Ⓢ