PEL Topics Index

Note: with PEL, type `<f11> <f1>` to open this PDF index.

Emacs Reference Cards

☝ With PEL you can access these via the `<f11> ? e r` key sequence.
See Σ Help/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 (license)

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

This table holds links to the PEL file tables. Each cell holds a hyperlink to the GitHub hosted raw PDF table.

☝ 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.

- General Information.
- Development Information
- Migration Guide

➢Legend	➢Recommended Emacs User Option		➢Themes		
➢PEL	iMenu/Speedbar support		PEL Naming Conventions		
➢CRiSP ∽ Emacs					

OS Desktop Key Bindings

(Bindings that don't clash with PEL)

macOS Fct Keys	macOS Keys	Ubuntu 16.04 Desktop Keys			
	terminal settings	Mint 20 Desktop Keys			

Feature Comparisons

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

Key Prefixes & Suffixes

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

Σ Emacs Features

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

☝ Run Emacs daemon & client on macOS

The PEL tables listed at right describe Emacs commands & key bindings for concepts & features. The cell color is light-blue for major mode, light-red for minor mode
Emacs commands can be executed by name or bound to key sequences. The commands may have ***arguments*** and keys can express 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 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 Transpose text
Σ Buffers	Σ Faces/Fonts	Σ Indentation	Σ Navigation	Σ term-mode	Σ𝑥 Treemacs
Σ Case Conversions	ΣP Fast Startup	Σ Input Method	Σ Object Files	Σ eat-mode	Σ Undo/Redo
Σ Close/Suspend	Σ File Encoding	Σ Inserting Text	Σ Outline	Σ vterm-mode	Σ VCS-Git 𝑥Magit
Σ Comments	Σ File-mngt	Σ Key-Chords	Σ Packages	Σ𝑥 Smartparens	Σ VCS-Mercurial
Σ Completion/Input	Σ File/Dir Variables	Σ Keyboard Macros	Σ𝑥 Projectile	Σ Sorting	Σ VCS-Subversion
Σ Counting	Σ Fill/Justify	𝔓ℓ𝑥- Lispy	Σ Rectangles	Σ Speedbar	Σ Web
ΣM CUA	Σ Frames		Σ Registers	Σ Spell Checking	Σ Whitespace
Σ Cursor				Σ SyntaxCheck	Σ Windows
Σ Customize					Σ Xref - Cross Refs
Σ Cut & Paste					

Σ𝔓ℓ - Emacs Lisp concepts & tools	ℓ display-buffer	ℓ※ - ELisp Types	ℓ ERT (regr-testing)	ℓ 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.

Xref-Support	Xref-Frontend	Xref-Backend			

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 - M4	𝔓ℓ - Make gmake			ls -l	

Command Line Scripting Languages:
bash, sh, zsh

Data Serialization	Ⓓ CWL	Ⓓ YAML			Utility: GNU readline
Data Modelling/ Specification	Ⓢ ASN.1 asn1-mode	Ⓢ MIB snmp-mode	Ⓢ YANG		
Hardware Description Languages	Verilog future	VHDL future			
Text Markup Languages	M AsciiDoc	M Markdown	M Org-Mode	M reStructuredText	
• **Graphics Markup**	M Graphviz Dot	M MscGen	M PlantUML		

OS App Control Scripting Languages
𝔓ℓ - AppleScript

Programming Languages
Main Paradigm of Programming Language Families
- ***Actor Model:*** Ⓐ
- ***Concatenative*** Ⓚ
- ***Concurrent:*** ⓒ
- ***Functional:*** ⓕ ***Pure:*** Ⓕ
- ***Imperative:*** ⓘ ***or no token***
- ***Object Oriented*** ∞
- ***Has Syntactic Macros:*** ⓜ

- The programming languages supported by PEL are listed here in alphabetical order.
- Emacs (and PEL) also provides basic support 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 (if any)).

Emacs has major mode support for several programming languages. PEL currently adds extra support for some of them, listed below.

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

The following lists the programming languages in alphabetical order.
- The cell colours give a coarse indication of the programming language family(ies).

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