PEL Topics Index

Last updated on: 2024-11-20 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 (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

General Info. ➢	➢Legend	➢Recommended Emacs User Option	➢Themes	Migrate from CRiSP
Startup. ➢		Run Emacs daemon & clients	iMenu/Speedbar support	
Development Info. ➢		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 | ➢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
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.

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 - CMake	𝔓𝔩 - M4	𝔓𝔩 - Make gmake			Command Line Scripting Languages: bash, sh, zsh
Ⓓ CWL	Ⓓ YAML				Utility: GNU readline
Ⓢ ASN.1 asn1-mode	Ⓢ MIB snmp-mode	Ⓢ YANG			ls -l
	RPM Files	Ⓜ X.509 Certificates			
Verilog	VHDL				
Ⓜ AsciiDoc	Ⓜ Markdown	Ⓜ Org-Mode	Ⓜ reStructuredText		OS App Control Scripting Languages
Ⓜ Graphviz Dot	Ⓜ MscGen	Ⓜ PlantUML			𝔓𝔩 - AppleScript

Programming Languages

Main Paradigm of Programming Language Families
- *Actor Model:* Ⓐ
- *Concatenative* Ⓚ
- *Concurrent:* ⓒ
- *Domain Specific* ⓓ
- *Dynamic* 𝒹
- *Functional:* ⓕ *Pure:* Ⓕ
- *Imperative:* ⓘ *or no token*
- *Object Oriented* ⓞ
- *Procedural* Ⓟ
- *Has Syntactic Macros:* ⓜ
- *System Level* Ⓢ

- 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).

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	Lisp Family Languages	Lisp-like Languages	
Curly Bracket Languages	Java Virtual Machine Languages	ML Family Languages	Scheme Language Dialects	Stack Based Languages	

Cell colours identifies the programming language family(ies).

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