PEL Topics Index

Last updated on: 2025-09-19 Note: with PEL; type <f11> <f1> to open this PDF index.

Emacs Reference Cards

Links to PDF version of official English version of the quick reference cards for GNU Emacs and popular external packages.
☝With PEL, access these PDF cards from within Emacs with the <f11> ? e r key sequence. See Σ Help/Info for more info.

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
- Emacs Mailing Lists
- Emacs project repo
- Contribute to Emacs

This table holds links to all other PEL topic oriented PDF table files (hosted on Github).
☝For the best user experience, use a browser that can render PDF directly instead of downloading: all PDFs are heavily hyperlinked.
- Mozilla Firefox (version > 78) does that perfectly. You may need to activate a plug-in for other browsers.

☝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
: GNU readline, ls -l, ssh

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

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

Emacs Manual , Guided Tour of Emacs.
- Mastering 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.

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

- Emacs Lisp concepts & tools	display-buffer	- ELisp Types	ERT (regr-testing)	Hooks	
Other tools extending Emacs functionalities — Parsing tools:	Language Servers	Tree-sitter			
Xref Tools:	Xref-Support	Xref-Frontend	Xref-Backend		Indentation Styles

Build Tools & Preprocessor

- CMake	- M4	- Make gmake	- Nix	- Tup

Data Serialization & Modelling

CWL	YAML		ASN.1 asn1-mode	MIB snmp-mode	YANG

Other File Formats

Changelog Files	Config/ini/toml... Files	RFC (RFC @ Wikipedia)	RPM Files (spec file format)	SSH files ssh

Hardware Description Languages

- Verilog	- VHDL	Language Server & Tools for HDL		X.509 Certificates

Lightweight Markup Languages

AsciiDoc	Markdown	Org-Mode	reStructuredText	

Graphics Markup

Graphviz Dot	MscGen	PlantUML		

Programming Languages

Main Paradigm of Programming Languages
- *Actor Model:* Ⓐ *Array* Ⓧ
- *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 APL, Carbon, Crystal, Dart, Elm, Groovy, Haxe, Kotlin, Purescript, ReasonML, Scala, Typescript and documentation of support for Fortran, Javascript, Java, Modula, (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	*Functional*	Javascript target	Pascal-style syntax	Lisp-like Languages	Stack Based
Curly Bracket	Java Virtual Machine	ML Family	Lisp Family	Scheme Dialects	OS App Control

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