PEL Quick Access Topics Index

Last updated on: 2025-11-18 Note: with PEL; type <f11> <f1> to open this PDF index.

GNU Emacs Reference Cards

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

- Emacs Release History
- EmacsWiki
- Emacs project repo

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

PEL

- Repo
- Manual
- Discussions

Readme License NEWS

- Emacs Mailing Lists
- Contribute to Emacs

This table holds links to all other PEL topic oriented PDF table files (hosted on Github).
☝ For best user experience, use a browser like Firefox that can render PDF directly instead of downloading: all PDFs are heavily hyperlinked.
☝ 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

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

Σ Emacs Features

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

Σ Emacs Manual , Guided Tour of Emacs , Emacs Lisp Manual
- Emacs Docs: Emacs, Emacs Lisp
- 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.

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

XPEL - Emacs Lisp concepts	& tools	X display-buffer	X EIEIO - ELisp Types	X Hooks	X Elisp Build Tools	X ERT (regr-testing)	
Parsing tools, Indentation &	Σ Xref Tools:	Language Servers	Tree-sitter	Indentation Styles	Xref-Support	Xref-Frontend	Xref-Backend

Build Tools

PEL - CMake | PEL - Make gmake | PEL - Meson | PEL - Ninja | PEL - Nix | PEL - Tup

Data Serialization & Configuration

Ⓓ CWL | Ⓓ JSON | Ⓓ PKL | Ⓓ XML | Ⓓ YAML

Modelling

Ⓜ ASN.1 asn1-mode | Ⓜ MIB snmp-mode | Ⓜ YANG

Other File Formats

Binary, Object, Executable Files		Log Files	RFC (RFC @ Wikipedia)	SSH files ssh
Σ Changelog Files	Config/ini/toml... Files		RPM Files (spec file format)	Ⓜ X.509 Certificates

Hardware Description Languages

HDL - Verilog | HDL - VHDL | Language Server & Tools for HDL

Lightweight Markup Languages

Ⓜ AsciiDoc | Ⓜ Markdown | Ⓜ Org-Mode | Ⓜ reStructuredText

- Graphics Markup

Ⓜ Graphviz Dot | Ⓜ MscGen | Ⓜ PlantUML

Programming Languages Major Modes

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

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* Ⓢ

PEL - Ada ↝Ⓢ	PEL - D ⓘⓕⒶ	PEL - Gambit ⓕⓜ	PEL - Janet ⓘⓕⓜ	PEL-Pascal	Scala
PEL - AppleScript	PEL - Dart ↝ⓕⓞ|	PEL - Gerbil ⓕⓜⒶ	PEL - Java	PEL - Perl (perl5)	PEL - Scheme ⓕⓜ
APL	PEL - Eiffel ⓞⓈ	PEL - GNU Guile ⓕⓜ	PEL - Javascript	PEL - Pike 𝒹ⓘⓞ	PEL-Seed7 ⓔⓖ↝
PEL - Arc ⓕⓜ	PEL - Elm Ⓕ	PEL - Gleam	PEL - Julia ⓜ	Pony	PEL-Smalltalk ⓞ
PEL - awk ⓓ	PEL - Elixir ⓒⓜⓕⒶ	PEL - Go Ⓢ	Kotlin	PEL - Python 𝒹ⓅⓄⓕ	PEL-Swift
PEL - C Ⓢ	XPEL - Emacs Lisp	Groovy	PEL - LFE ⓒⓜⓕⒶ	PEL - Purescript Ⓕ	PEL - Tcl ⓕⓘ
PEL - C++ ⓄⓈ	PEL - Erlang ⓒⓕⒶ	PEL - Haskell Ⓕ	PEL -Lua ⓕⓞⓅ	R ⓞⓅⓕⓧ|	PEL - Typescript
Carbon future Ⓢ	PEL - Factor ⓀⓕⓄⓜ	Haxe	PEL - M4	PEL - Racket ⓕⓜ	PEL - UNIX Shell
PEL - Chez ⓕⓜ	PEL - Forth Ⓚ	PEL - Hy (python) ⓜ	PEL-Modula	PEL - ReasonML	PEL - V
PEL - Chibi ⓕⓜ	Fortran		PEL - NetRexx	Rebol	PEL -Zig Ⓢ
PEL - Chicken ⓕⓜ			PEL - Nim ⓜⓈ	Red	
PEL - Clojure ⓕⓜ			PEL-Objective-C	PEL - REXX	
Common Lisp ⓕⓜ			PEL - OCaml ⓘⓕ	PEL - Ruby	
Crystal			PEL - Odin Ⓢ	PEL - Rust Ⓢ	

Future support for APL, Carbon, Crystal, Elm, Groovy, Haxe, Kotlin, Pony, Purescript, ReasonML, Rebol, Red, Scala, Typescript and documentation of support for Fortran (based on my need for them or requests).

- 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.