PEL Quick Access Topics Index

Last updated on: 2026-02-09 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 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
- EmacsConf

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

Prefix/Suffix & Numeric Arg Keys

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

Emacs Features

Emacs Manual , Guided Tour of Emacs , Emacs Lisp Manual

- Emacs Docs: Emacs, Emacs Lisp
- Mastering Emacs, Awesome-Emacs
- MELPA and GNU ELPA

The tables at right describe Emacs concepts/features commands & key bindings. Cell background is light-blue for major mode, light-red for minor mode specifics, grey for links to sections of tables. Cells link titles starting with Σ are Emacs generic features, blue links are external packages. The green links are mostly PEL extensions. 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.

1979 EMACS Intro memo by R.M. Stallman

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

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

Build Tools

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

Data Serialization & Configuration

CWL	HCL/Terraform	JSON	PKL	XML	xmake

Modelling

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

Other File Formats

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

Hardware Description Languages

- Verilog	- VHDL	Language Server & Tools for HDL			

Lightweight Markup Languages

AsciiDoc	Markdown	Org-Mode	reStructuredText		Haml

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
- Ada	Crystal	- Gambit	- Janet	-Pascal	Scala
- Algol	- D	- Gerbil	- Java	- Perl (perl5)	- Scheme
- AppleScript	- Dart	- GNU Guile	- Javascript	PHP future	- Schen
APL	- Eiffel	- Gleam	- Julia	- Pike	-Seed7
- Arc	- Elm	- Go	Kotlin	Pony	SQL
- awk	- Elixir	Groovy	- LFE	- Purescript	-Smalltalk
- C	- Emacs Lisp	- Haskell	- Lua	- Python	-Swift
C# future	- Erlang	Haxe	- M4	R	- Tcl
- C++	- Factor	- Hy (python)	-Modula	- Racket	- Typescript
- C3	FAUST future		Mojo future	- ReasonML	- UNIX Shell
Carbon future	Fennel future		- NetRexx	Rebol	- V
- Chez	- Forth		- Nim	Red	Vala future
- Chibi	- Fortran		-Objective-C	- REXX	-Zig
- Chicken			- OCaml	Rocq future	
- Clojure			- Odin	- Ruby	
Common Lisp				- Rust	

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, Elm, Groovy, Haxe, Kotlin, Pony, Purescript, ReasonML, Rebol, Red, Scala, Typescript (based on my need for them or requests).