PEL Topics Index

Emacs Reference Cards ☞With PEL you can access these via the `<f11> ? e r` key sequence. See ∑ 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 • PEL repo • PEL Readme • PEL Manual	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 quickly and reach a vast amount of information quickly. ☞From within Emacs open this topic index PDF by typing the `<f11> ? <f1>` key sequence. ☞The symbols, **colour coding** and various other conventions are described in the ➣Legend PDF.

• General Information.	➣Legend	➣Recommended Emacs User Option	➣Themes	
• Development Information	➣PEL	▰iMenu/Speedbar support	▰PEL Naming Conventions	
• Migration Guide	➣CRiSP ⮂ Emacs			

macOS Specific	macOS Keys	terminal settings		

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

Key Prefixes & Suffixes	∑▰ Modifier Keys	∑▰ Numkeypad	➣PEL	▰Keys - Fn	▰Keys - F11

∑ Emacs Features	The links that start with only ∑ Emacs generic features, the blue links are external packages. The green links are mostly PEL extensions.					
See a Guided Tour of Emacs. The PEL tables named at right describe the Emacs commands and key bindings for generic Emacs concepts and features. Emacs commands can be executed by name or bound to key sequences. The commands may have arguments and keys can express them. See: • Emacs Keys Emacs uses a concept of modes. See: • Emacs Major and Minor Modes • Major Modes • Minor Modes • Choosing Modes PEL provides several key sequences to toggle minor modes, described in the relevant PDFs.	∑ Abbreviations	∑ Cursor	∑ Filling/ Justification	∑ℒ𝔛- Lispy	∑ Scrolling	∑ Time Tracking
	∑ Align	∑ Customize	∑ Frames	∑ Marking	∑ Search/Replace	∑ Transpose
	∑ Auto-Completion	∑ Cut & Paste	∑ Grep	∑ Menus	∑ Semantic	∑𝔛 Treemacs
	∑ Autosave/Backup	∑ Diff & Merge	∑ Help/Info	∑ Mode Line	∑ Sessions	∑ Undo/Redo/ Repeat/Arg
	∑ Bookmarks	∑ Dired	∑ Hide/Show	∑ Mouse	∑ Shells, REPLs & terminal emulators	∑ VCS-Git 𝔛Magit
	∑ Buffers	∑ Display - Lines	∑ Highlight	∑ Narrowing	∑𝔛 Smartparens	∑ VCS-Mercurial
	∑ Case Conversions	∑ Drawing	∑ ibuffer-mode	∑ Navigation	∑ Sorting	∑ Web
	∑ Closing/ Suspending	∑ Enriched Text	∑ Indentation	∑ Outline	∑ Speedbar	∑ Whitespace
	∑ Comments	∑ Faces/Fonts	∑ Input Method	∑ Packages	∑ Spell Checking	∑ Windows
	∑ Completion/Input	∑P Fast Startup	∑ Inserting Text	∑𝔛 Projectile	∑ SyntaxCheck	∑ Xref - Cross References
	∑ Counting	∑ File-mngt	∑ Key-Chords	∑ Rectangles	T Templates	
	∑Ⓜ CUA	∑ File/Directory Variables	∑ Keyboard Macros	∑ Registers	∑ Text Modes	

ℒ𝔛 - Emacs Lisp concepts & tools	ℒ ERT (Emacs Lisp Regression Testing)	ℒ Hooks	ℒ⚙ - Emacs Lisp Types	

XRef - Cross Reference Tools See also: ∑ Xref	Emacs supports various cross reference mechanisms described in the ∑ 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. 🚧This is work in progress.	
	Xref-Support	Xref-Backend

Build Tools	PEL has support for several build tools but they are not all documented in a page. Aside from the list below, PEL supports installation and partial setup of the following tools: • 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.

𝔅𝔩 - Make			

Data Serialization	Ⓓ CWL	Ⓓ YAML		

Data Modelling/ Specification	Ⓢ ASN.1 asn1-mode	Ⓢ MIB snmp-mode	Ⓢ YANG	

Markup Languages	Ⓜ AsciiDoc	Ⓜ Graphviz Dot	Ⓜ Markdown	Ⓜ Org-Mode	Ⓜ PlantUML	Ⓜ reStructuredText

Programming Languages **Main Paradigm of Programming Language Families** • *Actor Model:* Ⓐ • *Concatenative:* Ⓚ • *Concurrent:* Ⓒ • *Functional:* Ⓕ *Pure:* Ⓕ • *Imperative:* Ⓘ or no token • *Has Syntactic Macros:* Ⓜ	Emacs has major mode support for several programming languages. PEL currently adds extra support for some of them, listed below. • The number of programming languages supported explicitly by PEL will grow over time.					
	BEAM Programming Languages	*Functional Languages*	Javascript target	**Lisp Family Languages**	Lisp-like Languages	**Command Line Scripting Languages**
	Curly Bracket Languages	**Java Virtual Machine Languages**	**ML Family Languages**	**Scheme Language Dialects**	**Stack Based Languages**	**OS App Control Scripting Languages**
• The programming languages supported by PEL are listed here in alphabetical order. • PEL also provides basic support for other programming languages not listed here. • Emacs supports other programming languages directly, not listed here. **Upcoming support** for Elm, Purescript, ReasonML, Typescript and documentation of support for Javascript.	The following lists the programming languages in alphabetical order. • The cell colours give a coarse indication of the programming language family(ies).					
	𝔅𝔩⌘- AppleScript	𝔅𝔩 - Clojure Ⓕ Ⓜ	𝔅𝔩 - Forth Ⓚ	𝔅𝔩 - Hy (python) Ⓜ	𝔅𝔩 - OCaml ⒾⒻ	𝔅𝔩 - Ruby
	𝔅𝔩 - Arc Ⓕ Ⓜ	Common Lisp Ⓕ Ⓜ	𝔅𝔩 - Gambit Ⓕ Ⓜ	𝔅𝔩 - Janet ⒾⒻ Ⓜ	𝔅𝔩 - Perl	𝔅𝔩 - Rust
	𝔅𝔩 - C	𝔅𝔩 - D ⒾⒻⒶ	𝔅𝔩 - Gerbil Ⓕ Ⓜ Ⓐ	𝔅𝔩 - Javascript	𝔅𝔩 - Python	𝔅𝔩 - Scheme Ⓕ Ⓜ
	𝔅𝔩 - C++	𝔅𝔩 - Elm Ⓕ	𝔅𝔩 - GNU Guile Ⓕ Ⓜ	𝔅𝔩 - Julia Ⓜ	𝔅𝔩 - Purescript Ⓕ	𝔅𝔩 - Typescript
	𝔅𝔩 - Chez Ⓕ Ⓜ	𝔅𝔩 - Elixir ⒸⓂ ⒻⒶ	𝔅𝔩 - Gleam Ⓕ	𝔅𝔩 - LFE ⒸⒻ Ⓜ ⒻⒶ	𝔅𝔩 - Racket Ⓕ Ⓜ	𝔅𝔩 - UNIX Shell
	𝔅𝔩 - Chibi Ⓕ Ⓜ	ℒ𝔅𝔩 - Emacs Lisp	𝔅𝔩 - Go	𝔅𝔩 - NetRexx	𝔅𝔩 - ReasonML	𝔅𝔩 - V
	𝔅𝔩 - Chicken Ⓕ Ⓜ	𝔅𝔩 - Erlang ⒸⒻⒶ	𝔅𝔩 - Haskell Ⓕ	𝔅𝔩 - Nim Ⓜ	𝔅𝔩 - REXX	