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 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.
  - Firefox does that. 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.
👌 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

These PEL tables describe the Emacs commands and key bindings for generic concepts and features.

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.

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**

The links that start with only ∑ are built-in Emacs, the links that are blue are external packages.

∑ Abbreviations	∑M CUA	∑ File/Directory Variables	∑ Keyboard Macros	∑ Registers	∑ Text Modes
∑ Align	∑ Cursor	∑ Filling/Justification	∑ℙℓ- Lispy	∑ Scrolling	∑ Transpose
∑ Auto-Completion	∑ Customize	∑ Frames	∑ Marking	∑ Search/Replace	∑𝔁 Treemacs
∑ Autosave/Backup	∑ Cut & Paste	∑ Grep	∑ Menus	∑ Semantic	∑ Undo/Redo/Repeat/Arg
∑ Bookmarks	∑ Diff & Merge	∑ Help/Info	∑ Mode Line	∑ Sessions	∑ VCS-Mercurial
∑ Buffers	∑ Dired	∑ Hide/Show	∑ Mouse	∑ Shells, REPLs & terminal emulators	∑ Web
∑ Case Conversions	∑ Display - Lines	∑ Highlight	∑ Narrowing	∑ Sorting	∑ Whitespace
∑ Closing/Suspending	∑ Drawing	∑ ibuffer-mode	∑ Navigation	∑ Speedbar	∑ Windows
∑ Comments	∑ Enriched Text	∑ Indentation	∑ Packages	∑ Spell Checking	∑ Xref - Cross References
∑ Completion/Input	∑ Faces/Fonts	∑ Inserting Text	∑ Projectile	∑ SyntaxCheck	
∑ Counting	∑ File-mngt	∑ Key-Chords	∑ Rectangles	T Templates	

XRef - Cross Reference Tools

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.

ℙℓ - Make

Data Serialization Languages

Ⓓ CWL	Ⓓ YAML

Markup Languages

M AsciiDoc	M Graphviz Dot	M Markdown	M Outline/Org-Mode	M PlantUML	M reStructuredText

Programming Languages

Emacs has 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.

£ℙℓ - Emacs Lisp concepts & Tools	£ ERT	£ Hooks

Programming Language Families
- *Functional:* ⓕ
- *Pure Functional:* Ⓕ

BEAM Programming Languages	Functional Languages ⓕ/Ⓕ	Javascript target	Lisp Family Languages	Command Line Scripting Languages	
Curly Bracket Languages	Java Virtual Machine Languages	ML Family Languages	Stack Based Languages	OS App Control Scripting Languages	

All Programming 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 and Racket.

The following lists the programming languages in alphabetical order.
- The cell colours give a coarse indication of the programming language family(ies).

ℙℓ🍎- AppleScript	ℙℓ - D	ℙℓ - Go	ℙℓ - LFE	ℙℓ - Purescript Ⓕ	ℙℓ - Scheme ⓕ
🚧ℙℓ - Arc ⓕ	ℙℓ - Elm Ⓕ	ℙℓ - Gleam	ℙℓ - NetRexx	🚧ℙℓ - Racket	ℙℓ - Typescript
ℙℓ - C	ℙℓ - Elixir ⓕ	ℙℓ - Haskell Ⓕ	ℙℓ - Nim	ℙℓ - ReasonML	ℙℓ - UNIX Shell
ℙℓ - C++	£ℙℓ - Emacs Lisp	ℙℓ - Hy	ℙℓ - OCaml ⓕ	ℙℓ - REXX	ℙℓ - V
ℙℓ - Clojure ⓕ	ℙℓ - Erlang ⓕ	ℙℓ - Javascript	ℙℓ - Perl	ℙℓ - Ruby	
ℙℓ - Common Lisp ⓕ	ℙℓ - Forth	ℙℓ - Julia	ℙℓ - Python	ℙℓ - Rust	