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**
- **PEL NEWS** ✎
- General Information.
- Development Information
- Migration Guide

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. More help topics with `<f11> ? p` keys.

👆 The symbols, **colour coding** and various other conventions are described in the **➤Legend** PDF.

➤Legend	➤Recommended Emacs User Option	➤Themes	
➤PEL	▦iMenu/Speedbar support	▦PEL Naming Conventions	
➤CRiSP ⇌ Emacs			

OS Desktop Key Bindings
(Bindings that don't clash with PEL)

macOS Fct Keys	macOS Keys	Ubuntu 16.04 Desktop Keys	
	terminal settings	Mint 20 Desktop Keys	

🚦 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

- A **Guided Tour of Emacs**.
- **Awesome-Emacs**
- **MELPA** and **GNU ELPA**

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.
- **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 key sequences to toggle minor modes.

The links that start with only Ⓧ Emacs generic features, the blue links are external packages. The green links are mostly PEL extensions.

Ⓧ Abbreviations	Ⓧ Cursor	Ⓧ Fill/Justify	Ⓧ𝒳- 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 (colors)	Ⓧ Narrowing	Ⓧ𝒳 Smartparens	Ⓧ VCS-Mercurial
Ⓧ Case Conversions	Ⓧ Drawing	Ⓧ ibuffer-mode	Ⓧ Navigation	Ⓧ Sorting	Ⓧ VCS-Subversion
Ⓧ Closing/Suspending	Ⓧ Enriched Text	Ⓧ Indentation	Ⓧ Outline	Ⓧ Speedbar	Ⓧ Web
Ⓧ Comments	Ⓧ Faces/Fonts	Ⓧ Input Method	Ⓧ Packages	Ⓧ Spell Checking	Ⓧ Whitespace
Ⓧ Completion/Input	ℙP Fast Startup	Ⓧ Inserting Text	Ⓧ𝒳 Projectile	Ⓧ SyntaxCheck	Ⓧ Windows
Ⓧ Counting	Ⓧ File-mngt	Ⓧ Key-Chords	Ⓧ Rectangles	T Templates	Ⓧ Xref - Cross References
ⓍM CUA	Ⓧ File/Dir Variables	Ⓧ Keyboard Macros	Ⓧ Registers	Ⓧ Text Modes	

ℒℙ𝓁 - 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			

PEL supports installation and partial setup of the following tools:

Build Tools & Preprocessor

PEL has support for several build tools but they are not all documented in a page.

- **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 - M4	ℙ𝓁 - Make			

Data Serialization

Ⓓ CWL	Ⓓ YAML			

Data Modelling/ Specification

Ⓢ ASN.1 **asn1-mode**	Ⓢ MIB **snmp-mode**	Ⓢ YANG		

Hardware Description Languages

Verilog 🚧future	VHDL 🚧future			

Text Markup Languages

ℳ AsciiDoc	ℳ Markdown	ℳ Org-Mode	ℳ reStructuredText	

Graphics Markup

ℳ Graphviz Dot	ℳ MscGen	ℳ PlantUML		

Programming Languages

Main Paradigm of Programming Language Families
- **Actor Model:** Ⓐ
- **Concatenative:** Ⓚ
- **Concurrent:** Ⓒ
- **Functional:** ⓕ **Pure:** Ⓕ
- **Imperative:** ⓘ or no token
- **Object Oriented** ∞
- **Has Syntactic Macros:** ⓜ
- 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.

Future support for Crystal, Elm, Kotlin, Lua, Purescript, ReasonML, Seed7, Typescript, Zig and documentation of support for Ada, Fortran, Javascript, Java, Modula, Pascal (based on my need for them or requests (if any)).

Emacs has major mode support for several programming languages. PEL currently adds extra support for some of them, listed below.

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

ℙ𝓁 - AppleScript	Crystal 🚧future	Fortran 🚧future	ℙ𝓁 - Janet ⓘⓕⓜ	Objective-C 🚧future	ℙ𝓁 - Rust
Ada 🚧future	ℙ𝓁 - D ⓘⓕⒻⒶ	ℙ𝓁 - Gambit ⓕⓜ	Java 🚧future	ℙ𝓁 - OCaml ⓘⓕ	Scala 🚧future
ℙ𝓁 - Arc ⓕⓜ	Dart 🚧future	ℙ𝓁 - Gerbil ⓕⓜⒶ	ℙ𝓁 - Javascript	Pascal 🚧future	ℙ𝓁 - Scheme ⓕⓜ
ℙ𝓁 - C	Eiffel 🚧future	ℙ𝓁 - GNU Guile ⓕⓜ	ℙ𝓁 - Julia ⓜ	ℙ𝓁 - Perl	Seed7 🚧future
ℙ𝓁 - C++	ℙ𝓁 - Elm 🚧future Ⓕ	ℙ𝓁 - Gleam	Kotlin 🚧future	ℙ𝓁 - Python	Swift 🚧future
ℙ𝓁 - Chez ⓕⓜ	ℙ𝓁 - Elixir ⒸⓜⓕⒶ	ℙ𝓁 - Go	ℙ𝓁 - LFE ⒸⓜⓕⒶ	ℙ𝓁 - Purescript Ⓕ	ℙ𝓁 - Tcl 🚧future ⓕⓘ
ℙ𝓁 - Chibi ⓕⓜ	ℒℙ𝓁 - Emacs Lisp	Groovy 🚧future	Lua 🚧future	ℙ𝓁 - Racket ⓕⓜ	ℙ𝓁 - Typescript
ℙ𝓁 - Chicken ⓕⓜ	ℙ𝓁 - Erlang ⒸⓕⒶ	ℙ𝓁 - Haskell Ⓕ	Modula 🚧future	ℙ𝓁 - ReasonML 🚧	ℙ𝓁 - UNIX Shell
ℙ𝓁 - Clojure ⓕⓜ	Factor Ⓚⓕ∞ⓜ	Haxe 🚧future	ℙ𝓁 - NetRexx	ℙ𝓁 - REXX	ℙ𝓁 - V
Common Lisp ⓕⓜ	ℙ𝓁 - Forth Ⓚ	ℙ𝓁 - Hy (python) ⓜ	ℙ𝓁 - Nim ⓜ	ℙ𝓁 - Ruby	Zig 🚧future