

PEL Topics Index

Last updated on:2024-10-24

Note: with PEL, type <f11> <f1> to open this PDF index.

Emacs Reference Cards

👉 With PEL you can access these via the <f11> ? e r key sequence.

See 📄 Help/Info

➤ PEL Overview (license)

• PEL repo

• PEL Readme

• PEL Manual

• PEL NEWS 🗞️

• Discussions

• General Information.

• Startup

• Development Information

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

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

🖱️ Feature Comparisons

Key Prefixes & Suffixes

🔠 Emacs Features

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. 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.

🔠 Completion Modes Compatibility

🔠 Speedbar/iMenu Mode Compatibility

🔠 Shells/Terminals Comparisons

🔠 Modifier Keys

🔠 Numkeypad

➤ PEL

🔠 Keys - Fn

🔠 Keys - F11

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

🔠 Abbreviations

🔠 Diff & Merge

🔠 Grep

🔠 Marking

🔠 Scrolling

🔠 Tab Bar

🔠 Align

🔠 Dired

🔠 Help/Info

🔠 Menus

🔠 Search/Replace

T Templates

🔠 Auto-Completion

🔠 Display - Lines

🔠 Hide/Show

🔠 Mode Line

🔠 Sessions

🔠 Text Modes

🔠 Autosave/Backup

🔠 Drawing

🔠 Highlight (colors)

🔠 Mouse

🔠 start Shells/REPLs

🔠 Time Tracking

🔠 Bookmarks

🔠 Enriched Text

🔠 ibuffer-mode

🔠 Narrowing

🔠 shell-mode

🔠 Tramp

🔠 Buffers

🔠 Faces/Fonts

🔠 Indentation

🔠 Navigation

🔠 term-mode

🔠 Transpose text

🔠 Case Conversions

🔠 P Fast Startup

🔠 Input Method

🔠 Object Files

🔠 eat-mode

🔠 X Treemacs

🔠 Close/Suspend

🔠 File Encoding

🔠 Inserting Text

🔠 Outline

🔠 vterm-mode

🔠 Undo/Redo

🔠 Comments

🔠 File-mngt

🔠 Key-Chords

🔠 Packages

🔠 X Smartparens

🔠 VCS-Git X Magit

🔠 Completion/Input

🔠 File/Dir Variables

🔠 Keyboard Macros

🔠 X Projectile

🔠 Sorting

🔠 VCS-Mercurial

🔠 Counting

🔠 Fill/Justify

🔠 I X - Lisp

🔠 Rectangles

🔠 Speedbar

🔠 VCS-Subversion

🔠 M CUA

🔠 Frames

🔠 Registers

🔠 Spell Checking

🔠 Web

🔠 Cursor

🔠 SyntaxCheck

🔠 Whitespace

🔠 Customize

🔠 Windows

🔠 Cut & Paste

🔠 Xref - Cross Refs

🔠 display-buffer

🔠 * - ELisp Types

🔠 ERT (regr-testing)

🔠 Hooks

XRef - Cross Reference Tools

See also: 🔠 Xref

🖱️ Xref-Support

🖱️ Xref-Frontend

🖱️ Xref-Backend

PEL supports installation and partial setup of the following tools:

Build Tools & Preprocessor

Data Serialization

Data Modelling/ Specification

Other File Formats

Hardware Description Languages

Text Markup Languages

• Graphics Markup

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.

🔠 CMake 🛠️future

🔠 M4

🔠 Make gmake

🔠 CWL

🔠 YAML

Utility: GNU readline

🔠 ASN.1 asn1-mode

🔠 MIB snmp-mode

🔠 YANG

Is -l

RPM Files 🐼

🔠 X.509 Certificates

Verilog 🛠️future

VHDL 🛠️future

🔠 AsciiDoc

🔠 Markdown

🔠 Org-Mode

🔠 reStructuredText

OS App Control Scripting Languages

🔠 Graphviz Dot

🔠 MscGen

🔠 PlantUML

🔠 Apple - AppleScript

Programming Languages

Main Paradigm of Programming Language Families

• Actor Model: 🌀

• Concatenative 🌀

• Concurrent: 🌀

• Domain Specific 🌀

• Functional: 🌀 Pure: 🌀

• Imperative: 🌀 or no token

• Object Oriented 🌀

• Has Syntactic Macros: 🌀

The programming languages supported by PEL are listed here in alphabetical order.

Emacs (and PEL) also provides basic support for other programming languages not listed here.

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

The following lists the programming languages in alphabetical order. Cell colours refer to the programming language family(ies).

Ada 🛠️future

🔠 - D 🌀🌀🌀

🔠 - Gambit 🌀🌀

🔠 - Janet 🌀🌀🌀

Objective-C 🛠️future

Scala 🛠️future

🔠 - Arc 🌀🌀

Dart 🛠️future

🔠 - Gerbil 🌀🌀🌀

Java 🛠️future

🔠 - OCaml 🌀🌀

🔠 - Scheme 🌀🌀

🔠 - awk 🌀

Eiffel 🛠️future

🔠 - GNU Guile 🌀🌀

🔠 - Javascript 🛠️

Pascal 🛠️future

Seed7 🛠️future

🔠 - C

🔠 - Elm 🛠️future 🌀

🔠 - Gleam

🔠 - Julia 🌀

🔠 - Perl

Swift 🛠️future

🔠 - C++

🔠 - Elixir 🌀🌀🌀🌀

🔠 - Go

Kotlin 🛠️future

🔠 - Python

🔠 - Tcl 🛠️future 🌀🌀

🔠 - Chez 🌀🌀

🔠 - Emacs Lisp

Groovy 🛠️future

🔠 - LFE 🌀🌀🌀🌀

🔠 - Purescript 🌀

🔠 - Typescript 🛠️

🔠 - Chibi 🌀🌀

🔠 - Erlang 🌀🌀🌀

🔠 - Haskell 🌀

Lua 🛠️future

🔠 - Racket 🌀🌀

🔠 - UNIX Shell

🔠 - Chicken 🌀🌀

Factor 🌀🌀🌀🌀🌀

Haxe 🛠️future

Modula 🛠️future

🔠 - ReasonML 🛠️

🔠 - V

🔠 - Clojure 🌀🌀

🔠 - Forth 🌀

🔠 - Hy (python) 🌀

🔠 - NetRexx

🔠 - REXX

Zig 🛠️future

Common Lisp 🌀🌀

Fortran 🛠️future

🔠 - Nim 🌀

🔠 - Ruby

Crystal 🛠️future

🔠 - Rust