

PEL Quick Access Topics Index

Last updated on: 2025-12-26		Note: with PEL; type <a href="#">&lt;f11&gt; &lt;f1&gt;</a> to open this PDF index.					
<div>GNU Emacs</div> <div>Reference Cards</div>		With PEL, access these PDF cards from within Emacs with the <a href="#">&lt;f11&gt; ? e r</a> key sequence. See <a href="#">ℹ Help/Info</a> for more info. Links to PDF version of official English version of the quick reference cards for <a href="#">GNU Emacs</a> and popular external packages.					
<div><div>• Emacs Release History</div><div>• EmacsWiki</div></div> <div><div>• Emacs project repo</div></div>		Emacs	Calc	Gnus	Magit Cheatsheet	Org	Viper
		Emacs survival card	Dired	Gnus booklet	Magit Ref-card		VIP
<div>➤ PEL</div> <div><div>• Repo</div><div>• Manual</div><div>• Discussions</div></div> <div><div>Readme License</div><div>NEWS🔧</div></div>		<div>• Emacs Mailing Lists</div> <div>• Contribute to Emacs</div> <div>• EmacsConf</div>					
<div>Terminal Multiplexers:</div> <div>GNU screen , Tmux</div> <div>Command Line Scripting</div> <div>Languages: bash, sh, zsh</div> <div>🐉: GNU readline, ls -l, ssh</div>		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 🖨️		🍏 macOS Fct Keys		🍏 macOS Keys		🐉Mint 20 Desktop Keys	
(Bindings that don't clash with PEL)				🍏 terminal settings		🐉Rocky Linux 8 Desktop Keys	
🐉 Feature Comparisons		🐉 Completion Modes Compatibility		🐉 Speedbar/iMenu Mode Compatibility		🐉 Shells/Terminals Comparisons	
Key Prefixes & Suffixes		⌘ Modifier Keys	⌘ Numkeypad	⌘ Keys - Fn	⌘ Keys - F11	⌘ Keys - F12	➤PEL
ℹ Emacs Features		ℹ Abbreviations	Debuggers🐉🐉	ℹ Grep	ℹ Man pages	ℹ Scrolling	ℹ Tab Bar
ℹ Emacs Manual , Guided Tour of Emacs , Emacs Lisp Manual		ℹ Align	ℹ Diff & Merge	ℹ Help/Info	ℹ Marking	ℹ Search/Replace	ℹ Templates
• Emacs Docs: Emacs, Emacs Lisp		ℹ Auto-Completion	ℹ Dired	ℹ Hide/Show	ℹ Menus ℹiMenu	ℹ Sessions	ℹ Text Modes
• Mastering Emacs, Awesome-Emacs		ℹ Autosave/Backup	ℹ Display - Lines	ℹ Highlight (colors)	ℹ Mode Line	ℹ start Shells/REPLs	ℹ Time Stamps
• MELPA and GNU ELPA		ℹ Bookmarks	ℹ Drawing	ℹ ibuffer-mode	ℹ Mouse	ℹ shell-mode	ℹ Time Tracking
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 ⓘ 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.		ℹ Buffers	ℹ Eldoc	ℹ Indentation	ℹ Narrowing	ℹ term-mode	ℹ Tramp 🐉
• Emacs Keys		ℹ Case Conversions	ℹ Enriched Text	ℹ Input Method	ℹ Navigation	eat-mode	ℹ Transpose text
• Numeric Arguments		ℹ Close/Suspend	ℹ Execute Cmds	ℹ Inserting Text	ℹ Object Files	vterm-mode	ℹ x Treemacs
You can also:		ℹ Comments	ℹ Exec Shell Cmds	ℹ Key-Chords	ℹ Outline	ℹ x Smartparens	ℹ Tree Sitter
• Run Command by Name		ℹ Compilation Mode	ℹ Faces/Fonts	ℹ Keyboard Macros	ℹ Packages	ℹ Sorting	ℹ Undo/Redo/Repeat
Emacs uses a concept of modes:		ℹ Completion/Input	ℹP Fast Startup	ℹx - Lispy	Programming	ℹ Speech To Text	ℹ VCS-Git xMagit
• Emacs Major and Minor Modes		ℹ Counting	ℹ File Encoding	Logging key strokes	ℹ Project Tools	ℹ Speedbar	ℹ VCS-Mercurial
• Major Modes		ℹM CUA	ℹ File-mngt		ℹx Projectile	ℹ Spell Checking	ℹ VCS-Subversion
• Minor Modes		ℹ Cursor	ℹ File/Dir Variables		ℹ Recursive Edit	ℹ SyntaxCheck	ℹ Web
• Choosing Modes		ℹ Customize	ℹ Fill/Justify		ℹ Rectangles		ℹ Whitespace
PEL provides several key sequences to toggle minor modes.		ℹ Cut & Paste	ℹ Frames		ℹ Registers		ℹ Windows
							Writing Tools
							ℹ Xref - Cross Refs
ℹx - Emacs Lisp concepts	& tools	ℹ display-buffer	ℹx - ELisp Types	ℹ Hooks	ℹ Elisp Build Tools	ℹ ERT (regr-testing)	
Parsing tools, Indentation	ℹ Xref Tools:	🐉 Language Servers	🐉 Tree-sitter	🐉 Indentation Styles	🐉 Xref-Support	🐉 Xref-Frontend	🐉 Xref-Backend
Build Tools		ℹ - CMake 🐉🐉	ℹ - Make gmake	ℹ - Meson	ℹ - Ninja	ℹ - Nix	ℹ - Tup
Data Serialization & Configuration		ⓘ CWL	ⓘ JSON 🐉🐉	ⓘ PKL 🐉🐉	ⓘ XML 🐉🐉🐉	ⓘ YAML	xmake
Modelling		ⓘ ASN.1 asn1-mode	ⓘ MIB snmp-mode	ⓘ YANG			
Other File Formats		Binary, Object, Executable Files		Log Files	RFC (RFC @ Wikipedia)		SSH files 🐉ssh
		ℹ 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		
• 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	
Main Paradigm of Programming Languages							
• Actor Model: Ⓐ Array X	ⓘ - Ada 🐉🐉 ↗ Ⓢ	ⓘ - D Ⓛ Ⓣ Ⓐ	ⓘ - Gambit Ⓣ Ⓜ	ⓘ - Janet Ⓛ Ⓣ Ⓜ	ⓘ -Pascal	Scala 🐉🐉	
• Concatenative Ⓚ Concurrent: ©	ⓘ🍏 - AppleScript	ⓘ - Dart ↗ Ⓣ Ⓢ	ⓘ - Gerbil Ⓣ Ⓜ Ⓐ	ⓘ - Java 🐉🐉	ⓘ - Perl (perl5)	ⓘ - Scheme Ⓣ Ⓜ	
• Domain Specific Ⓓ	ⓘ APL 🐉🐉	ⓘ - Eiffel 🐉🐉 Ⓢ Ⓢ	ⓘ - GNU Guile Ⓣ Ⓜ	ⓘ - Javascript 🐉🐉	ⓘ - Pike d Ⓛ Ⓢ Ⓢ	ⓘ -Seed7 🐉🐉 Ⓢ Ⓢ ↗	
• Dynamic d Extensible ©	ⓘ - Arc Ⓣ Ⓜ	ⓘ - Elm 🐉🐉 Ⓟ	ⓘ - Gleam	ⓘ - Julia Ⓜ	ⓘ Pony 🐉🐉	ⓘ -Smalltalk 🐉🐉 Ⓢ Ⓢ Ⓢ	
• Functional: Ⓣ Pure: Ⓟ	ⓘ - awk Ⓢ	ⓘ - Elixir Ⓢ Ⓢ Ⓣ Ⓐ	ⓘ - Go Ⓢ	ⓘ Kotlin 🐉🐉	ⓘ - Purescript 🐉🐉 Ⓟ	ⓘ -Swift	
• Generic Ⓞ	ⓘ - C Ⓢ	ⓘx - Emacs Lisp	ⓘ Groovy 🐉🐉	ⓘ - LFE Ⓢ Ⓢ Ⓣ Ⓐ	ⓘ - Python d Ⓢ Ⓢ Ⓢ	ⓘ - Tcl Ⓣ Ⓛ	
• Imperative: Ⓛ or no token	ⓘ C# 🐉🐉 future	ⓘ - Erlang Ⓢ Ⓣ Ⓐ	ⓘ - Haskell Ⓟ	ⓘ -Lua Ⓣ Ⓢ Ⓢ Ⓢ	ⓘ R 🐉🐉 Ⓢ Ⓢ Ⓢ Ⓢ X	ⓘ - Typescript 🐉🐉	
• Object Oriented Ⓢ Procedural Ⓟ	ⓘ - C++ Ⓢ Ⓢ	ⓘ - Factor Ⓚ Ⓣ Ⓢ Ⓢ	ⓘ Haxe 🐉🐉	ⓘ - M4	ⓘ - Racket Ⓣ Ⓜ	ⓘ - UNIX Shell	
• Has Syntactic Macros: Ⓜ	ⓘ - C3 Ⓢ	ⓘ - Forth Ⓚ	ⓘ - Hy (python) Ⓜ	ⓘ -Modula	ⓘ - ReasonML 🐉🐉🐉	ⓘ - V	
• Multi-paradigm ↗ Reflective	ⓘ Carbon 🐉🐉 future	ⓘ Fortran 🐉🐉		ⓘ - NetRexx	ⓘ Rebol 🐉🐉	ⓘ - Zig Ⓢ	
• System Level Ⓢ	ⓘ - Chez Ⓣ Ⓜ			ⓘ - Nim Ⓜ Ⓢ	ⓘ Red 🐉🐉		
The programming languages supported by PEL are listed here in alphabetical order.	ⓘ - Chibi Ⓣ Ⓜ			ⓘ -Objective-C 🐉🐉	ⓘ - REXX		
Emacs (and PEL) also provides basic support for some of the one PEL does not support and for other programming languages not listed here.	ⓘ - Chicken Ⓣ Ⓜ			ⓘ - OCaml Ⓛ Ⓣ	ⓘ - Ruby		
	ⓘ - Clojure Ⓣ Ⓜ			ⓘ - Odin Ⓢ	ⓘ - Rust Ⓢ		
	ⓘ Common Lisp Ⓣ Ⓜ						
	ⓘ Crystal 🐉🐉						
Future support for APL, Carbon, Crystal, Elm, Groovy, Haxe, Kotlin, Pony, Purescript, ReasonML, Rebol, Red, Scala, Typescript and documentation of support for Fortran (based on my need for them or requests).							