## **PEL Topics Index**

		Last updated on:	2025-01-12		Note: with PEL	_, type <u><f11> <f1></f1></f11></u> t	o open this PDF index.
Emacs Reference Cards			the PDF version of official English version of the quick reference cards for <u>GNU Emacs</u> and popular external pac macs key bindings as well, these cards provide useful complement to what PEL provides.			external packages.	
With PEL, access these cards from Emacs with the <f11> ? e r key sequence. See ∑ Help/Info for more info.</f11>		Emacs	Calc	Gnus	Magit Cheatsheet	Org	Viper
		Emacs survival card	Dired	Gnus booklet	Magit Ref-card		VIP
➤ PEL Overview PEL li	license	This table holds links to	the <b>PEL file tables</b> (ho	sted on Github as raw P	DF files).		
• PEL repo	<u></u>	This table holds links to the <u>PEL file tables</u> (hosted on Github as raw PDF files).  Solution of the best user experience, use a browser that can render PDF directly instead of downloading.					
<ul> <li>PEL Readme</li> <li>PEL Manual</li> <li>PEL NEWS</li> <li>Discussions</li> </ul>		<ul> <li>Mozilla Firefox (version &gt; 78) does that perfectly. You may need to activate a plug-in for other browsers.</li> <li>With that in place, you can browse through all the PDFs and reach a vast amount of information quickly.</li> </ul>					
		From within Emacs open this topic index PDF by typing the <f11> ? <f1> key sequence. More help topics with <f11> ? p_keys.</f11></f1></f11>					
		▼ The symbols, colour coding and various other conventions are described in the  ➤ Legend PDF.					
Terminal Multiplexers: GNU screen , Tmux Command Line Scripting Languages: bash, sh, zsh Cmdline: GNU readline, ls -I		<u>≻Legend</u>	≻Recommended Ema	acs User Option	<u>≻Themes</u>	Migrate from CRiSP	
			Run Emacs daemon 8	k clients <b>∉</b> ூ	■iMenu/Speedbar su	<u>upport</u>	
			PEL Naming Conve	antions	PEL Environment V	/ariables	PEL utilities
omanio <u>arro rodanto, io .</u>		4 00 - 111		enuona 	P LL LIIVIIOIIIIEIIL V	anabies	r LL utilities
OS Desktop Key Bindings (Bindings that don't clash with PEL)				Mint 20 Desktop Ke	<u>eys</u>	<b>16.04 Desk</b>	top Keys
			<b>terminal settings</b>	Nocky Linux 8 Designation	ktop Keys		
Feature Comparisons		Completion Modes	Compatibility	§ Speedbar/iMenu M	lode Compatibility	§ Shells/Terminals Co	omparisons
Key Prefixes & Suffixes		∑ Modifier Keys	∑ <b></b> Numkeypad	Keys - Fn	Keys - F11	Keys - F12	≻PEL
<ul> <li>∑ Emacs Features</li> <li>• A Guided Tour of Emacs.</li> <li>• Awesome-Emacs</li> <li>• MELPA and GNU ELPA</li> </ul>		Cells link titles starting	with only ∑ are Emacs o	eneric features, blue link	s are external packages	. The green links are mo	stly PEL extensions.
		∑ Abbreviations	∑ Diff & Merge	∑ Grep	∑ Marking	∑ Scrolling	∑ Tab Bar
		∑ Align	∑ Dired	∑ Help/Info	∑ Menus	∑ Search/Replace	T Templates
The tables listed at right describe Emacs		∑ Auto-Completion	∑ Display - Lines	∑ Hide/Show	∑ Mode Line	∑ Sessions	∑ Text Modes
commands & key bindings for concepts & features. The cell is light-blue for major mode,		∑ Autosave/Backup	∑ Drawing	∑ Highlight (colors)	∑ Mouse	∑ start Shells/REPLs	∑ Time Tracking
light-red for minor mode specific concepts.  Grey cells are links into other pages for		∑ Bookmarks	∑ Enriched Text	∑ ibuffer-mode	∑ Narrowing	∑ shell-mode	∑ Tramp
important concepts. Emacs commands can be executed by name or		∑ Buffers	∑ Execute Cmds	∑ Indentation	∑ Navigation	∑ term-mode	∑ Transpose text
bound to key sequences. They describe the		∑ Case Conversions	∑ Exec Shell Cmds	∑ Input Method	∑ Object Files	∑ eat-mode	∑X Treemacs
commands, their <u>arguments</u> and the key sequences bound to them.		∑ Close/Suspend	∑ Faces/Fonts	∑ Inserting Text	∑ Outline	∑ vterm-mode	∑ Undo/Redo
• Emacs Keys • Numeric Arguments		∑ Comments	∑P Fast Startup	∑ Key-Chords	∑ Packages	∑X Smartparens	∑ VCS-Git XMagit
You can also:		∑ Completion/Input	∑ File Encoding	∑ Keyboard Macros	∑X Projectile	∑ Sorting	∑ VCS-Mercurial
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.		∑ Counting	∑ File-mngt	Pίχ- Lispy	∑ Rectangles	∑ Speedbar	∑ VCS-Subversion
		<u>∑M CUA</u>	∑ File/Dir Variables		∑ Registers	∑ Spell Checking	∑ Web
		∑ Cursor	∑ Fill/Justify		-	∑ SyntaxCheck	∑ Whitespace
		∑ Customize	∑ Frames				∑ Windows
		∑ Cut & Paste					∑ Xref - Cross Refs
ருடு - Emacs Lisp concepts & tools		<u>≴ display-buffer</u>	<b>≴</b> - ELisp Types	<u>≴ ERT</u> (regr-testing)	<u></u> Hooks		
XRef - Cross Reference Tools See also:   XRef - Cross Reference Tools See		Emacs supports various cross reference mechanisms described in the <u>S</u> Xref table. These mechanisms take advantage of various external					
			_	se tools are available in	the tables listed in this s	ection. Also describes in	•
		Xref-Support	Xref-Frontend	Xref-Backend			Indentation Styles
E-Hdarkt			veral build tools but they nix-mode external pac	vare not all documented	in a page. en <b>pel-use-nix-mode</b> u	ser-ontion is tuned on	
Build Tools & Preprocessor			ckage 🛂 activated wh		•		
Build 100is & Freprocessor		Đῖ - CMake ﷺ	ฆเ - M4	ஆ≀ - Make gmake			
Data Serialization		① CWL	① YAML				
Data Modelling/ Specificati	ion	S ASN.1 asn1-mode	S MIB snmp-mode	S YANG			
Other File Formats		_	RPM Files 🐠	M X.509 Certificates			
		Verilog 🚧	VHDL ##				
Hardware Description Langua	<u>ages</u>		' '				
<u>Lightweight Markup Languag</u>	<u>jes</u>	<u>M AsciiDoc</u>	<u>M Markdown</u>	M Org-Mode	<u>M</u> reStructuredText		OS App Control Scripting Languages
Graphics Markup		M Graphviz Dot	MscGen	<u>M PlantUML</u>			⊉≀∉- AppleScript
Programming Languages		Emace has major mode	support for several pro	gramming languages. P	El extende Emace cunn	port for some of them (at	pers are marked (***)
Main Paradigm of Programming La Families	anguage	BEAM Programming	Functional	Javascript target	Lisp Family	Lisp-like Languages	icis are marked maj.
• Actor Model: (A)		<u>Languages</u>	Languages	Javascript target	Languages	Lisp-like Languages	
Concatenative (©     Concurrent: ©		Curly Bracket	Java Virtual Machine	ML Family	Scheme Language	Stack Based	
Domain Specific	l	Languages	Languages	Languages	<u>Dialects</u>	Languages	
• Dynamic & • Functional: (f) Pure: (F)			he programming languag		mr lenet 0000	Oktober O šmå	Soolo ini
• Imperative: (i) or no token		Ada 🚧	<u><b>B</b>L-D</u> () () (A)	PI - Gambit 🗇	<u>βί - Janet</u> ①fm	Objective-C ##	Scala 🚧
Object Oriented       Procedural		<u>PI - Arc</u> ①	Dart ##	PI - Gerbil (T) (A)	Java ##	<u>Pi - OCaml</u> if	PI - Scheme (fm)
Has <u>Syntactic Macros</u> :		<u>βι - awk</u>	Eiffel 🚧 🔞	BI - GNU Guile fm	भा - Javascript 🚧	Pascal ##	Seed7 ##
• System Level 🕲		<u>\$\pi - C</u> ⊗	pῖ - Elm 🗯 🕞	<b>Ֆ</b> ῖ - Gleam	<u>β</u> ῖ - Julia @	<b><u>B</u>l - Perl</b> ( <u>perl5</u> )	Swift ##
The programming languages support		<u>₩ - C++</u> @8	<u>PI - Elixir</u> ©®€A	<u>βρί - Go</u> Θ	Kotlin 🚧	PI-Python dPOF	pt - Tcl ₩ fi
<ul> <li>PEL are listed here in alphabetical of Emacs (and PEL) also provides based</li> </ul>		<u>MI-Chez</u> fm	₹PI - Emacs Lisp	Groovy 🚧	<u>pι-lfe</u> ©mfA		អ្រ - Typescript 🚧
for other programming languages n		<u>Bl - Chibi</u> fm	<u>at - Erlang</u> ©FA	<b>β</b> Ι - Haskell ⑤	Lua 🚧	<u>ֆι - Racket</u> ∱®	βί - UNIX Shell
		<b>Bl-Chicken</b>	Factor (S) © (9)	Haxe 🚧	Modula 🚧	ា្រ្	<u> р V</u>
Future support for Crystal, Elm, Ko Purescript, ReasonML, Seed7, Typeso	cript, Zig	<u>pι - Clojure</u> fm	<u>B</u> ℓ - Forth ®	<u>ൂi - Hy</u> (python) ™	រា្ធរ - NetRexx	ֆῖ - REXX	Zig 🗯 🕒
and documentation of support for Ada, Fortran, Javascript, Java, Modula, Pascal (based on my		Common Lisp fm	Fortran 🚧		<u>ֆ≀ - Nim</u> @⊗	ា្ន្រ - Ruby	
need for them or requests)	,	On and all and				my Duet O	