PEL Topics Index

	Last updated on: 2024-10-08			Note: with PEL, type <f11> <f1> to open this PDF index.</f1></f11>		
Emacs Reference Cards	These are links to the PDF version of official English version of the quick reference cards for GNU Emacs and popular external packages.					
With PEL you can access these via the			ese cards provide usefu		1 -	
<f11> ? e r key sequence. See ∑ Help/Info</f11>	Emacs	Calc	Gnus	Magit Cheatsheet	Org	<u>Viper</u>
	Emacs survival card	Dired	Gnus booklet	Magit Ref-card		VIP
 PEL Overview (license) PEL repo 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. 						
PEL Readme	 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 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.</f11></f1></f11> 					
• PEL Manual • PEL NEWS						
• <u>Discussions</u>	The symbols, colour coding and various other conventions are described in the <u>>Legend</u> PDF.					<u></u>
General Information.	<u>≻Legend</u>	≻Recommended Ema	acs User Option	<u>≻Themes</u>	Migrate from CRiSP	
Startup		Run Emacs daemon 8	k clients É 🐠	iMenu/Speedbar su	<u>upport</u>	
Development Information	<u>≻PEL</u>	PEL Naming Conve	entions	PEL Environment V	/ariables	PEL utilities
OS Desktop Key Bindings (Bindings that don't clash with PEL)	≰ macOS Fct Keys	≰ macOS Keys				
	w macoo i ct keys		Mint 20 Desktop Ko	<u>eys</u>	MUbuntu 16.04 Desk	top Keys
		≰ terminal settings	Nocky Linux 8 Des	ktop Keys		
Feature Comparisons	₿ Completion Modes	Compatibility	§ Speedbar/iMenu N	Mode Compatibility	§ Shells/Terminals Co	omparisons
Key Prefixes & Suffixes	∑ Modifier Keys ∑ Numkeypad		<u>≻PEL</u>	Keys - Fn	Keys - F11	
∑ Emacs Features	Cells link titles starting	with only $\mathbb Z$ are Emacs g	eneric features, blue link	s are external packages	s. The green links are mo	stly PEL extensions.
A Guided Tour of Emacs. Awesome-Emacs MELPA and GNU ELPA	∑ 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. Emacs commands can be executed by name	∑ Bookmarks	∑ Enriched Text	∑ ibuffer-mode	∑ Narrowing	∑ shell-mode	<u>∑ Tramp</u>
or bound to key sequences. They describe	<u>∑ Buffers</u>	∑ Faces/Fonts	∑ Indentation	∑ Navigation	<u> ▼ term-mode</u>	∑ Transpose text
the commands, their <u>arguments</u> and the key sequences bound to them.	∑ Case Conversions	∑P Fast Startup	∑ Input Method	∑ Object Files	∑ eat-mode	∑X Treemacs
• Emacs Keys • Numeric Arguments	∑ Close/Suspend	∑ File Encoding	∑ Inserting Text	∑ Outline	<u>∑ vterm-mode</u>	∑ Undo/Redo
You can also: Run Command by Name	∑ Comments	∑ File-mngt	∑ Key-Chords	∑ Packages	<u>∑X Smartparens</u>	∑ VCS-Git XMagit
Emacs uses a concept of modes:	∑ Completion/Input	∑ File/Dir Variables	∑ Keyboard Macros	<u>∑X Projectile</u>	∑ Sorting	∑ VCS-Mercurial
Emacs Major and Minor Modes Major Modes Minor Modes	∑ Counting	∑ Fill/Justify	<u>ptx- Lispy</u>	∑ Rectangles	∑ Speedbar ∑ Spell Checking	∑ VCS-Subversion
	<u>∑M CUA</u> <u>∑ Cursor</u>	∑ Frames		∑ Registers	∑ SyntaxCheck	∑ Web ∑ Whitespace
 Choosing Modes PEL provides several key sequences to toggle 	∑ Customize				<u>z cymaxoneck</u>	∑ Windows
minor modes.	∑ Cut & Paste					∑ Xref - Cross Refs
±n - Emacs Lisp concepts & tools	± display-buffer	<u> 1× - ELisp Types</u>	<u>★ ERT</u> (regr-testing)	⊈ Hooks		
XRef - Cross Reference Tools	, ,	, ,	, , , , , , , , , , , , , , , , , , ,	-	Lanisms take advantag	e of various external
XRef - Cross Reference Tools See also: Xref Emacs supports various cross reference mechanisms described in the Xref tools and integrate with them. Notes about those tools are available in the tables listed in this section.						
	Xref-Support	Xref-Frontend	3 Xref-Backend			
PEL supports installation and partial setup of	PEL has support for se	veral build tools but they	are not all documented	in a page.		Command Line
the following tools: • Nix • Requires nix-mode external package i activated when pel-use-nix-mode user-option is tuned on. Languages:						
Build Tools & Preprocessor * Tup Requires tup-mode external package activated when pel-use-tup user-option is tuned on. bash, sh, zsh						
	<u>ൂĭ - CMake</u> ‱future	<u> ұй - М4</u>	<u>Pl - Make</u> gmake			
Data Serialization	© CWL	<u> </u>				Utility: GNU readline
Data Modelling/ Specification	S ASN.1 asn1-mode	S MIB snmp-mode	<u>©</u> YANG			<u>ls -l</u>
Other File Formats		RPM Files	M X.509 Certificates			
Hardware Description Languages	Verilog ##future	VHDL ##future				
Text Markup Languages	M AsciiDoc	M Markdown	M Org-Mode	M reStructuredText		OS App Control
						Scripting Languages
Graphics Markup	M Graphviz Dot	M MscGen	<u>M PlantUML</u>			<u>nte- AppleScript</u>
Programming Languages Main Paradigm of Programming Language	Emacs has major mode	e support for several pro	gramming languages. P	EL currently adds extra	support for some of ther	n, listed below.
Families	BEAM Programming Languages	Functional Languages	Javascript target	Lisp Family Languages	Lisp-like Languages	
 Actor Model: (A) Concatenative (K) 	Curly Bracket	Java Virtual Machine	ML Family	Scheme Language	Stack Based	
· Concurrent: ©	Languages	Languages	Languages	<u>Dialects</u>	Languages	
 Functional: f Pure: F Imperative: î or no token 			1		rogramming language far	
• Object Oriented ∞	Ada ##future		<u>βι - Gambit</u> fm	<u>nu - Janet</u> infm	Objective-C #future	Scala ##future
• Has <u>Syntactic Macros</u> : 🕅	Bt - Arc fm	Dart ###future	<u>B</u> ũ - Gerbil ∱®A	Java ##future	<u>βι - OCaml</u> if	PI - Scheme fm
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.	<u> 1βί - C</u>	Eiffel ##future	B	រុរ្ - Javascript ﷺ	Pascal ##future	Seed7 future
	<u> </u> <u> Φ</u>	pt - Elm ∰future ♠	<u></u> βι - Gleam	<u>aμτ - Julia</u> m	<u></u> βι - Perl	Swift ##future
	<u>aβt - Chez</u> ∱m	<u>al - Elixir</u> cmfA	<u> 1</u> βί - Go	Kotlin future	្ប - Python	β ũ - Tcl ₩ future (f) i
	<u>βι - Chibi</u> fm	քֆն - Emacs Lisp	Groovy ##future	BI-LFE COTA	p	ֆῖ - Typescript ##
Future support for Crystal, Elm, Kotlin, Lua,	<u>nu - Chicken</u> fm	<u>βι - Erlang</u> ©(f)A	β ῖ - Haskell ⑤	Lua ##future	<u>nu - Racket</u> fm	ֆῖ - UNIX Shell
Purescript, ReasonML, Seed7, Typescript, Zig and documentation of support for Ada,	乳I - Clojure	Factor (K) f ∞(m)	Haxe #future	Modula ##future	ា្រ្	<u> 1βί - V</u>
Fortran, Javascript, Java, Modula, Pascal (based on my need for them or requests (if	Common Lisp fm	<u>au - Forth</u> €	<u>ൂ≀ - Hy</u> (python) ®	ន្ទរ - NetRexx	ֆῖ - REXX	Zig #future
(based on my need for them or requests (if any)).	Crystal ##future	Fortran ###future		3 Σ - Nim	ֆῖ - Ruby	
					®[- Rust	