PEL Topics Index

Emacs Reference Cards				k reference cards for GN Il complement to what P		external packages.
With PEL you can access these via the <f11> ? e r key sequence.</f11>	Emacs	Calc	Gnus	Magit Cheatsheet	Org	Viper
See <u>∑ Help/Info</u>	Emacs survival card	Dired	Gnus booklet	Magit Ref-card		VIP
PEL repo PEL Readme PEL Manual	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. From within Emacs open this topic index PDF by typing the <f11>? <f1> key sequence. The symbols, colour coding and various other conventions are described in the ➤ Legend PDF.</f1></f11>					
					DF.	
General Information.	<u>≻Legend</u>	➤ Recommended Emacs User Option		<u>≻Themes</u>		
Development Information	<u>>PEL</u>	iMenu/Speedbar support		PEL Naming Conventions		
Migration Guide	<u>>CRiSP </u>					
macOS Specific	≰ macOS Keys	≰ terminal settings				
Feature Comparisons						
o i catale compansons	Completion Modes Compatibility Speedbar/iMenu Modes			Mode Compatibility		
Key Prefixes & Suffixes						
	<u> ∑ </u>		<u>∑</u> Numkeypad	≻PEL	<u> ■Keys - Fn</u>	<u>■Keys - F11</u>
Emacs Features	The links that start with	n only ∑ Emacs generic	features, the blue links a	re external packages. Th	ne green links are mostly	PEL extensions.
Fhese PEL tables describe the Emacs	∑ Abbreviations	<u>∑ Cursor</u>	∑ Filling/	<u> ֆῖ- Lispy</u>	<u>∑ Scrolling</u>	∑x Treemacs
commands and key bindings for generic concepts and features.	W 41.		<u>Justification</u>	W		T. 1
	<u></u> <u>X Align</u>	<u> ∑ Customize</u>	<u></u> Frames	<u></u> Marking	∑ Search/Replace	∑ Undo/Redo/ Repeat/Arg
Emacs uses a concept of modes.	∑ Auto-Completion	∑ Cut & Paste	<u></u> Grep	<u></u> Menus	∑ Semantic	∑ VCS-Git XMagit
See: Emacs Major and Minor Modes	∑ Autosave/Backup	∑ Diff & Merge	∑ Help/Info	∑ Mode Line	∑ Sessions	∑ VCS-Mercurial
Major Modes Minor Modes Choosing Modes PEL provides several key sequences to toggle minor modes, described in the relevant PDFs.	<u></u> Bookmarks	<u></u> Dired	∑ Hide/Show	<u></u> Mouse	∑ Shells , REPLs &	<u></u> Web
					terminal emulators	
	<u>» Buffers</u>	∑ Display - Lines	<u></u> Highlight	Narrowing	<u></u> Sorting	<u> ▼ Whitespace</u>
	<u> ∑ Case Conversions</u>	<u></u> Drawing	<u>∑ ibuffer-mode</u>	<u> </u>	<u></u> Speedbar	<u> ▼ Windows</u>
Emacs commands can be executed by name or bound to key sequences. The commands may have arguments and keys can express them. See: • Emacs Keys	∑ Closing/ Suspending	∑ Enriched Text	∑ Indentation	<u>» Outline</u>	∑ Spell Checking	<u>∑ Xref</u> - Cross References
	<u>∑ Comments</u>	∑ Faces/Fonts	∑ Input Method	<u> ▼ Packages</u>	∑ SyntaxCheck	
	∑ Completion/Input	<u> ∑P Fast Startup</u>	∑ Inserting Text	∑x Projectile	T Templates	
	<u></u> ∑ Counting	<u></u> File-mngt	<u> </u>	<u> </u>	<u> ▼ Text Modes</u>	
	<u>∑M CUA</u>	∑ File/Directory Variables	∑ Keyboard Macros	<u> </u>	<u> ∑ Transpose</u>	
£%ு - Emacs Lisp concepts & tools	<u>⊈ ERT</u>	<u></u> <u>⊀ Hooks</u>		es		
XRef - Cross Reference	Emacs supports various cross reference mechanisms described in the <u>Xref</u> table. These mechanisms take advantage of various external					
Tools				the tables listed in this s		
	3 Xref-Support	Xref-Backend				
Build Tools	PEL has support for several build tools but they are not all documented in a page. Aside from the list below, PEL supports installation and partial setup of the following tools: Nix Requires nix-mode external package					
		s tup-mode external pa	_ ·	when pel-use-tup user-	•	
	भ्रा - Make					
Data Serialization Languages	© CWL	<u>© YAML</u>				
Markup Languages	M AsciiDoc	M Graphviz Dot	M Markdown	M Org-Mode	M PlantUML	M reStructuredTex
Programming Languages				adds extra support for s		
Main Paradigm of Programming	The number of progr	ramming languages supp	ported explicitly by PEL	will grow over time.		
.anguage Families • Actor Model: (A) • Concatenative (K)	BEAM Programming Languages	<u>Functional</u> <u>Languages</u>	Javascript target	Lisp Family Languages	Stack Based Languages	Command Line Scripting Language
· Concurrent: ©	Curly Bracket Languages	Java Virtual Machine Languages	ML Family Languages	Scheme Language Dialects		OS App Control Scripting Language
 Functional: Pure: Imperative: or no token The programming languages supported by PEL are listed here in alphabetical order. 	The following lists the programming languages in alphabetical order. • The cell colours give a coarse indication of the programming language family(ies).					
	માં € cell colours give	\$1 - Clojure f	38i - Forth	Bι - Hy	βĭ - Perl	भ्रा - Rust
PEL also provides basic support	PI - Arc f			มี - Javascript	भ्रा - Python	भूर - Nust
for other programming languages not listed here. • Emacs supports other programming languages directly, not listed here. Upcoming support for Elm, Purescript, ReasonML, Typescript and documentation of support for Javascript.				• • • • • • • • • • • • • • • • • • • •		
	<u> </u>	<u>BI-D</u> (IFA)	BI - Gerbil (FA)	βι - Julia	भ्रा - Purescript 🕞	
	<u> βί - C++</u>	βI - Elm F	क्षा - GNU Guile 🕥			
	<u>βι - Chez</u> f	<u>Bt - Elixir</u> © (f) A	鸦ῦ - Gleam	乳ῖ - NetRexx	भृर - ReasonML	<u>₽1 - V</u>
	<u>apı - Chibi</u>	<u>≭्रभा - Emacs Lisp</u>	<u>βί - Go</u>	Bῖ - Nim	Bι - REXX	