## **PEL Topics Index**

	Last updated on: 2024-09-29			Note: with PEL, type <f11> <f1> to open this PDF index.</f1></f11>		
Emacs Reference Cards					U Emacs and popular e	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>S Help/Info</u>	Emacs survival card	Dired	Gnus booklet	Magit Ref-card	<u>Olg</u>	VIP
➤ PEL Overview  • PEL repo • PEL Readme • PEL Manual • PEL NEWS  • Discussions	This table helds links to the DEL file tables. Each call helds a hunswink to the Citi link heated your DDE table.					
General Information.	<u>≻Legend</u>	≻Recommended Ema	acs User Option	<u>≻Themes</u>		
Development Information	<u>▶PEL Naming Converted Naming Converted</u>		entions PEL Environment V		<u>'ariables</u>	PEL utilities
Migration Guide	>CRISP → Emacs	Menu/Speedbar support		Run Emacs daemon & clients		
OS Desktop Key Bindings (Bindings that don't clash with PEL)	<b>★ macOS Fct Keys</b> ★ macOS Keys		Mint 20 Desktop Ke	<u>eys</u>	<b>⊕</b> Ubuntu 16.04 Desktop Keys	
		terminal settings Procky Linux 8 Desktop Keys				
	<b>♣</b> Completion Modes	Compatibility	♣ Speedbar/iMenu N	Ande Compatibility	₿ Shells/Terminals Co	omparisons
Feature Comparisons  Kev Prefixes & Suffixes		Compatibility				
	<u> </u>		<u>∑</u> Numkeypad	<u>≻PEL</u>	Keys - Fn	Keys - F11
<ul> <li>Emacs Features</li> <li>A Guided Tour of Emacs.</li> <li>Awesome-Emacs</li> <li>MELPA and GNU ELPA</li> </ul>		_			s. The green links are mo	
	∑ Abbreviations	∑ Diff & Merge	<u>∞ Grep</u>	∑ Marking	∑ Scrolling	∑ Tab Bar
	<u>∑ Align</u>	∑ Dired	∑ Help/Info	<u> ∑ Menus</u>	∑ Search/Replace	<u>T Templates</u>
The tables listed at right describe Emacs commands & key bindings for concepts &	∑ Auto-Completion	∑ Display - Lines	∑ Hide/Show	∑ Mode Line	∑ Sessions	∑ Text Modes
features. The cell is light-blue for major mode,	∑ Autosave/Backup	∑ Drawing	∑ Highlight (colors)	<u>∑ Mouse</u>	∑ 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	∑ term-mode	∑ 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	<b>∑</b> X Treemacs
• Emacs Keys • Numeric Arguments	∑ Close/Suspend	∑ File Encoding	∑ Inserting Text	∑ Outline	∑ vterm-mode	∑ Undo/Redo
You can also: Run Command by Name	∑ Comments	∑ File-mngt	∑ Key-Chords	∑ Packages	<b>∑</b> X Smartparens	∑ VCS-Git XMagit
	∑ Completion/Input	∑ File/Dir Variables	∑ Keyboard Macros	<b>∑</b> X Projectile	∑ Sorting	VCS-Mercurial
Emacs uses a concept of modes:     Emacs Major and Minor Modes	∑ Counting	∑ Fill/Justify	<u>Pίχ- Lispy</u>	∑ Rectangles	∑ Speedbar	VCS-Subversion     VCS-
<ul><li>Major Modes</li><li>Minor Modes</li></ul>	<u>∞M CUA</u>	<u>∑ Frames</u>		<u> </u>	∑ Spell Checking	<u> ℤ Web</u>
Choosing Modes	<u>∑ Cursor</u>				∑ SyntaxCheck	Whitespace     Whitespace
PEL provides several key sequences to toggle minor modes.	<u>∑ Customize</u>					<u>∑ Windows</u>
	∑ Cut & Paste					<u>∑ Xref</u> - Cross Refs
វូស្ត្រ - Emacs Lisp concepts & tools	<u>≴ display-buffer</u>	<b>≴</b> - ELisp Types	<u>★ ERT</u> (regr-testing)	<u>≴ Hooks</u>		
XRef - Cross Reference Tools See also:   Xref	Emacs supports various cross reference mechanisms described in the X Xref table. These mechanisms take advantage of various external tools and integrate with them. Notes about those tools are available in the tables listed in this section.					
	Xref-Support	Xref-Frontend				
PEL supports installation and partial setup of the following tools: <b>Build Tools &amp; Preprocessor</b>	• Nix Pequires nix-mode external package activated when pel-use-nix-mode user-option is tuned on.					Command Line Scripting Languages: bash, sh, zsh
	<u><b>Ֆῖ - CMake</b></u> ‱future	<u> ұт - М4</u>	<b>Şũ - Make</b> gmake			
Data Serialization	© CWL	① YAML				Utility: GNU readline
Data Modelling/ Specification	S ASN.1 asn1-mode	S MIB snmp-mode	© YANG			ls -l
Hardware Description Languages	Verilog ##future	VHDL ##future	<u> </u>			
	•		M Over Made	M wa Churrah wa dTaut		00 4 0
Text Markup Languages	M AsciiDoc	<u>M Markdown</u>	<u>M Org-Mode</u>	<u>M</u> reStructuredText		OS App Control Scripting Languages
Graphics Markup	<u> </u>	<u> МscGen</u>	<u> М PlantUML</u>			⊉෭ <b>க்- AppleScript</b>
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.
Main Paradigm of Programming Language Families  • Actor Model: (A)  • Concatenative (K)  • Concurrent: ©	BEAM Programming Languages	Functional Languages	Javascript target	Lisp Family Languages	Lisp-like Languages	
	Curly Bracket Languages	Java Virtual Machine Languages	ML Family Languages	Scheme Language Dialects	Stack Based Languages	
• Functional: f Pure: f	The following lists the p	programming languages	in alphabetical order. Co	ell colours refer to the pr	ogramming language far	mily(ies).
<ul> <li>Imperative: ① or no token</li> <li>Object Oriented ∞</li> <li>Has Syntactic Macros: ⑪</li> </ul>	Ada ##future	PI-D ()(f)A	<b>β</b> ℓ - Gambit fm	<u>aβι - Janet</u> (i)⊕m	Objective-C ##future	Scala ## future
	<u> pi - Arc</u> fm	Dart ##future	<u>ൂu - Gerbil</u> ∱mA	Java ##future	pι - OCaml if	pĭ - Scheme ∱m
	<b>₽</b> [ - C	Eiffel ##future	Bt - GNU Guile ∱®	®เ - Javascript ﷺ	Pascal #future	Seed7 #future
<ul> <li>The programming languages supported by PEL are listed here in alphabetical order.</li> <li>Emacs (and PEL) also provides basic support for other programming languages not listed here.</li> </ul>		- 11122	Pĭ - Gleam	ង្គរ - Julia	BI - Perl	Swift ##future
	βι - Chez ∱m	BI - Elixir ©@fA	BI - Go	Kotlin #future	ន្ទរ - Python	®I - Tcl ∰future (f)(i)
			<u> </u>			
	P  Ω  Chibi  T  T  T  T  T  T  T  T  T  T  T  T  T	<u> </u>	Groovy future	PI-LFE CMTA	भूर - Purescript 🕞	段ℓ - Typescript ##
Future support for Crystal, Elm, Kotlin, Lua, Purescript, ReasonML, Seed7, Typescript, Zig and documentation of support for Ada, Fortran, Javascript, Java, Modula, Pascal (based on my need for them or requests (if any)).	PII - Chicken 🗇	<u>βί - Erlang</u> CfA	PI - Haskell F	Lua #future	PI - Racket fm	野ǔ - UNIX Shell
	<u>βι - Clojure</u> fm	Factor (Cfoom	Haxe ##future	Modula ##future	ֆῖ - ReasonML ﷺ	<u> </u> βι - V
	Common Lisp fm	<u>B</u> ῖ - Forth (€)	<u>ൂi - Hy</u> (python) ₪	<u>βι - NetRexx</u>	<u>aμι - REXX</u>	Zig ##future
	Crystal ##future	Fortran ##future		<b>β</b> Ι - Nim	្សĭ - Ruby	
					क्षा - Rust	