PEL Topics Index

| Emacs Reference Cards | | PDF version of official En | | | | external packages. | |
|--|---|--|--|--|--|--|--|
| With PEL you can access these via | | s key bindings as well, th | · - | | <u>'</u> | Vinor | |
| the <f11> ? e r key sequence. See <u>∑ Help/Info</u></f11> | Emacs Emacs survival card | Dired | Gnus Gnus booklet | Magit Cheatsheet Magit Ref-card | Org | <u>Viper</u> | |
| >> PEL Overview | | o the PEL tables. Each o | | | DF table. | | |
| > 1 LL OVEI VIEW | For the best user ex | perience, use a browser | that can render PDF dire | ectly instead of downloa | | | |
| PEL repoPEL Readme | Firefox does that. 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. | | | | | | |
| PEL Manual | | open this topic index PD | | | | | |
| General Information. | | r coding and various other | | Tibed in the <u>≻Legend</u> Pi ≻Themes | JF. | | |
| Development Information | <u>>Legend</u> >PEL | iMenu/Speedbar su | <u> </u> | PEL Naming Conventions | | | |
| Migration Guide | >CRiSP ≈ Emacs | iwenu/Speedbar st | арроги | PEL Naming Conve | enuons | | |
| - Iviigration duide | > Onior - Linacs | | | | | | |
| | ≰ macOS Keys | ≰ terminal settings | | | | | |
| Feature Comparisons | | | | | | | |
| | ∄ Completion Modes Compatibility ∄ Speedbar/iMenu Mode Compatibility ∄ S | | | | § Shells/Terminals C | omparisons | |
| Key Prefixes & Suffixes | | | | | | | |
| | <u> ∑ </u> | | <u>∑</u> Numkeypad | <u>≻PEL</u> | <u>Keys - Fn</u> | <u>Keys - F11</u> | |
| ∑ Emacs Features | The links that start with | n only ∑ are built-in Emad | cs, the links that are blue | e are external packages. | | | |
| These PEL tables describe the Emacs | ∑ Abbreviations | <u></u> M CUA | <u></u> File-mngt | <u></u> <u>Xey-Chords</u> | <u> ∑ Projectile</u> | ∑ SyntaxCheck | |
| commands and key bindings for generic concepts and features. | <u>∑ Align</u> | <u>∑ Cursor</u> | ∑ File/Directory Variables | ∑ Keyboard Macros | <u> </u> | <u>T Templates</u> | |
| | ∑ Auto-Completion | ∑ Customize | ∑ Filling/ | 致ι- Lispy | <u></u> Registers | <u> ∑ Text Modes</u> | |
| Emacs uses a concept of modes. See: | | <u>// Customize</u> | Justification | | | | |
| Emacs Major and Minor Modes Major Modes | ∑ Autosave/Backup | ∑ Cut & Paste | <u>∑ Frames</u> | <u></u> Marking | <u></u> Scrolling | <u>> Transpose</u> | |
| Minor ModesChoosing Modes | <u> ∑ Bookmarks</u> | <u>∑ Diff & Merge</u> | <u>∑ Grep</u> | <u>∑ Menus</u> | ∑ Search/Replace | ∑x Treemacs | |
| PEL provides several key sequences to toggle minor modes, described in | <u></u> <u>Buffers</u> | <u></u> Dired | ∑ Help/Info | <u> Mode Line</u> | ∑ Semantic | <u></u> <u>Undo/Redo/</u> <u>Repeat/Arg</u> | |
| the relevant PDFs. | ∑ Case Conversions | ∑ Display - Lines | ∑ Hide/Show | <u></u> Mouse | ∑ Sessions | ∑ VCS-Mercurial | |
| Emacs commands can be executed by name or bound to key sequences. | ∑ Closing/ | <u></u> Drawing | <u></u> Highlight | Narrowing | <u>Shells</u> , REPLs & | <u></u> Web | |
| The commands may have arguments and keys can express them. | Suspending | | X 11 66 | W-A1 | terminal emulators | <u>~</u> 14# · | |
| See: • Emacs Keys | ∑ Comments ∑ Completion/Input | ∑ Enriched Text ∑ Faces/Fonts | ∑ ibuffer-mode ∑ Indentation | Navigation Outline | ∑ Sorting ∑ Speedbar | ∑ Whitespace ∑ Windows | |
| Lillaus ricys | ∑ Counting | | ∑ Inserting Text | ∑ Packages | ∑ Spell Checking | ∑ Xref - Cross | |
| | <u>z counting</u> | <u>// raot otartap</u> | Z moorting roxe | <u>// Tuokugoo</u> | Z Opon Oncoming | References | |
| ⊈₽ῖ - Emacs Lisp concepts & tools | <u>≴ ERT</u> | <u>≴ Hooks</u> | ±* - Emacs Lisp Type | <u>98</u> | | | |
| XRef - Cross Reference | Emacs supports variou | Emacs supports various cross reference mechanisms described in the <u>Natural Xiref</u> table. These mechanisms take advantage of various external | | | | | |
| Tools | tools and integrate with them. Notes about those tools are available in the tables listed in this section. This is work in progress. | | | | | | |
| | | | se tools are available in | | | in progress. | |
| | Xref-Support | Xref-Backend | se tools are available iii | the tables listed in this s | | in progress. | |
| Build Tools | Xref-Support PEL has support for se | Xref-Backend everal build tools but they | are not all documented | in a page. | | in progress. | |
| Build Tools | PEL has support for se Aside from the list belo | Xref-Backend | are not all documented tion and partial setup of | in a page. | user-option is tuned or | | |
| Build Tools | PEL has support for se Aside from the list belo Nix Requires | Xref-Backend everal build tools but they www, PEL supports installar | vare not all documented tion and partial setup of kage | in a page. the following tools: | • | | |
| Build Tools | PEL has support for se Aside from the list belo Nix Requires | Everal build tools but they bw, PEL supports installar s nix-mode external pace | vare not all documented tion and partial setup of kage | in a page. the following tools: when pel-use-nix-mode | • | | |
| Data Serialization | PEL has support for se Aside from the list belo Nix Requires Tup Requires PI - Make | Exernal build tools but they bw, PEL supports installates nix-mode external pacts tup-mode external pacts. | vare not all documented tion and partial setup of kage | in a page. the following tools: when pel-use-nix-mode | • | | |
| Data Serialization Languages | PEL has support for se Aside from the list belo Nix Requires Tup Requires | Everal build tools but they bw, PEL supports installar s nix-mode external pace | vare not all documented tion and partial setup of kage | in a page. the following tools: when pel-use-nix-mode | • | | |
| Data Serialization Languages | PEL has support for se Aside from the list belo Nix Requires Tup Requires PI - Make | Exernal build tools but they bw, PEL supports installates nix-mode external pacts tup-mode external pacts. | vare not all documented tion and partial setup of kage | in a page. the following tools: when pel-use-nix-mode | • | | |
| Data Serialization Languages Markup Languages Programming Languages | PEL has support for se Aside from the list belo Nix Requires Tup Requires PI - Make C CWL M AsciiDoc Emacs has support for | Exeral build tools but they by, PEL supports installar is nix-mode external pact is tup-mode ext | v are not all documented tion and partial setup of kage activated activated activated with a comparation of the comparation of | in a page. the following tools: when pel-use-nix-mode when pel-use-tup user- | option is tuned on. My PlantUML | <u>M</u> reStructuredText | |
| Data Serialization Languages Markup Languages Programming Languages Main Paradigm of Programming Language Families | ■ Xref-Support PEL has support for se Aside from the list belo • Nix | Wereal build tools but they by, PEL supports installar is nix-mode external pacts is tup-mode external pacts tup-mode external pacts. DYAML Graphviz Dot several programming lar ramming languages supports. | v are not all documented tion and partial setup of kage activated activated activated with a comparation of the comparation of | in a page. the following tools: when pel-use-nix-mode when pel-use-tup user- | option is tuned on. My PlantUML | <u>M</u> reStructuredText | |
| Data Serialization Languages Markup Languages Programming Languages Main Paradigm of Programming | ■ Xref-Support PEL has support for se Aside from the list belo • Nix | Wref-Backend Everal build tools but they buy, PEL supports installar s nix-mode external pact s tup-mode external pact D YAML M Graphviz Dot Several programming lar ramming languages supp Functional Languages | ware not all documented tion and partial setup of kage activated was act | in a page. the following tools: when pel-use-nix-mode when pel-use-tup user- M Org-Mode adds extra support for s will grow over time. Lisp Family Languages | M PlantUML ome of them, listed belo | M reStructuredText W. Command Line Scripting Languages | |
| Data Serialization Languages Markup Languages Programming Languages Main Paradigm of Programming Language Families • Actor Model: (A) • Concatenative (K) • Concurrent: (©) | PEL has support for se Aside from the list belo Nix Requires Tup Requires BI - Make D CWL M AsciiDoc Emacs has support for The number of progr | Exertal build tools but they by, PEL supports installar is nix-mode external pacts tup-mode external pacts tup-mode external pacts tup-mode external pacts tup-mode external pacts for the pacts of the pacts of the pacts of the packs of the | y are not all documented tion and partial setup of kage activated where the chage activated where the chage activated where the chage activated ac | in a page. the following tools: when pel-use-nix-mode when pel-use-tup user- M Org-Mode adds extra support for s will grow over time. Lisp Family | M PlantUML ome of them, listed belo | <u>M</u> reStructuredText w. Command Line | |
| Data Serialization Languages Markup Languages Programming Languages Main Paradigm of Programming Language Families • Actor Model: • Concatenative • Concurrent: • Functional: • Imperative: • or no token | ■ Xref-Support PEL has support for se Aside from the list belo • Nix | Wref-Backend Everal build tools but they buy, PEL supports installar is nix-mode external pact is tup-mode external pact is sup-mode external pact i | ware not all documented tion and partial setup of kage activated was act | in a page. the following tools: when pel-use-nix-mode when pel-use-tup user- M Org-Mode adds extra support for s will grow over time. Lisp Family Languages Scheme Language Dialects | M PlantUML ome of them, listed belo | M reStructuredText W. Command Line Scripting Languages OS App Control | |
| Data Serialization Languages Markup Languages Programming Languages Main Paradigm of Programming Language Families • Actor Model: (A) • Concatenative (K) • Concurrent: (C) • Functional: (T) Pure: (F) • Imperative: (T) or no token • The programming languages supported by PEL are listed here in | ■ Xref-Support PEL has support for se Aside from the list belo • Nix | Exeral build tools but they by, PEL supports installar is nix-mode external pact is tup-mode external pact is exertal programming languages support in the pact is t | ware not all documented tion and partial setup of kage activated ockage ockage activated ockage activated ockage activated ockage ockage ockage activated ockage o | in a page. the following tools: when pel-use-nix-mode when pel-use-tup user- M Org-Mode adds extra support for s will grow over time. Lisp Family Languages Scheme Language Dialects ge family(ies). | M PlantUML ome of them, listed belo | M reStructuredText W. Command Line Scripting Languages OS App Control Scripting Languages | |
| Data Serialization Languages Markup Languages Programming Languages Main Paradigm of Programming Language Families • Actor Model: (A) • Concatenative (K) • Concurrent: (C) • Functional: (T) • Imperative: (T) or no token • The programming languages supported by PEL are listed here in alphabetical order. • PEL also provides basic support | ■ Xref-Support PEL has support for se Aside from the list belo Nix Requires Tup Requires PEL has support for se Aside from the list belo Nix Requires Requires PEL has support for se Aside from the list belo Make D CWL M AsciiDoc Emacs has support for The number of progress Emacs has support for The number of progress Curly Bracket Languages The following lists the period of the cell colours give PEL has support for services the period of the cell colours give | Exeral build tools but they by, PEL supports installar is nix-mode external pacts tup-mode external pacts support in the pacts of the pa | Markdown Markdo | in a page. the following tools: when pel-use-nix-mode when pel-use-tup user- M Org-Mode adds extra support for s will grow over time. Lisp Family Languages Scheme Language Dialects ge family(ies). NI - Hy | M PlantUML ome of them, listed belo Stack Based Languages | M reStructuredText W. Command Line Scripting Languages OS App Control Scripting Languages | |
| Data Serialization Languages Markup Languages Programming Languages Main Paradigm of Programming Language Families • Actor Model: (A) • Concatenative (K) • Concurrent: (©) • Functional: (†) Pure: (E) • Imperative: (†) or no token • The programming languages supported by PEL are listed here in alphabetical order. | ■ Xref-Support PEL has support for se Aside from the list belo • Nix | Exeral build tools but they by, PEL supports installar is nix-mode external pacts tup-mode external pacts tup-mode external pacts tup-mode external pacts tup-mode external pacts support in the pact | A are not all documented tion and partial setup of kage | in a page. the following tools: when pel-use-nix-mode when pel-use-tup user- when pel-use-tup user- y Org-Mode adds extra support for swill grow over time. Lisp Family Languages Scheme Language Dialects ge family(ies). \$\text{y}\tau - Hy \$\text{y}\tau - Javascript} | M PlantUML ome of them, listed belo Stack Based Languages PI - Perl PI - Python | M reStructuredText W. Command Line Scripting Languages OS App Control Scripting Languages \$\partial \text{1} - Rust \$\partial \text{1} - Scheme \text{1}\$ | |
| Data Serialization Languages Markup Languages Programming Languages Main Paradigm of Programming Language Families - Actor Model: (A) - Concatenative (K) - Concurrent: (C) - Functional: (T) Pure: (F) - Imperative: (T) or no token - The programming languages supported by PEL are listed here in alphabetical order. - PEL also provides basic support for other programming languages | MasciiDoc Emacs has support for ending the following lists the programming Languages The following lists the programming Languages The cell colours give PEL has support for sea Aside from the list below the following lists the programming Languages The cell colours give PI ← AppleScript PI - Arc The cell colours give | Exeral build tools but they by, PEL supports installar is nix-mode external pacts is tup-mode external pacts tup-mode external pacts tup-mode external pacts is tup-mode external pacts is tup-mode external pacts. DYAML M Graphviz Dot is several programming languages support in the pacts intered in the pacts in the pacts in the pacts in the pacts in the | Markdown Inguages. PEL currently ported explicitly by PEL v Javascript target ML Family Languages in alphabetical order. The programming languages PI - Forth PI - Gambit TA | in a page. the following tools: when pel-use-nix-mode when pel-use-tup user- M Org-Mode adds extra support for swill grow over time. Lisp Family Languages Scheme Language Dialects ge family(ies). MI - Hy MI - Javascript MI - Julia | M PlantUML ome of them, listed belo Stack Based Languages \$\mathbb{B}\tilde{\chi} - Perl \$\mathbb{B}\tilde{\chi} - Python \$\mathbb{B}\tilde{\chi} - Purescript \$\mathbb{F}\tilde{\chi} - Purescript \$\mathbb{F}\c | M reStructuredText W. Command Line Scripting Languages OS App Control Scripting Languages Pt - Rust Pt - Scheme Pt - Typescript | |
| Data Serialization Languages Markup Languages Programming Languages Main Paradigm of Programming Language Families • Actor Model: (A) • Concatenative (K) • Concurrent: (C) • Functional: (T) Pure: (C) • Imperative: (T) or no token • The programming languages supported by PEL are listed here in alphabetical order. • PEL also provides basic support for other programming languages not listed here. • Emacs supports other programming languages directly, not listed here. | MaciiDoc Emacs has support for ending anguages Curly Bracket Languages The following lists the poort of the cell colours give Bi - Arc Maci Tcup MaciiDoc Emacs has support for ending the following lists the poort of the cell colours give MaciiDoc Emacs has support for ending the following lists the poort of the cell colours give MaciiDoc Emacs has support for ending the following lists the poort of the cell colours give MaciiDoc Emacs has support for ending the following lists the poort of the cell colours give Macii Doc Bi - AppleScript Macii Doc Bi - C Macii Doc Mac | Exeral build tools but they by, PEL supports installar in inix-mode external pacts tup-mode external pacts support in initial pacts in | A are not all documented tion and partial setup of kage | in a page. the following tools: when pel-use-nix-mode when pel-use-tup user- when pel-use-tup user- M Org-Mode adds extra support for s will grow over time. Lisp Family Languages Scheme Language Dialects ge family(ies). Pit - Hy Pit - Julia Pit - LFE © (F)(A) | PlantUML ome of them, listed belo Stack Based Languages PI - Perl PI - Python PI - Purescript PI - Racket (*) | M reStructuredText W. Command Line Scripting Languages OS App Control Scripting Languages PI - Rust PI - Scheme PI - Typescript PI - UNIX Shell | |
| Data Serialization Languages Markup Languages Main Paradigm of Programming Language Families • Actor Model: (A) • Concatenative (K) • Concurrent: (©) • Functional: (†) Pure: (F) • Imperative: (1) or no token • The programming languages supported by PEL are listed here in alphabetical order. • PEL also provides basic support for other programming languages not listed here. • Emacs supports other programming languages directly, not listed here. Upcoming support for Elm, Purescript, ReasonML, Typescript | MasciiDoc Emacs has support for ending the following lists the programming Languages The following lists the programming Languages The cell colours give PEL has support for sea Aside from the list below the following lists the programming Languages The cell colours give PI ← AppleScript PI - Arc The cell colours give | Exeral build tools but they by, PEL supports installar is nix-mode external pacts is tup-mode external pacts tup-mode external pacts tup-mode external pacts is tup-mode external pacts is tup-mode external pacts. DYAML M Graphviz Dot is several programming languages support in the pacts intered in the pacts in the pacts in the pacts in the pacts in the | Markdown Inguages. PEL currently ported explicitly by PEL v Javascript target ML Family Languages in alphabetical order. The programming languages PI - Forth PI - Gambit TA | in a page. the following tools: when pel-use-nix-mode when pel-use-tup user- M Org-Mode adds extra support for swill grow over time. Lisp Family Languages Scheme Language Dialects ge family(ies). MI - Hy MI - Javascript MI - Julia | M PlantUML ome of them, listed belo Stack Based Languages \$\mathbb{B}\tilde{\chi} - Perl \$\mathbb{B}\tilde{\chi} - Python \$\mathbb{B}\tilde{\chi} - Purescript \$\mathbb{F}\tilde{\chi} - Purescript \$\mathbb{F}\c | M reStructuredText W. Command Line Scripting Languages OS App Control Scripting Languages Pt - Rust Pt - Scheme Pt - Typescript | |
| Data Serialization Languages Markup Languages Main Paradigm of Programming Language Families - Actor Model: (A) - Concatenative (K) - Concurrent: (C) - Functional: (F) Pure: (F) - Imperative: (I) or no token - The programming languages supported by PEL are listed here in alphabetical order. - PEL also provides basic support for other programming languages not listed here. - Emacs supports other programming languages directly, not listed here. Upcoming support for Elm, | MaciiDoc Emacs has support for ending anguages Curly Bracket Languages The following lists the poort of the cell colours give Bi - Arc Maci Tcup MaciiDoc Emacs has support for ending the following lists the poort of the cell colours give MaciiDoc Emacs has support for ending the following lists the poort of the cell colours give MaciiDoc Emacs has support for ending the following lists the poort of the cell colours give MaciiDoc Emacs has support for ending the following lists the poort of the cell colours give Macii Doc Bi - AppleScript Macii Doc Bi - C Macii Doc Mac | Exeral build tools but they by, PEL supports installar in inix-mode external pacts tup-mode external pacts support in initial pacts in | A are not all documented tion and partial setup of kage | in a page. the following tools: when pel-use-nix-mode when pel-use-tup user- when pel-use-tup user- M Org-Mode adds extra support for s will grow over time. Lisp Family Languages Scheme Language Dialects ge family(ies). Pit - Hy Pit - Julia Pit - LFE © (F)(A) | PlantUML ome of them, listed belo Stack Based Languages PI - Perl PI - Python PI - Purescript PI - Racket (*) | M reStructuredText W. Command Line Scripting Languages OS App Control Scripting Languages PI - Rust PI - Scheme PI - Typescript PI - UNIX Shell | |