PEL Topics Index

| Emacs Reference Cards | | | glish version of the quick lese cards provide usefu | | IU Emacs and popular of | external packages. |
|--|--|--|--|--|---|--|
| With PEL you can access these via the <f11>? e r key sequence. See ∑ Help/Info</f11> | Emacs | Calc | Gnus | Magit Cheatsheet | Org | Viper |
| <u>у пер/шо</u> | Emacs survival card | Dired | Gnus booklet | Magit Ref-card | | VIP |
| PEL repo PEL Readme PEL Manual | For the best user ex • Mozilla Firefox • With that in pla | perience, use a browser (version > 78) does that ce, you can browse thro | ach cell holds a hyperlinl that can render PDF dir t perfectly. You may nee ugh all the PDFs quickly IF by typing the <f11></f11> | ectly instead of downloa ed to activate a plug-in for and reach a vast amour | nding. or other browsers. nt of information quickly. | |
| | He symbols, colou | r coding and various oth | ner conventions are desc | cribed in the <u>>Legend</u> F | PDF. | |
| General Information. | <u>≻Legend</u> | ≻Recommended Ema | acs User Option | <u>≻Themes</u> | | |
| Development Information | <u>>PEL</u> | iMenu/Speedbar si | <u>upport</u> | PEL Naming Conve | entions | |
| Migration Guide | >CRiSP ≈ Emacs | | | | | |
| OS Desktop Key Bindings | <u>≰ macOS Keys</u> | ≰ terminal settings | | | | |
| § Feature Comparisons | Completion Modes | Compatibility | § Speedbar/iMenu M | Mode Compatibility | § Shells/Terminals C | <u>omparisons</u> |
| Key Prefixes & Suffixes | <u> </u> | | Numkeypad | <u>>PEL</u> | <u>■Keys - Fn</u> | <u>■Keys - F11</u> |
| Emacs Features | The links that start with | only ∑ Emacs generic t | features, the blue links a | re external packages. Th | ne green links are mostly | PEL extensions. |
| See a Guided Tour of Emacs. | ∑ Abbreviations | <u>∑ Cursor</u> | ∑ Filling/ Justification | Bίχ- Lispy | <u>∑ Scrolling</u> | ∑ Time Tracking |
| The PEL tables named at right 🕒 | <u>∑ Align</u> | <u>∑ Customize</u> | <u>∑ Frames</u> | <u></u> Marking | ∑ Search/Replace | <u></u> <u>∑ Transpose</u> |
| describe the Emacs commands and key bindings for generic Emacs concepts and features. Emacs commands can be executed | ∑ Auto-Completion | ∑ Cut & Paste | <u>∑ Grep</u> | <u>≫ Menus</u> | ∑ Semantic | ∑ X Treemacs |
| | ∑ Autosave/Backup | ∑ Diff & Merge | ∑ Help/Info | <u>∑ Mode Line</u> | ∑ Sessions | ∑ Undo/Redo/ Repeat/Arg |
| by name or bound to key sequences. The commands may have arguments and keys can express them. | <u></u> <u>Bookmarks</u> | <u></u> <u>Dired</u> | <u>∑ Hide/Show</u> | <u>∑ Mouse</u> | ∑ Shells, REPLs & terminal emulators | ∑ VCS-Git XMagit |
| See: | <u></u> Buffers | ∑ Display - Lines | <u></u> Highlight | ∑ Narrowing | ∑ ℜ Smartparens | ∑ VCS-Mercurial |
| Emacs Keys Numeric Arguments | ➤ Case Conversions | | | ➤ Navigation | Sorting | > VCS-Subversion |
| Running Command by Name | ∑ Closing/ Suspending | <u>∑ Enriched Text</u> | ∑ Indentation | <u>∑ Outline</u> | ∑ Speedbar | <u>∑ Web</u> |
| Emacs uses a concept of modes. See: | ∑ Comments | ∑ Faces/Fonts | ∑ Input Method | ∑ Packages | ∑ Spell Checking | Whitespace |
| Emacs Major and Minor Modes Major Modes | ∑ Completion/Input | | ∑ Inserting Text | ∑X Projectile | ∑ SyntaxCheck | ∑ Windows |
| Minor Modes Choosing Modes PEL provides several key sequences | ∑ Counting | ∑ File-mngt | ∑ Key-Chords | <u>∑ Rectangles</u> | T Templates | <u>∑ Xref</u> - Cross |
| to toggle minor modes, described in | | | | | | References |
| the relevant PDFs. | <u>∑M CUA</u> | ∑ File/Directory Variables | ∑ Keyboard Macros | <u> </u> | <u>∑ Text Modes</u> | References |
| | <u>∑M CUA</u> <u>≨ ERT</u> (Emacs Lisp Re | Variables | ∑ Keyboard Macros | ∑ Registers | | References |
| ұ와ւ - Emacs Lisp concepts & tools | <u>⋠ ERT</u> (Emacs Lisp Re | Variables egression Testing) | ₫ Hooks | <u> </u> | | |
| ⊈∰≀ - Emacs Lisp concepts & tools XRef - Cross Reference Tools | # ERT (Emacs Lisp Re Emacs supports variou | Variables egression Testing) s cross reference mecha | # Hooks anisms described in the | ≴% - Emacs Lisp Type <u>▼ Xref</u> table. These me | 95 | e of various external |
| ⊈∰≀ - Emacs Lisp concepts & tools XRef - Cross Reference Tools | # ERT (Emacs Lisp Re Emacs supports variou | Variables egression Testing) s cross reference mecha | # Hooks anisms described in the | ≴% - Emacs Lisp Type <u>▼ Xref</u> table. These me | es echanisms take advantag | e of various external |
| ¥∰I - Emacs Lisp concepts & tools XRef - Cross Reference Tools See also: <u>▼ Xref</u> | Emacs supports variou tools and integrate with Xref-Support PEL has support for se Aside from the list belo Nix Requires | egression Testing) s cross reference mecha them. Notes about the A Xref-Backend veral build tools but they | # Hooks anisms described in the parameters are available in a vare not all documented the parameters are no | ** - Emacs Lisp Type ** Xref table. These me the tables listed in this s lin a page. the following tools: | echanisms take advantage section. This is work | e of various external in progress. |
| XRef - Cross Reference Tools See also: ∑ Xref Build Tools | Emacs supports variou tools and integrate with Xref-Support PEL has support for se Aside from the list belo Nix Requires Tup Requires | egression Testing) s cross reference mecha them. Notes about the Xref-Backend veral build tools but they w, PEL supports installa inix-mode external pace | # Hooks anisms described in the parameters are available in a vare not all documented the parameters are no | ** - Emacs Lisp Type ** Xref table. These me the tables listed in this s l in a page. the following tools: when pel-use-nix-mode | echanisms take advantage section. This is work | e of various external in progress. |
| XRef - Cross Reference Tools See also: Xref Build Tools Data Serialization | ## ERT (Emacs Lisp Re Emacs supports variou tools and integrate with ## Xref-Support PEL has support for se Aside from the list belo Nix Requires Tup Requires ## Requires | gression Testing) s cross reference mechanthem. Notes about the Market M | ## Hooks anisms described in the parameters are not all documented tion and partial setup of kage activated activat | ** - Emacs Lisp Type ** Xref table. These me the tables listed in this s l in a page. the following tools: when pel-use-nix-mode | echanisms take advantage section. This is work | e of various external in progress. |
| XRef - Cross Reference Tools See also: Xref Build Tools Data Serialization Data Modelling/ Specification | ## ERT (Emacs Lisp Received From the list below Requires Tup Requires PI - Make ASN.1 asn1-mode Asing Requires Asn.1 asn1-mode Asing Requires Asn.1 asn1-mode Asn.2 Asn.2 Asn.2 Asn.3 Asn.3 Asn.3 Asn.3 Asn.4 Asn | veral build tools but they w, PEL supports installar in inx-mode external pacts tup-mode external pacts tup-mode external pacts tup-mode external pacts with the external pacts tup-mode external pact | ### Hooks anisms described in the parameter and all documented tion and partial setup of kage activated a | ** - Emacs Lisp Type ** Xref table. These me the tables listed in this s I in a page. the following tools: when pel-use-nix-mode when pel-use-tup user- | echanisms take advantage section. This is work | e of various external in progress. |
| XRef - Cross Reference Tools See also: Xref Build Tools Data Serialization Data Modelling/ Specification Markup Languages | Emacs supports variou tools and integrate with Exref-Support PEL has support for se Aside from the list belo Nix Requires Tup Requires PL Make COLUMN ASSIN.1 asn1-mode | veral build tools but they w, PEL supports installar intx-mode external paces tup-mode external paces MIB snmp-mode M Markdown | # Hooks anisms described in the parameter and the parameter and partial setup of the parameter activated with the parameter activat | ** - Emacs Lisp Type ** Xref table. These me the tables listed in this s l in a page. the following tools: when pel-use-nix-mode | echanisms take advantage section. This is work | e of various external in progress. |
| XRef - Cross Reference Tools See also: Xref Build Tools Data Serialization Data Modelling/ Specification Markup Languages Graphics Markup Programming Languages | Emacs supports variou tools and integrate with Exert Support PEL has support for se Aside from the list belo Nix Requires Tup Requires Tup Requires CWL SASN.1 asn1-mode M AsciiDoc M Graphviz Dot Emacs has major mode | Egression Testing) s cross reference mecha in them. Notes about the Exercise Streft S | ## Hooks anisms described in the pase tools are available in a vare not all documented tion and partial setup of tage activated activat | ** - Emacs Lisp Type ** Xref table. These me the tables listed in this s Lin a page. the following tools: when pel-use-nix-mode when pel-use-tup user- **M reStructuredText** **EL currently adds extra* | echanisms take advantage section. This is work | e of various external in progress. |
| XRef - Cross Reference Tools See also: Xref Build Tools Data Serialization Data Modelling/ Specification Markup Languages Graphics Markup Programming Languages Main Paradigm of Programming | Emacs supports variou tools and integrate with Exert Support PEL has support for se Aside from the list belo Nix Requires Tup Requires Tup Requires ASN.1 asn1-mode MASCIIDOC MASCI | egression Testing) s cross reference mecha in them. Notes about the Exref-Backend veral build tools but they w, PEL supports installa inix-mode external pace is tup-mode external pace is tup-mode M Markdown M MscGen e support for several pro amming languages supp Functional | # Hooks anisms described in the passe tools are available in a real and a real and a real action and partial setup of the passes of the passe | ** - Emacs Lisp Type ** Xref table. These me the tables listed in this s lin a page. the following tools: when pel-use-nix-mode when pel-use-tup user- **M reStructuredText **EL currently adds extra will grow over time. Lisp Family | echanisms take advantage section. This is work to be user-option is tuned on option is tuned on. | n, listed below. |
| XRef - Cross Reference Tools See also: Xref Build Tools Data Serialization Data Modelling/ Specification Markup Languages Graphics Markup Programming Languages Main Paradigm of Programming Language Families | Emacs supports variou tools and integrate with Exercise Support PEL has support for se Aside from the list belo Nix Requires Tup Requires Tup Requires ASN.1 asn1-mode M AsciiDoc M Graphviz Dot Emacs has major mode The number of progr BEAM Programming Languages Curly Bracket | Pariables Pagression Testing) Socross reference mechanthem. Notes about the More about the Mor | ## Hooks anisms described in the passe tools are available in a partial setup of the passes of the | ** - Emacs Lisp Type ** Xref table. These me the tables listed in this s Lin a page. the following tools: when pel-use-nix-mode when pel-use-tup user- **M reStructuredText PEL currently adds extra will grow over time. Lisp Family Languages **Scheme Language** | essection. This is work e user-option is tuned on option is tuned on. support for some of ther Lisp-like Languages Stack Based | m, listed below. Command Line Scripting Languages OS App Control |
| ENTERPORT MARKUP Programming Languages Main Paradigm of Programming Language Families Actor Model: (A) Concatenative (K) Concurrent: (C) Functional: (T) Imperative: (T) or no token | Emacs supports variou tools and integrate with Exref-Support PEL has support for se Aside from the list belo Nix Requires Tup Requires Tup Requires ASN.1 asn1-mode MASciiDoc MGraphviz Dot Emacs has major mode The number of progr BEAM Programming Languages Curly Bracket Languages The following lists the page of the support of the page of | Variables Pagression Testing) Socross reference mechan them. Notes about the them. Notes about the veral build tools but they were produced by the veral build tools but they were produced by the veral build tools but they were produced by the veral build tools but they were produced by the veral build tools but they were produced by the veral produ | # Hooks anisms described in the parameter of the paramet | ** - Emacs Lisp Type ** - Emacs Lisp Type the tables listed in this s l in a page. the following tools: when pel-use-nix-mode when pel-use-tup user- M reStructuredText **EL currently adds extra will grow over time. Lisp Family Languages Scheme Language Dialects | es echanisms take advantage section. This is work this is work the user-option is tuned on option is tuned on option is tuned on. Support for some of ther Lisp-like Languages | m, listed below. Command Line Scripting Language: |
| XRef - Cross Reference Tools See also: ∑ Xref Build Tools Data Serialization Data Modelling/ Specification Markup Languages Graphics Markup Programming Languages Main Paradigm of Programming Language Families · Actor Model: ④ · Concatenative ⑥ · Concurrent: ⓒ · Functional: ① Pure: ⑥ · Imperative: ① or no token · Has Syntactic Macros: ⑩ | £ERT (Emacs Lisp Re Emacs supports variou tools and integrate with £ Xref-Support PEL has support for se Aside from the list belo Nix | Pariables Degression Testing) Socross reference mechanthem. Notes about the mode and them. Notes about the mode weral build tools but they were mode external pacts tup-mode with mode external pacts tup-mode exte | ## Hooks anisms described in the pase tools are available in a read and partial setup of the pase of | ∑ Xref table. These me the tables listed in this such that tables listed in the tables listed in this such tables listed in the tables listed in this such tables listed in the tables lis | essection. This is work e user-option is tuned on option is tuned on. support for some of ther Lisp-like Languages Stack Based Languages | m, listed below. Command Line Scripting Languages OS App Control Scripting Languages |
| **Pit - Emacs Lisp concepts & tools XRef - Cross Reference Tools See also: ** Xref Build Tools Data Serialization Data Modelling/ Specification Markup Languages • Graphics Markup Programming Languages Main Paradigm of Programming Language Families • Actor Model: *(A) • Concatenative (K) • Concurrent: © • Functional: *(F) Pure: (F) • Imperative: (I) or no token • Has Syntactic Macros: *(III) • The programming languages supported by PEL are listed here in | Emacs supports variou tools and integrate with Stref-Support PEL has support for se Aside from the list belo Nix Requires Tup Requires Tup Requires MASCIIDOC M | Pariables Pagression Testing) Socross reference mechan them. Notes about the page of the | # Hooks anisms described in the parameter of the paramet | ** - Emacs Lisp Type ** - Emacs Lisp Type ** Tref table. These me ** the tables listed in this s ** lin a page. ** the following tools: ** when pel-use-nix-mode ** when pel-use-tup user- ** Mare Tructured Text ** PEL currently adds extra ** will grow over time. ** Lisp Family Languages ** Scheme Language ** Dialects ** ge family(ies). ** PI - Hy (python) ** These ** The properties of the | essection. This is work e user-option is tuned on option is tuned on. support for some of ther Lisp-like Languages Stack Based Languages | n, listed below. Command Line Scripting Language: Scripting Language: |
| XRef - Cross Reference Tools See also: ∑ Xref Build Tools Data Serialization Data Modelling/ Specification Markup Languages Graphics Markup Programming Languages Main Paradigm of Programming Language Families Actor Model: ⑥ Concatenative ⑥ Concurrent: ⑥ Functional: ⑦ Pure: ⑥ Imperative: ① or no token Has Syntactic Macros: ⑪ The programming languages supported by PEL are listed here in alphabetical order. PEL also provides basic support | # ERT (Emacs Lisp Re Emacs supports variou tools and integrate with # Xref-Support PEL has support for se Aside from the list belo • Nix Requires • Tup Requires • The number • The number of programming Languages Curly Bracket Languages The following lists the p • The cell colours give ### AppleScript ################################### | Pariables Pagression Testing) Socross reference mechanthem. Notes about the management of them. Notes about the management of the manage | ### Hooks anisms described in the pase tools are available in are not all documented tion and partial setup of kage activated | ** - Emacs Lisp Type ** Xref table. These me the tables listed in this s Lin a page. the following tools: when pel-use-nix-mode when pel-use-tup user- **M reStructuredText **DEL currently adds extra will grow over time. Lisp Family Languages **Scheme Language Dialects **ge family(ies). **Pi - Hy (python) **The Company of the C | essection. This is work e user-option is tuned on option is tuned on. support for some of ther Lisp-like Languages Stack Based Languages | n, listed below. Command Line Scripting Languages OS App Control Scripting Languages |
| XRef - Cross Reference Tools See also: ∑ Xref Build Tools Data Serialization Data Modelling/ Specification Markup Languages Graphics Markup Programming Languages Main Paradigm of Programming Language Families - Actor Model: ♠ - Concatenative ⓒ - Functional: ♠ Pure: ℮ - Imperative: ① or no token - Has Syntactic Macros: ⑪ 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 variou tools and integrate with Stref-Support PEL has support for se Aside from the list belo Nix Requires Tup Requires Tup Requires MASCIIDOC M | Pariables Pagression Testing) Socross reference mechan them. Notes about the page of the | # Hooks anisms described in the parameter of the paramet | ** - Emacs Lisp Type ** - Emacs Lisp Type ** Tref table. These me ** the tables listed in this s ** lin a page. ** the following tools: ** when pel-use-nix-mode ** when pel-use-tup user- ** Mare Tructured Text ** PEL currently adds extra ** will grow over time. ** Lisp Family Languages ** Scheme Language ** Dialects ** ge family(ies). ** PI - Hy (python) ** These ** The properties of the | essection. This is work e user-option is tuned on option is tuned on. support for some of ther Lisp-like Languages Stack Based Languages | n, listed below. Command Line Scripting Languages OS App Control Scripting Languages |
| Tools See also: ▼ Xref Build Tools Data Serialization Data Modelling/ Specification Markup Languages • Graphics Markup Programming Languages Main Paradigm of Programming Language Families • Actor Model: ④ • Concatenative ⑥ • Concurrent: ⑥ • Functional: ① Pure: ⑥ • Imperative: ① or no token • Has Syntactic Macros: ⑪ • 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, | Emacs supports variou tools and integrate with Emacs support variou tools and integrate with Emacs support variou tools and integrate with Emacs support or se Aside from the list belo Nix Requires Tup Requires Tup Requires ASN.1 asn1-mode MASCIIDOC | Pariables Pagression Testing) Socross reference mechan them. Notes about the them. Notes are the them. Notes a tup-mode external pactions as tup-mode external pactions. Markdown M. Markdown M. Markdown M. Markdown M. MacGen Pannetional Languages Java Virtual Machine Languages Torogramming languages a coarse indication of the them. Pagrenetics of the them. The them them. Common Lisp Them. Pagrenetics of the them. | ## Hooks anisms described in the passe tools are available in a real and a real action and partial setup of a charge activated activate | ** - Emacs Lisp Type ** Xref table. These me the tables listed in this s Lin a page. the following tools: when pel-use-nix-mode when pel-use-tup user- ** YestructuredText ** PEL currently adds extra will grow over time. Lisp Family Languages ** Scheme Language Dialects ** pe family(ies). ** PI - Hy (python) ** pi - Janet () ** pi ** PI - Janet () ** pi ** PI - Javascript ** PI - Julia () ** PI | essection. This is work e user-option is tuned on option is tuned on. support for some of ther Lisp-like Languages Stack Based Languages \$\text{Pi} - OCaml | n, listed below. Command Line Scripting Languages OS App Control Scripting Languages Pi - Ruby Pi - Rust Pi - Scheme Pii - Typescript |
| TPI - Emacs Lisp concepts & tools XRef - Cross Reference Tools See also: ∑ Xref Build Tools Data Serialization Data Modelling/ Specification Markup Languages • Graphics Markup Programming Languages Main Paradigm of Programming Language Families • Actor Model: ④ • Concatenative ⑥ • Concurrent: ⑥ • Functional: ♠ Pure: ♠ • Imperative: ① or no token • Has Syntactic Macros: ⑪ • 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, | Emacs supports variou tools and integrate with Exref-Support PEL has support for se Aside from the list belo Nix Requires Tup Requires Tup Requires ASN.1 asn1-mode MASCIIDOC MASCI | Pariables Degression Testing) Secross reference mechan them. Notes about the memory of them. Notes are the memory of them. Degrees a coarse indication of the memory of them. Degrees a coarse indication of the memory of them. Degrees a coarse indication of the memory of them. Degrees a coarse indication of the memory of them. | # Hooks anisms described in the pase tools are available in an available in a are not all documented tion and partial setup of kage activated act | ** - Emacs Lisp Type ** Xref table. These me the tables listed in this s lin a page. the following tools: when pel-use-nix-mode when pel-use-tup user- ** MreStructuredText ** EL currently adds extra will grow over time. Lisp Family Languages ** Scheme Language Dialects ** Dialects * | essection. This is work e user-option is tuned on option is tuned on option is tuned on. support for some of ther Lisp-like Languages Stack Based Languages \$\text{21} \text{1} - OCaml \text{1} \text{1} \text{1} \$\text{21} - Perl \text{21} \text{21} - Python | m, listed below. Command Line Scripting Languages Pi - Ruby Pi - Rust Pi - Scheme The original contents of the contents o |
| XRef - Cross Reference Tools See also: ∑ Xref Build Tools Data Serialization Data Modelling/ Specification Markup Languages • Graphics Markup Programming Languages Main Paradigm of Programming Language Families • Actor Model: ⑥ • Concatenative ⑥ • Concurrent: ⑥ • Functional: ∱ Pure: ⑥ • Imperative: ① or no token • Has Syntactic Macros: ⑪ • 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, | Emacs supports variou tools and integrate with Emacs support variou tools and integrate with Emacs support variou tools and integrate with Emacs support or se Aside from the list belo Nix Requires Tup Requires Tup Requires ASN.1 asn1-mode MASCIIDOC | Pariables Pagression Testing) Socross reference mechan them. Notes about the them. Notes are the them. Notes a tup-mode external pactions as tup-mode external pactions. Markdown M. Markdown M. Markdown M. Markdown M. MacGen Pannetional Languages Java Virtual Machine Languages Torogramming languages a coarse indication of the them. Pagrenetics of the them. The them them. Common Lisp Them. Pagrenetics of the them. | ## Hooks anisms described in the passe tools are available in a real and a real action and partial setup of a charge activated activate | ** - Emacs Lisp Type ** Xref table. These me the tables listed in this s Lin a page. the following tools: when pel-use-nix-mode when pel-use-tup user- ** Year Currently adds extra will grow over time. Lisp Family Languages ** Scheme Language Dialects ** Jeff family(ies). ** Pi - Hy (python) m ** Pi - Janet () fm ** Pi - Javascript ** Julia m | essection. This is work e user-option is tuned on option is tuned on. support for some of ther Lisp-like Languages Stack Based Languages \$\text{Pi} - OCaml | n, listed below. Command Line Scripting Languages OS App Control Scripting Languages Pi - Ruby Pi - Rust Pi - Scheme Pi - Typescript |