













Emacs support for Rust 🚧

| Description | Keystroke | Function | Note |
|--|--|---|--|
| Rust Programming Language Support | <div>  This is is an early version of this page. More information will be provided soon. </div> <div>  PEL activates Rust support when the pel-use-rust user-option is turned on (t). </div> <div> PEL provide support for the Rust programming language and its various implementations by providing access to the following external packages: </div> <div> <div>  The rust-mode external package. </div> <div>  PEL activates it when the pel-use-rust-mode user-option is turned on (t). </div> </div> <div> <div>  The rustic external package. </div> <div>  PEL activates it when the pel-use-rustic user-option is turned on (t). </div> </div> <div> <div>  The flycheck-rust external package. </div> <div>  PEL activates it when the pel-use-flycheck-rust user-option is turned on (t). </div> </div> <div> <div>  The emacs-racer external package. </div> <div>  PEL activates it when the pel-use-emacs-racer user-option is turned on (t). </div> </div> <div> <div>  The cargo external package. </div> <div>  PEL activates it when the pel-use-cargo user-option is turned on (t). </div> </div> | | |
| Open this PDF file. See also: 🔗 Help/Info | <div><f11> SPC r <f1></div> <div><f12> <f1></div> | <div>(pel-help-pdf &optional OPEN-WEB-PAGE)</div> | Open the 🔗 - Rust PDF using method specified by the pel-open-pdf-method user-option or the alternate one if a command prefix (like C-u) was used. |
| 🔗 Customize PEL Rust support | <div><f11> SPC r <f2></div> <div><f12> <f2></div> | <div>(pel-customize-pel &optional OTHER-WINDOW)</div> | Customize PEL Rust support. <ul style="list-style-type: none"> If OTHER-WINDOW is non-nil (use C-u), display in another window. |
| 🔗 Customize Emacs Rust support | <div><f11> SPC r <f3></div> <div><f12> <f3></div> | <div>(pel-customize-library &optional OTHER-WINDOW)</div> | Customize Emacs Rust support: rust-mode, rustic, racer, cargo. <ul style="list-style-type: none"> If OTHER-WINDOW is non-nil (use C-u), display in another window. |
| | | | |
| Cargo run | <f12> c | (rust-run) | Build the Rust file using Cargo and run it. |
| Add/Remove the dbg! macro | <f12> d | (rust-dbg-wrap-or-unwrap) | Either remove or add the dbg! macro. |
| Run Clippy, Rust Lint Checker | <f12> l | (rust-run-clippy) | Run ‘cargo clippy’. |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Emacs & Rust — References

| Document | Notes |
|--|---|
| Fancy Rust development with Emacs | May 2016. Describes how to use rust-mode |
| rust-mode: A major Emacs mode for editing Rust source code | A GitHub site |
| Racer for emacs | |
| company-mode ; Modular in-buffer completion framework for Emacs | |
| | |
| Why Rust? | Safari book online |
| rust-cross | This GitHub site states: Everything you need to know about compiling rust programs! |
| Taking Rust everywhere with rustup | A Rust site blog on rustup |
| Cross compiling Rust on OS X for Raspberry Pi 3 | March 2016 article on cross compiling Rust on Raspberry Pi3 |
| Raspberry Pi Bare Metal Programming with Rust | |
| Rust source code | |
| | |
| | |
| | |