

# Emacs support for Rust 🚧

| Description  | Keystroke   | Function  | Note  |
|--|---|---|---|
| <a href="#">Rust Programming Language Support</a>            | This is an early version of this page. More information will be provided soon.  |   |   |
|  | PEL activates <a href="#">Rust</a> support when the <b>pel-use-rust</b> user-option is turned on ( <b>t</b> ).<br>PEL provide support for the Rust programming language and its various implementations by providing access to the following external packages: |   |   |
|  | The <a href="#">rust-mode</a> external package.   | PEL activates it when the <b>pel-use-rust-mode</b> user-option is turned on ( <b>t</b> ).     |   |
|  | The <a href="#">rustic</a> external package.  | PEL activates it when the <b>pel-use-rustic</b> user-option is turned on ( <b>t</b> ).        |   |
|  | The <a href="#">flycheck-rust</a> external package.   | PEL activates it when the <b>pel-use-flycheck-rust</b> user-option is turned on ( <b>t</b> ). |   |
|  | The <a href="#">emacs-racer</a> external package.   | PEL activates it when the <b>pel-use-emacs-racer</b> user-option is turned on ( <b>t</b> ).   |   |
|  | The <a href="#">cargo</a> external package.   | PEL activates it when the <b>pel-use-cargo</b> user-option is turned on ( <b>t</b> ).         |   |
| Open this PDF file.<br>See also: <a href="#">🔗 Help/Info</a> | <a href="#">&lt;f11&gt; SPC r &lt;f1&gt;</a><br><a href="#">&lt;f12&gt; &lt;f1&gt;</a>  | <a href="#">(pel-help-pdf &amp;optional OPEN-WEB-PAGE)</a>                                    | Open the <a href="#">📄 - Rust</a> local PDF. If the prefix argument (like <b>C-u</b> or <b>M--</b> ) is used, then it opens the remote GitHub hosted raw PDF instead. If the <b>pel-flip-help-pdf-arg</b> user-option is set it's the other way around. |
| <a href="#">🔗 Customize</a> PEL Rust support                 | <a href="#">&lt;f11&gt; SPC r &lt;f2&gt;</a><br><a href="#">&lt;f12&gt; &lt;f2&gt;</a>  | <a href="#">(pel-customize-pel &amp;optional OTHER-WINDOW)</a>                                | Customize PEL Rust support. <ul style="list-style-type: none"><li>If OTHER-WINDOW is non-nil (use <b>C-u</b>), display in another window.</li></ul>   |
| <a href="#">🔗 Customize</a> Emacs Rust support               | <a href="#">&lt;f11&gt; SPC r &lt;f3&gt;</a><br><a href="#">&lt;f12&gt; &lt;f3&gt;</a>  | <a href="#">(pel-customize-library &amp;optional OTHER-WINDOW)</a>                            | Customize Emacs Rust support: rust-mode, rustic, racer, cargo. <ul style="list-style-type: none"><li>If OTHER-WINDOW is non-nil (use <b>C-u</b>), display in another window.</li></ul>  |
|  |   |   |   |
| Cargo run  | <a href="#">&lt;f12&gt; c</a>   | <a href="#">(rust-run)</a>  | Build the Rust file using Cargo and run it.   |
| Add/Remove the dbg! macro                                    | <a href="#">&lt;f12&gt; d</a>   | <a href="#">(rust-dbg-wrap-or-unwrap)</a>   | Either remove or add the dbg! macro.  |
| Run Clippy, Rust Lint Checker                                | <a href="#">&lt;f12&gt; l</a>   | <a href="#">(rust-run-clippy)</a>   | Run 'cargo clippy'.   |
|  |   |   |   |
|  |   |   |   |
|  |   |   |   |
|  |   |   |   |
|  |   |   |   |
|  |   |   |   |

## Emacs & Rust — References

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