

Rust のインストール

[インストール - The Rust Programming Language 日本語版](#)

Rust は rustup というツールコマンドのインストールから始まる。
rustup コマンド経由で、Rust 本体やら便利コマンドやらをインストールしていく。

rustup のインストール、そして Rust のインストール

(専用の解説ページ)

[Introduction - The rustup book](#)

mac / Linux

RUSTUP_HOME と CARGO_HOME の環境変数を事前に設定しておく、\$HOME/.rustup や \$HOME/.cargo が勝手に作られないようにできる。

わたしは

```
$ export RUSTUP_HOME=$HOME/local/rust/rustup
$ export CARGO_HOME=$HOME/local/rust/cargo
```

としてインストールした。

- 直接インストールする(かんたんだしおすすめ)
 - `$ curl --proto '=https' --tlsv1.2 https://sh.rustup.rs -sSf | sh`
 - (わたしはこの方法あんまり信用していないので、ファイルをダウンロードして中身を眺めて変なことしていないか確認してから実行するようにしてる)
 - 選択を求められたら
 - 「1) Proceed with installation (default)」が一番かんたん
 - (.profile とかに勝手に追記されているので、適宜好きなように整理)
- (mac) homebrew 経由でインストールする
 - `$ brew install rustup-init`
 - 上記のシェルファイルを取ってきてるだけっぽい？
 - rustup-init という名前でエイリアスになっているらしい
 - `$ brew install rustup` でもたぶん同じ
 - `$ rustup-init`
 - `$ rustup update stable`
- Docker 経由で使うので mac / Linux には直接入れない
 - `$ docker pull rust:1.62.0-slim-buster`

Windows

- rustup を直接インストールする(かんたんだしおすすめ)

- [Install Rust - Rust Programming Language](#)
 - rustup-init.exe をダウンロードして実行する
 - でもたぶんライブラリが足りないと言われるので、そちらを先にインストールする
 - <https://visualstudio.microsoft.com/ja/visual-cpp-build-tools/>
 - もし Visual Studio をインストールしているなら、Visual Studio 経由で以下を追加インストールでも OK
 - .NET desktop development
 - Desktop development with C++
 - Universal Windows Platform development
 - (Chocolatey みたいな管理ツール経由でもいけるけど、難易度ちょっと高い?)

Rust インストール確認

```
$ rustc --version
rustc 1.62.0 (a8314ef7d 2022-06-27)
```

rustup ごと全部消すとき

```
$ rustup self uninstall
```

その他にインストールされるコマンド周辺

- cargo
 - Ruby という bundler みたいなやつ。これからよく使う
- cargo-clippy
 - cargo 経由で clippy を使うやつ。clippy は「Rust はこう書いたほうがいい」の例集で、いい書き方を教えてくれるけど無視もできる
- rustfmt
 - 公式のコード整形ツール。エディタと連携させて使うと便利
- rustdoc
 - コード内のコメントをドキュメントにしてくれる。Rust のドキュメントはブラウザで見ることが前提となっている

たぶん直接使うのは cargo だけ。

オンラインでちょっと試すだけの環境

[Rust Playground](#)

Hello, world

どこでもいいので以下のファイルを作成する。

rustc コマンドでコンパイルして、実行する。

(以後、rustc コマンドを直接使う機会は実はほとんどないので、rustc コマンドを使うのはこれで最後かもしれない)

main.rs

```
fn main() {  
    println!("Hello, world!");  
}
```

mac / Linux

```
$ rustc main.rs  
$ ./main
```

Windows

```
> rustc main.rs  
> .\main.exe
```

Visual Studio Code が便利

Visual Studio Code の Rust 専用ページ

[Rust with Visual Studio Code](#)

余談

Visual Studio Code で Rust を書くなら、補完や型推論してくれる言語サーバをインストールするととても便利。

RLS という LSP (Language Server Protocol) の実装があったが、つい最近これの非推奨が決まった。

代わりに rust-analyzer を使うべし、となった。

[RLS Deprecation | Rust Blog](#)

というわけで、Visual Studio Code の拡張機能から、rust-analyzer を入れておくのがよい。

rust-analyzer

<https://marketplace.visualstudio.com/items?itemName=rust-lang.rust-analyzer>

(非推奨) RLS

<https://marketplace.visualstudio.com/items?itemName=rust-lang.rust>

ちなみに RLS 自体は Rust 自体に付属していて、rustc コマンドと同じところに rls コマンドがある。

Rust 入門いろいろ

[Rust の最初のステップ - Learn | Microsoft Docs](#)

これも割とおすすめ。演習問題もある。

[Rust ツアー - Let's go on an adventure!](#)

長め。公式ではない？ 途中で急に難しくなる。
オンラインで試しながらできるので理解度は高い。