

はじめてのプログラミング言語 Rust

中林 智之

本稿の使い方

● 後回しにしても大丈夫

実験のコードをスムーズに理解できるように、一般的なプログラミングの概念に対して、Rustの文法を解説します。先に実験を読んで、不明点があれば戻っても構いません。ここでは、次の内容について説明します。

- 変数
- データ型 (プリミティブ型)
- コメント
- 関数
- フロー制御

● 丁寧なコンパイル・エラーについて

Rustで特徴的な機能については、実験内で適宜、解説します。

コンパイルが失敗するサンプル・コードに対しては、Rustコンパイラのコンパイル・エラーも掲載します。Rustのコンパイル・エラーは、エラーになった箇所を的確に指摘し、その原因や解決方法まで示してくれます。筆者はここまで丁寧なコンパイル・エラー・メッセージを他のコンパイラで見たことがありません。Rustは難しいプログラミング言語であるた

め、できるだけ丁寧にエラーを出力しよう、というコンパイラ製作者たちの意図が垣間見えます。もちろん、難しいコードを書き始めると、「コンパイル・エラーになる理由は分かるが、直し方が分からない」という事態になることもあります。ぜひRustコンパイラが出力する恐ろしく丁寧なコンパイル・エラーも確認してみてください。

● 基本的な動かし方

掲載するソースコードでは、`fn main() {}`を省略している場合があります。もし実際にコンパイルして実行結果を試したい場合は、必要に応じて`fn main() {}`を補ってください。下記のようなコードがある場合、

```
let x: i32 = 42;
```

次のように書くと、コンパイルおよび実行ができます。

```
fn main() {
    let x: i32 = 42;
}
```

文法入門のサンプル・コードを動かしたい場合、「Rust Playground」を使うのがお勧めです。ウェブ・ブラウザで<https://play.rust-lang.org/>にアクセスするか、「Rust Playground」で検索してください。

図1のようなページが表示されます。左のエディタ



図1 Rustをすぐに動かして試せるウェブ・ページ Rust Playground
サンプル・コードを動かしたい場合に便利