



新しく言語を学ぶ私のやり方

Tomohisa Oda — GMO Pepabo, Inc.

Fukuoka.rs#1 Jan 22, 2019



@linyows

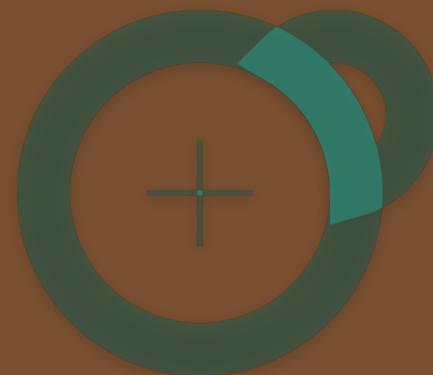
blog: tomohisaoda.com

GMOペパボ

プリンシパルエンジニア



WEB+DB PRESS vol.104
HashiCorp Vaultの記事寄稿



OCTOPASS

linuxユーザの名前解決を
GitHubからマッピングする
ソフトウェアの開発



福岡のGolangコミュニティ
Fukuoka.goの主催の1人



```
println!("Hello Rustaceans");
```



Rustと私

- No Runtime、No GC??? すごーーい?????
- なかなかコンパイル通らないって聞く
- Go使ってて困ることないよ？何が嬉しいの？
- ずっとこの調子なので、発表ドリブンで勉強を開始しよう！
- んで、気づいたら正月も終わって勉強会の1週間前



さあ勉強を始めるぞ

- とりあえず環境の構築 (Rustup入れたり、vim plugin探したり…)
- とりあえずRustのThe Bookを読み始める
- すぐに頭に入ってこなくなったので手を動かし始める
- リポジトリ作って写経 github.com/linyows/rust-learning
- そして飽きて、github.comで★の沢山ついたRustリポジトリを眺める
- 終了…



\モチベーションが続かないゾ!/
🥲





モチベーションを探す旅

- ようし、モチベーションをひねり出すぞ
- Rustって何がいいの！（再）公式サイトにあるはず
- つかいどころはどこ？？（切実）これも、公式サイトにあるはず
- 自分だったらどう使う🤔



Rustの特徴

- とても高速でメモリ効率がよいのでハイパフォーマンス、そして組み込みとか簡単に他言語と統合できる
- 豊富な型とOwnershipモデルによりメモリやスレッドが安全であることを保証する、だから多くの種類のバグをコンパイラで排除できる
- 豊富なドキュメント、賢いコンパイラ、統合されたパッケージツールで生産性高い



Rustのキャッチコピー

信頼性や効率性の高いソフトウェアを
誰もが書ける力を与えるよ！

まじか… 🤪



Rust vs Go

| | Rust | Go |
|--------------------------|-------|-------|
| Performance | Great | Good |
| Simple and Readable Code | OK | Great |
| Concurrency | Yes | Yes |
| Forcing error checking | Yes | No |



マシンリソースを残らず使い切るまで
パフォーマンスを重視したいのならRust!



Rustのつかいどころ

- パフォーマンス重視で安全なソフトウェアの開発であれば一択でしょう！
(パフォーマンスってあとから要求される?)
- Rustでそれなりに生産性高めるには練度が必要そうなので、教育体制もある開発、または熟練集団での開発
- 未熟者は、OSSとか、俺得CLIツールとかはぴったりののでは？



定めたゴール

- まず動くものを作ろう
- 実装の正しさ、スマートさは気にしない
- 実装していく中で、つぎにつながる課題を見つける
- ゴールを明確にするとやることが定まってやる気が出る 💪
- よし!!!!!!!!!!!!



 **linyows**
@linyows

 むずい... 学習コスト高えよ

1:43 - 2019年1月17日

 1   

 別のツイートを追加

 **yoshih** @_yoshih_ · 1月16日

返信先: @linyowsさん

コンパイル通すだけで一苦勞ですよねw でも、コンパイル通るとだいたい意図通りに動いてくれて安心して使える気がしますー

 1   1 

 他1件の返信





作ったCLI

- GeolIPの地理情報をIPで検索して表示するくん

<https://github.com/linyows/glip>

- 1 weekでも動くものはできた！
- そして、crates.ioでパッケージを公開できた！



つまずいたところ

- ファイル分割した際のmoduleの概念
- ライブラリとCLIの両方を提供するパッケージのディレクトリ構成
- エラーチェック強制のためのエラー処理
- unwrapばかりになって、何なのこれ???
- ifの中でmutableな変数更新で怒られる



linyows

@linyows



メモリ安全性のためにある🦀独自の
Ownership、References、Borrowingにちょっ
とだけ慣れてきたような... あと、Option<T>
とResult<T>便利な気がする

10:13 - 2019年1月22日

3件のいいね



3



別のツイートを追加



見えてきた課題

- テスト書いてないの (coverageもとるぞ) 🐱
- うまく書けてないの (genericsやtrait使えば綺麗になるはず)
- そもそも、やってることあってる??
- 配布にcargo installじゃなくてCross Compilingしておきたいやん
- 並行して、足りてない知識を「The Bookを網羅」 🦀本を読了」で得る



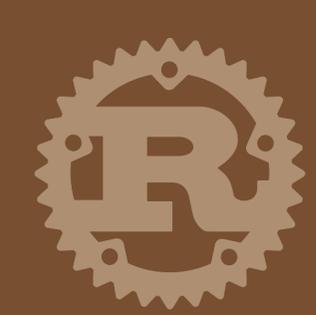
開発ドリブンの何が良いか

- 自分ごとになるので頓挫しにくい👉夜なべしてやってまう🙈
- しょうもないことでよくつまづく👉フラストレーション溜まる🤯
- 体系的に学ぶ意欲が湧く！🙈
- やった結果が残る👉やばいものが出来て恥ずかしい😓
- やばいコードをちゃんとする意欲が湧く！🙈



\なんだかんだ結果的に良い/





まとめ

新しく言語を学ぶために以下の手順ですすめました。

- まずは学ぶ理由を探す、なければ作る
- 開発ドリブンで手を動かしながら学ぶ（明確なゴール）
- 興味が湧いてきたら体系的に学ぶ
- そしてまた作る！



Fukuoka.rsのみなさん
Rust 教えてくださいいいいい!!!!!!!
(コードレビュー希望)

ご静聴ありがとうございました



References

- **The Book** <https://doc.rust-lang.org/book/>
- **Rust** <https://www.rust-lang.org/>
- **Should I Rust, or Should I Go** <https://codeburst.io/should-i-rust-or-should-i-go-59a298e00ea9>
- **RipGrep** <https://github.com/BurntSushi/ripgrep>
- **Bat** <https://github.com/sharkdp/bat>
- **プログラミング Rust** <https://www.oreilly.co.jp/books/9784873118550/>