

Ruby力を上げるための コードリーディング

2019.12.14 Sat. 平成Ruby会議@株式会社ドリコム

kinopyd

SmartHR product engineer





おめでとう

kinoppyd @SmartHR

Ruby好きエンジニア

SmartHRで働いている

ポーカーとかロードバイクとか
する

Rubyだけの話ではないが
力をつけるには
他人のコード読むのが良い

- ・イディオムや習慣
- ・似たようなものを作る時の参考
- ・時々イカしたメタプロが見れる

- DSL作りしたい -> Rakeとか読めばいい
- WAF作りしたい -> Sinatraとか読めば(ry
- 既存機能拡張したい -> ActiveSupport(ry

Rubyのプロダクトコードを書く力をあげるなら、
「これ〇研ゼミで見たやつだ！」って思える
引き出しを増やすことが一番いい

- ・不思議なコードが怖くなる
- ・なぜ動くのかを知ると慌てなくなる
- ・気持ちに余裕がでる

このプロダクトに今日
から参戦するけど俺
ActiveRecordがどう
やって動いてるのか
知ってるし何も怖くない
ぜ

って気持ちになれると多少は不安も解消するかも

でも 🤔



とりあえず読んでみたいけど
何からよんでいいのか
よくわからない

2019/12/11現在 156,610個

**Gemは星の数ほどあるので
選ぶのは大変だと思います**

- ・有名なの
- ・メンテされてるの
- ・歴史があるの



- ・自分のプロダクトで使ってるもの
- ・比較的小規模なもの
- ・他のGemへの依存性が低いもの

▪ **個人的なおすすめ**

▪ **Sinatra**

▪ **ActiveSupport**

▪ **ActiveRecord**



余談: GitHubのrubyタグがついてるリポジトリの上位1000件のなかから、gemspec持っていない(つまりGemじゃない)リポジトリ 141件のGemfileを全部パースしてみた結果、一番使われてた Gemはrake、uglifier、puma、rubocopだった。

読むために
ひつようなもの



- **メタプログラミング Ruby第2版**
- **RubyMine**
- **pry-byebugu**



メタプログラミング Ruby第2版

Gemを読む前に、まず必読

この本の知識がないと、まず自分が

何を理解しなくては良いのかすらわからない

RubyMine

vim派だったんですが、コードジャンプがあまりにも優秀すぎるので乗り換えました
メタプロの前には無力ですが、
通常のコードであれば一瞬で飛んでくれから便利だし、Gemfileを読むので関連する Gemのコードも飛んでくれる

pry-byebug

デバッグ

ステップ実行があれば

RubyMineで立ち向かえないメタプロも怖くない

**実際にやってみる
(実演付きで)**



- **いくつかの注意事項**
- **bundle install --path は避ける**
- **Gemに仕込んだデバッグコードは必ず消す**

- Gemfileにsinatraとpry-byebugを入れる
- RubyMineで読み込む
- 書いたコードでジャンプしてみる

- ・ステップ実行してみる
- ・methodからsource_locationしてみる
- ・grepしてみる

あ、最初に言うのを忘れてましたが
なんだかんだ grepも重要です

bundle show [gem名] でgemの場所はわかるので
grep hogehoge \$(bundle show sinatra) とかで
検索できます

まとめ



- ・メタプロ Rubyは絶対読もう
- ・好きなエディタで書けばいいけど、読むのは
RubyMine使おう
- ・ステップ実行と定義元探しはメタプロ倒せる

いろんな Gem のソース読んで
知っているパターンを増やして
Ruby力をあげよう

Thank you for your attention



Twitter: @GhostBrain

GitHub: kinoppyd

**SmartHR needs more
engineers**