

# Ruby リファレンスマニュアル改善計画 2022 進捗報告

Kazuhiro NISHIYAMA

株式会社 *Ruby* 開発

第90回 *Ruby* 関西 勉強会  
2023-03-25

# self.introduction

---

- 西山 和広
- Ruby のコミッター
- twitter, github など: @znz
- 株式会社Ruby開発 [www.ruby-dev.jp](http://www.ruby-dev.jp)

# agenda

---

- るりまについて
- Ruby リファレンスマニュアル改善計画 2022 について
- 歴史
- 目標

# What is rurema (るりま)?

---

- Japanese Ruby reference manual  
Rubyリファレンスマニュアル刷新計画
- <https://github.com/rurema>
  - [rurema/doctree](#)
    - ドキュメント
  - [rurema/bitclust](#)
    - 独自のドキュメントシステム

# るりま != るびま

---

- Rubyist Magazine
  - <https://magazine.rubyist.net/>
  - <https://github.com/rubima>

- 似ているけど無関係

- FAQより

Q. るびま、って「ネギま！」のぱくりですか？

A. 違います。多分。「るびま」を考えた人たちは「ネギま！」を知りませんでした(または、言われるまで気付かなかった)。

# Ruby リファレンスマニュアル改善計画 2022 とは？

---

Ruby リファレンスマニュアル を Markdown 記法に対応させる計  
画

# 現状の予定

---

- bitclust に Markdown 対応を追加
- doctree でいくつかのドキュメントを Markdown で書いてみる
- (現状は上2点の途中)
- 問題点をみつけて直しつつ、既存のドキュメントも移行していく
- RD のドキュメントを削除して bitclust から RD 対応を削除して移行完了

# Markdown 変換の進捗

---

- 簡単なサンプルを作成して実現可能性を検証
- `kramdown-parser-gfm` でほぼ RD と同じように変換可能
- 分割処理を RD から流用  
→ `code block` 中のコメントを見出しと誤認識する問題発生
- 実現可能性の検証用実装を捨てて Kramdown のパース結果を利用するように作り直し中

# 歴史

---

なぜこういう状況なのかという歴史の話

# るりまより前

---

- Ruby 1.4.6 以前
  - <https://ftp.ruby-lang.org/pub/ruby/doc/>
  - ドキュメントは **RD** で書かれていた
  - English: [ruby-man-1.4.6.tar.gz](http://ruby-man-1.4.6.tar.gz)
  - Japanese: [ruby-man-1.4.6-jp.tar.gz](http://ruby-man-1.4.6-jp.tar.gz)  
(中の HTML ファイルは iso-2022-jp なので文字化けに注意)
- Ruby 1.6 から 1.8 の時代
  - **RWiki** で編集・公開
  - RWiki はいくつかの拡張機能つきの RD を使った Wiki

# るりまが始まった頃

---

- Ruby 1.8 時代
  - Rubyリファレンスマニュアル刷新計画が開始
  - **bitclust** という **RD** ベースの独自記法のシステム
    - bitclust は **RDtool** (<https://rubygems.org/gems/rdtool>) を使っていない
  - **RWiki** で編集していたドキュメントを取り込んだ

# ドキュメントのライセンス

---

- RWiki の編集フォームにライセンスの変更手順を書いておいた
  - (freeml の rubyist ML での合意で変更可能とした)
- **Creative Commons – Attribution 3.0 Unported** に変更  
<https://github.com/rurema/doctree/blob/master/refm/doc/license.rd>

# bitclust 時代になってからの改善点

---

- EUC-JP から UTF-8 に
- コードの色付け (#@samplecode)
  - #@ で始まる行はプリプロセッサ
- chm, ePub に出力可能
  - あまり使われていないので動かない可能性あり
  - デフォルトは静的HTML ([docs.ruby-lang.org](https://docs.ruby-lang.org)もこれ)

# るりまプロジェクトの現状

---

- ドキュメント改善のサブプロジェクト
- Rubyの新しいバージョン対応
- <https://rurema-review.connpass.com/>

# ドキュメント改善ノサブプロジェクト

---

- <https://github.com/rurema/doctree/issues/433>

コピペ可能なサンプルコードを整備する

- サンプルコードの色付け

→ ほとんど完了したからか現在はほぼ停止中

# Rubyの新しいバージョン対応

---

- ところどころ対応
  - メソッドの変更 (追加、引数や機能の変更、削除)
- ほとんど対応できていない
  - 文法の変更 (パターンマッチ, &., ...)
  - どこに書けばいいのかわかりしていない

# rurema-review

---

- <https://rurema-review.connpass.com/>
    - 毎週火曜の夜
    - るりまレビュー会 → るりまもくもく会
    - 鹿児島Ruby会議01をきっかけに開始
  - たくさんの pull request をマージ
- 現在は活動できている人がいない

# 協力者募集

---

- こういう状況なので協力者を増やしたい
- GitHub issues や ruby-jp slack の #rurema
  - <https://github.com/rurema/doctree/issues>
  - <https://ruby-jp.github.io/>

# 目標

---

- 短期目標
- 中期目標
- 長期目標

# RD ベース → Markdown ベース

---

- **最重要短期目標**

- RD より Markdown の方が馴染みがある人が多い
- 貢献してくれる人が増えるはず
- Ruby リファレンスマニュアル改善計画 2022 で作業開始

# 現在の記法

- #@ はプリプロセッサ要の行
- --- は RD の MethodList

```
#@since 3.1  
--- intersect?(other)  -> bool
```

other と共通の要素が少なくとも1個あれば true を、なければ false を返します。

```
#@samplecode 例  
a = [ 1, 2, 3 ]  
b = [ 3, 4, 5 ]  
c = [ 5, 6, 7 ]  
a.intersect?(b)  # => true  
a.intersect?(c)  # => false  
#@end  
#@end
```

# プリプロセッサの移行

---

- バージョン分岐や include など
- エディタや GitHub.com でのサポートも気にしたい
- → Jekyll でも使われている Liquid を採用

# MethodList

---

- MethodList の代用案
- 現在の記法は `def m(args)` ベース
- ブロックの書き方は揺れがある
  - `{|x| ... }`, `{|x| block }`, `&block`
- 返り値は書いてあるが有効活用はあまりされていない

# MethodList の移行

---

- `### def m(args) -> nil` 形式
- 最初は単純に既存の記法を Markdown にするだけ
- 将来は RBS 形式もサポート予定

# 他の記法

---

- `#@samplecode` → code block
- リンク
  - `[[c:String]]` → `[c:String]`
  - Markdown の記法に合わせて `[]` が1段減る
- 細かい問題は臨機応変に対応予定

# 他の短期目標と問題

---

- 使われていないファイルや古いファイルの削除
  - ChangeLog, setup.rb, ...
- tools のファイルがまだ使えるかどうかははっきりしない
- **使い方のドキュメント不足**
  - これが最重要の課題
- 再現可能なビルド
  - container? devcontainer?

# 中期目標(他のツールとの連携)

---

- RBS連携
  - 例: signatures の連携
- IRB連携
  - rdoc の代わりに rurema を表示したい

# 中期目標(ドキュメント)

---

- WASMでサンプルコードを実行可能にしたい
  - hanachin さんが試したものがあ <https://github.com/hanachin/bitclust/commit/1ae60bfabd09c0d241e6966a6800e27a797ce175> <https://github.com/rurema/doctree/issues/2730>
- 標準添付から外れたライブラリや古いドキュメントの整理
  - いくつかは <https://github.com/rurema/historical-documents> に移動済

# 長期目標

---

- l18n 対応
  - rdoc と ruerema は記法だけに限らず、ドキュメントの書き方が違いすぎて統一しにくい
  - gettext か何かを使う？
  - 英語ベースの方が良さそうなので遠い将来の夢になりそう

# end

---

- RD から Markdown への移行開始中
- 進捗があれば github や slack で報告予定
- ご協力よろしく申し上げます