

Re:VIEW テンプレート

TechBooster 編 著

2019-04-14 版 TechBooster 発行

前書き

前書きなのだ。

免責事項

本書に記載された内容は、情報の提供のみを目的としています。したがって、本書を用いた開発、製作、運用は、必ずご自身の責任と判断によって行ってください。これらの情報による開発、製作、運用の結果について、著者はいかなる責任も負いません。

目次

前書き	2
免責事項	2
第 1 章 はじめに	4
第 2 章 コーディングスタイルで見た目を Ruby っぽくする	5
第 3 章 Enumerable のメソッドで反復処理を Ruby っぽくする	6
3.1 まず for は原則利用禁止	6
3.1.1 Ruby における反復処理の考え方	7
3.2 each 文を使う	7
第 4 章 RubySliver を受けて Ruby の全体像を掴む	8
第 5 章 おわりに	9
著者紹介	10

第 1 章

はじめに

第 2 章

コーディングスタイルで見た目を Ruby っぽくする

第 3 章

Enumerable のメソッドで反復処理を Ruby っぽくする

本章では反復処理を Ruby っぽくする方法を解説する。反復処理は大抵のプログラムで用いられる頻出の処理だ。ここを Ruby っぽくすることで、効率よくそれっぽいコードが書ける、と言う寸法だ。

まず結論から言うと、極力 Enumerable のメソッド (map とか select とか) を使おう！ということになる。

これでさっと納得した人は、もうこの章は読む必要はない。次の章に進んで欲しい。

何を言ってるか分からない、map や select は分かるが Enumerable って何よ？、という人はこの章を読むことで得られる事があると思う。是非続きを読んで欲しい。

3.1 まず for は原則利用禁止

おそらく Ruby の初学者が反復処理として一番最初に学ぶのが for 文だろう。もしくは他の言語から来た人にとっては、for 文は馴染みが深い反復処理だろう。

しかし Ruby では for 文は原則利用禁止とっておく方が良い。少なくとも for 文を使ったコードは、Ruby っぽさから最も遠い箇所に位置することになる。

前章で紹介した「Ruby Style Guide」においても、for 文は避けるように書かれている。

<https://github.com/fortissimo1997/ruby-style-guide/blob/japanese/README.ja.md#no-for-loops> “for は、どうしても使わなければいけない明確な理由が明言できる人以外は、使ってはいけません。”

3.1.1 Ruby における反復処理の考え方

なぜ for 文を使う事が推奨されていないのか？ それは for 文が Ruby における反復処理の考え方と相容れないから。

for 文の考え方は、「ある処理」を指定の回数繰り返す事

Ruby における反復処理の考え方は、オブジェクトの各要素に、「ある処理」を適用する事

図で表すと図 XX の形になる。

=== 改めて for 文を見してみる具体的に for 文のコードを用いて説明する。

図 XX を見て欲しい。for XXX で回数を指定した回数、ブロックの中身を実行している。

「ある処理」を指定の回数繰り返す事という書き方になっており、Ruby の考え方に合わない。

3.2 each 文を使う

--- 前章では見た目の観点から Ruby っぽくする方法を模索した。この章では、Ruby の機能を利用する事で、Ruby っぽいコードに近づく。

Ruby には標準で数多くの機能があるが、特に利用頻度が高いのが標準組込ライブラリ。適切なユースケースで適切な組込ライブラリのメソッドを使う事が出来れば、コードはグッと Ruby っぽさを増す、はず。

しかしその多様さゆえに、ライブラリの全てを網羅することは不可能。

そこでこの章では、繰り返し処理をする際に便利な Enumerable のメソッドに絞って説明する。繰り返し処理は。。。であり、プログラムを書いていると大抵遭遇する頻出の処理。また Ruby においても、Enumerable モジュールのメソッドとして様々な機能が手厚く提供されている箇所。

たとえ繰り返し処理だけであっても、Ruby の標準組込ライブラリを使いこなしていれば、コードの見た目はグッと Ruby っぽさに近づく。

本章では Ruby っぽい繰り返し処理を書くための流れと、よく使うメソッドを紹介する。

どんなプログラムでもほぼ確実に利用する繰り返し処理に関して、

多様な組込ライブラリは Ruby の特徴であり、適切に使う事で Ruby っぽいコードが書けるしかしその多様さ故に、全てを網羅するのは不可能。入門者には特に。そこで繰り返し処理に絞って

第 4 章

RubySliver を受けて Ruby の全体像を掴む

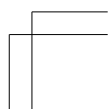
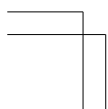
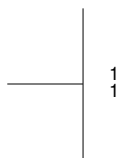
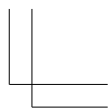
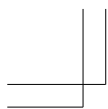
第 5 章

おわりに

著者紹介

第1章 ひつじ / @mhidaka

ひつじだよ～



Re:VIEW テンプレート

2019 年 4 月 14 日 技術書典 6 版 v1.0.0

著 者 TechBooster 編
編 集 mhidaka
発行所 TechBooster

(C) 2017-2019 TechBooster