

卒業論文

プログラミング言語 Ruby 学習の効率化アプリの
開発

関西学院大学理工学部
情報科学科 西谷研究室

27015464

大津隆輝

2019 年 3 月

目 次

第 1 章	はじめに	5
1.1	研究の目的	5
1.2	研究の動機	5
第 2 章	基本的事項	6
2.1	Emacs	6
2.2	Ruby	6
2.3	RubyGems	6
2.4	CUI(Character User Interface)	6
2.5	使用した gem ファイル	7
2.5.1	Thor	7
2.5.2	RSpec	7
2.5.3	Rubocop	7
第 3 章	ruby_learner の概要	8
3.1	Install/Uninstall	8
3.2	動作環境	8
3.3	sequential_check	8
3.3.1	教材	8
3.3.2	課題	8
3.3.3	評価	8
3.3.4	normal_mode	9
3.3.5	manual_mode	9
3.3.6	last	9
3.3.7	next	9

3.3.8	workshop	9
3.4	restore	9
3.4.1	check	9
3.4.2	open	9
3.4.3	refresh	10
3.5	pair_popup	10
3.6	install_emacs	10
3.7	emacs_key	10
3.7.1	string	10
3.7.2	image	10
3.8	theme	10
3.9	copspec	10
第 4 章	実装コードの解説	11
4.1	sequential_check	11
4.2	restore	11
4.3	pair_popup	11
第 5 章	他のソフトとの比較	12
5.1	考察	13
第 6 章	総括	14

表 目 次

5.1 他のソフトとの比較.	12
------------------------	----

图 目 次

第1章 はじめに

1.1 研究の目的

西谷研究室ではプログラミング言語 Ruby を使用して、言語学習や卒業研究を行なっている。つまり、3年生は Ruby の習得が早ければ自分の研究の為の時間を確保できる。しかし、個人で言語学習を行う際のハードルは高いものである [1]。その原因として、言語に対しての知識がない初心者にとって環境構築や学習の教材選びは非常に困難であることが要因の一つではないかと考えられる。環境構築なしで教材選びも必要ない言語学習サービスには Progate[2] や codecademy[3] 等があるが、環境構築という部分を排斥しているが故にそのサービス外での学習に一定の壁が生じているのではないかと考える。また、近年ではコードを Git で管理することで、チーム内でのコード編集の履歴をメンバーがそれぞれ確認することが出来るので、チームでの開発が管理しやすくなった [4]。このような開発では他人の書いたコードを読み解き、それに適切な形で自分のコードを書き加える能力が必要となる。そういった意味でも、近年の言語学習は動くだけのコードを書くのではなく言語が持つスタイルについてもきちんと学ぶ必要があると考える。そこで本研究では、環境構築の自動化に加え、Ruby の体系的な言語学習とプログラミングスタイルの学習ができるアプリケーションを開発し、Ruby を容易に学習できる環境を整え、実践的な学習を提供することで、個人のスキル向上の効率化を目指す。

1.2 研究の動機

本研究室に在籍していた和田によって開発された editor_learner の再開発を検討していたが、本研究室で行う開発に必要なスキルの習得に焦点を当てた開発に切り替えた。ここでのスキルとは、動くだけでなくコーディング規約に則ったコードが書けることや開発に必要な emacs や rspec といった基本的なツールの使い方の習得のことである。

第2章 基本的事項

2.1 Emacs

Emacs とはプログラミング可能なテキストエディタである。また、西谷研究室で使用を勧めているエディタである。本研究では西谷研究室の早期学習を目的としているので、本研究で用いるエディタは Emacs とする。

2.2 Ruby

Ruby とはプログラミング言語の一種である。また、西谷研究室で使用を進めている言語である。本研究では西谷研究室の早期学習を目的としているので、本研究で用いる言語は Ruby とする。

2.3 RubyGems

RubyGems とは Ruby で用いることのできるライブラリを公開することができるサービスである。公開されたサービスは `gem` コマンドを用いることで取得できる。本研究では、RubyGems のライブラリとして教育ツールを作成することで一般に公開する。

2.4 CUI(Character User Interface)

CUI とはマウス等で行う操作 (GUI) を文字のみで行う方法である。西谷研究室では Mac の Terminal で研究開発を行う。Terminal では CUI に慣れることで作業効率が上がると考えられるので、本研究では CUI での操作を基本とする。

2.5 使用した gem ファイル

2.5.1 Thor

コマンドライン作成ツールである Thor によりサブコマンドを自然言語に近い形で設定することができる。Thor についての基本的な説明は以下の通り、

2.5.2 RSpec

本研究で Ruby 言語のソースコードとして利用した教科書では Minitest が使われている。assert_equal により出力結果が正しいか判定してくれる。

2.5.3 Rubocop

自分の打ち込んだコードがコーディング規則に従っているかをチェックするのに Rubocop と呼ばれる gem を使用した。Rubocop は、

第3章 ruby_learnerの概要

3.1 Install/Uninstall

gem によるインストール方法は以下の通りである。

3.2 動作環境

Ruby の version が 2.4.0 以上でなければ動かない。理由としては、gem に格納されているパスを正しく受け渡しできないからである。2.4.0 以下で動作させるためには editor_learner の最新 version のみを入れることによって動作することが確認できている。さらに、open_terminal が MacOSX でのみ動作するので MacOSX を使用すること。

3.3 sequential_check

特別な初期設定はほとんどないが起動方法は以下の通りである，

3.3.1 教材

error が出た場合は以下の方法を試してください

3.3.2 課題

error が出た場合は以下の方法を試してください

3.3.3 評価

error が出た場合は以下の方法を試してください

3.3.4 normal_mode

error が出た場合は以下の方法を試してください

3.3.5 manual_mode

error が出た場合は以下の方法を試してください

3.3.6 last

error が出た場合は以下の方法を試してください

3.3.7 next

error が出た場合は以下の方法を試してください

3.3.8 workshop

error が出た場合は以下の方法を試してください

3.4 restore

特別な初期設定はほとんどないが起動方法は以下の通りである,

3.4.1 check

error が出た場合は以下の方法を試してください

3.4.2 open

error が出た場合は以下の方法を試してください

3.4.3 refresh

error が出た場合は以下の方法を試してください

3.5 pair_popup

特別な初期設定はほとんどないが起動方法は以下の通りである,

3.6 install_emacs

特別な初期設定はほとんどないが起動方法は以下の通りである,

3.7 emacs_key

特別な初期設定はほとんどないが起動方法は以下の通りである,

3.7.1 string

error が出た場合は以下の方法を試してください

3.7.2 image

error が出た場合は以下の方法を試してください

3.8 theme

特別な初期設定はほとんどないが起動方法は以下の通りである,

3.9 copspec

特別な初期設定はほとんどないが起動方法は以下の通りである,

第4章 実装コードの解説

本章では，今回作成したプログラムをライブラリ化し継続的な発展が可能なようにそれぞれの処理の解説を記述する．

4.1 sequential_check

`initialize` と名前に付けられたメソッドは特殊なメソッドでクラス内に記述した場合にはオブジェクトが作成されるときに自動的に呼び出される．`editor_learner` を動作したとき自動的に呼び出される部分である．

4.2 restore

`initialize` と名前に付けられたメソッドは特殊なメソッドでクラス内に記述した場合にはオブジェクトが作成されるときに自動的に呼び出される．`editor_learner` を動作したとき自動的に呼び出される部分である．

4.3 pair_popup

`initialize` と名前に付けられたメソッドは特殊なメソッドでクラス内に記述した場合にはオブジェクトが作成されるときに自動的に呼び出される．`editor_learner` を動作したとき自動的に呼び出される部分である．

第5章 他のソフトとの比較

他のタイピングソフトとの比較を行った表が以下の通りである．

表 5.1: 他のソフトとの比較．

上記のタイピングソフトは自分もよく使っていたタイピングソフトであり，評価も高いソフトである．それぞれの特徴は以下の通り．

1. PTYPING: 豊富なプログラミング言語がタイピング可能
2. e-typing: 資格取得にもつながる練習が可能. 間違いが多い箇所を指摘してくれる.
3. 寿司打: 自分が一番よく使ったソフト, GUI ベースで飽きずに継続しやすい.

それぞれの特徴があるが, 人気ソフトの中でもプログラムの実行が可能なソフトは発見できなかった. プログラマにとってコードを書いて実行しないのは, テストを受けて結果を見ないのと同義である. また, これらのソフトは全て Web 上で行なっており, editor は全く使わない. プログラマにとってコードだけ書いて editor を使って実行しないことなどない. よっていかに editor_learner がプログラマ向けのソフトかがわかる.

5.1 考察

editor_learner は様々な機能を有しており, {} や () などの約物の記入, editor 操作やプログラムの実行などが可能な点であり, editor 操作によるキーバインドの習熟が期待される.

GUI ベースのソフトに比べて継続性では劣るが, CUI ベースの editor_learner ではファイルの保存, 開閉による操作を習熟することができ作業を効率化, 高速化することができる. これらの理由により editor_learner では目的に沿った技術の向上が期待される.

第6章 総括

実際に今までたくさんのタイピングソフトやプログラムコードの打てるタイピングソフトを数多く利用してきたが，editor 操作の習熟が可能なソフトは見たことも聞いたこともなかった．実際にタイピングだけが早い学生はたくさんいるが editor 操作やキーバインドも使いこなせる学生は少なかった．本研究で開発した editor_learner によりそれらの技術も上達し，作業効率などの向上が期待される．

謝辞

本研究を行うにあたり，終始多大なるご指導，御鞭撻をいただいた西谷滋人教授に対し，深く御礼申し上げます．また，本研究の進行に伴い，様々な助力，知識の供給をいただきました西谷研究室の同輩，先輩方に心から感謝の意を示します．本当にありがとうございました．

参考文献

- [1] Andrew Hunt, David Thomas, 「達人プログラマー」, (オーム社, 2016 年).
- [2] Ruby ホームページ, <https://www.ruby-lang.org/ja/>, accessed 2018.2.8.
- [3] S. Koichiro, Rubygems のススメ, <https://qiita.com/sumyapp/items/5ec58bf3567e557c24d7>, accessed 2018.2.8.
- [4] Weblio, キーバインド, <https://www.weblio.jp/content/キーバインド>, accessed 2018.2.8.
- [5] Weblio, CUI, <https://www.weblio.jp/content/CUI> , accessed 2018.2.8.
- [6] S. Kouichiro, Thor の使い方, <https://qiita.com/succi0303/items/32560103190436c9435b>, accessed 2018.2.8.
- [7] 伊藤淳一, 「プロを目指す人のための Ruby 入門」, (技術評論社, 2017 年)
- [8] SideCI, Rubocop の使用, <http://blog-ja.sideci.com/entry/2015/03/12/160441>, accessed 2018.2.9
- [9] Ruby, 相対パス, <https://www.xmisao.com/2013/11/21/ruby-require-relative.html>, accessed 2018.2.8.