RubyとLispの切っても切れない関係

Why Emacs Lisp is an acceptable Ruby?

平成Ruby会議01

2019-12-14 #heiseirubykaigi

お前誰よ

- うさみけんた (@tadsan) / Zonu.EXE
  - GitHub/Packagistでは id: zonuexe
- ピクシブ株式会社 pixiv運営本部
- Emacs Lisper, PHPer, Rubyist
  - Emacs PHP Modeのメンテナ引き継ぎました
  - 好きなリスプはEmacs Lispです
- Qiitaに記事を書いたり変なコメントしてるよ

pixiv

お絵描きがもっと楽しくなる場所を創る

プレスリリース

Friends of Emacs-PHP development

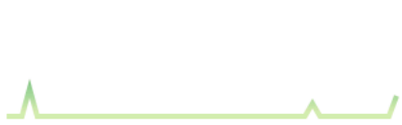
Join us!

<?php

Find a repository... Type: All Language: All Customize pinned repositories New

php-runtime.el

PHP-Emacs bridge, call PHP function from Emacs



php emacs melpa

Emacs Lisp ★ 2 1 GPL-3.0 Updated 2 days ago

phpactor.el

Interface to Phpactor (an intelligent code-completion and refactoring tool for PHP)

Emacs Lisp ★ 8 4 1 issue needs help Updated 3 days ago

php-mode

A PHP mode for GNU Emacs

Top languages

Emacs Lisp PHP

Most used topics

Manage

emacs melpa php major-mode

People

5 >

Invite someone

emacs-jp

Japan http://emacs-jp.github.com/

Find a repository... Type: All Language: All Customize pinned repositories New

reading-init.el

init.el読書会のページ

html dotfiles emacs emacs-lisp

Ruby ★ 6 1 Updated 6 days ago

helm-c-yasnippet

Helm source for yasnippet

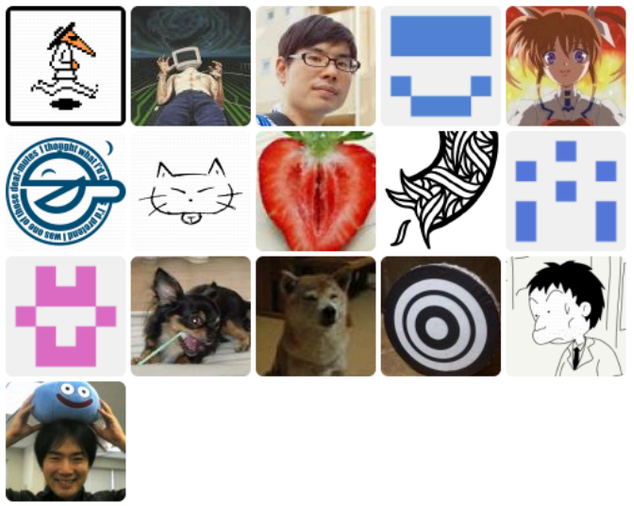
Emacs Lisp ★ 14 5 Updated on 21 Mar

replace-colorthemes

Top languages

Emacs Lisp Ruby TeX

People 37 >



いちおう2010年頃にparse.yを弄って遊んでたり闇ルビ廾會議2011で初めて人前でしゃべったり2012年くらいまでRuby札幌方面に居たりしたルビーストです

@tadsan 2012年12月21日に投稿 13796 views

Ruby Advent Calendar 2012 | 20日目

Rubyのコードを解析する...?

⚠ この記事は最終更新日から5年以上が経過しています。

はいこんにちはこんにちは。プログラミング初心者のZonu.EXEです。

今日はRuby Advent Calendar 2012の20日めです...?

ちょっと前まで自宅警備員で、一個月ほど前からアルバイトでるびーおんれーるずを書くお仕事に就いたのですが、Ruby基礎力が低すぎて困ることが多々あります。

これは重要な
ことですが

平成元年1月
生まれです

三峰結華

みつみねゆいか

ゲーム『アイドルマスターシャイニーカラーズ』のサブカル系アイドル。

pixivで「三峰結華」のイラストを見る
pixivで「三峰結華」の小説を読む
pixivで「三峰結華」のイラストを投稿する

目次 [非表示]

ちなみにスライドの本文書体は
ゲームプラットフォームenza様で
絶賛運営されているアイドルマス
ター シャイニーカラーズという
ゲームで採用されている
FOT-ハミング Stdです

事前知識

この話で言及して
いる“Lisp”は複数
の言語の総称です

C言語

```
puts("foo")
```

Ruby

```
puts "foo"
```

Lisp

```
(puts "foo")
```

(imaginary)

((((LISP))))

Common Lisp, Scheme, Clojure, Emacs Lisp

それぞれの言語は
C#とJavaくらい
離れている

あとＬＴなのでかっ

とばしてきます

さて

TextbringerでつくるTextbringer

平成Ruby会議01 2019-12-14

前田 修吾

株式会社ネットワーク応用通信研究所

この発表を聴いた人が何を持ち帰れるか

- Rubyの動的な性質を活かす方法とTextbringerステッカー

Textbringerステッカー

“

”

–TextbringerでつくるTextbringer

https://github.com/shugo/heiseirk01/blob/master/README.md

shugoさんの発表楽しかったですね

“

タッドサン
@tadsan

僕も法と秩序の番人になりたくなった #heiseirubykaigi

Translate Tweet

2:03 PM · Dec 14, 2019 · Twitter for iPhone

”

“

Cosmic Balance

”

–TextbringerでつくるTextbringer

https://github.com/shugo/heiseirk01/blob/master/README.md

もっと強い混沌の
力を手に入れたく
はないか？

GNU Emacs

An extensible, customizable, free/libre text editor — and more.

At its core is an interpreter for Emacs Lisp, a dialect of the Lisp programming language with extensions to support text editing.

↓ GNU/Linux ↓ Windows ↓ MacOS

TextbringerでつくるTextbringer

平成Ruby会議01 2019-12-14

前田 修吾

株式会社ネットワーク応用通信研究所

この発表を聴いた人が何を持ち帰れるか

- Rubyの動的な性質を活かす方法とTextbringerステッカー

Textbringerステッカー

“

”

–TextbringerでつくるTextbringer

https://github.com/shugo/heiseirk01/blob/master/README.md

Textbringerの
あらゆる機能はRuby
で実装されている

Emacsの
あらゆる機能はLisp
で実装されている

Textbringerの機能開発時はeval_bufferで編集中のコードを評価して動かしながら開発できる

Emacsの機能開発時は eval-bufferで編集中の コードを評価して 動かしながら開発できる

ためしにTextbringerのコードをEmacs Lispで書いてみよう

```ruby
def calculate_indentation
  if @buffer.current_line == 1
    return 0
  end
  @buffer.save_excursion do
    @buffer.beginning_of_line
    start_with_period = @buffer.looking_at?(/[ \t]*\./)
    bol_pos = @buffer.point
    base_indentation = beginning_of_indentation
    start_pos = @buffer.point
    start_line = @buffer.current_line
    tokens = Ripper.lex(@buffer.substring(start_pos, bol_pos))
    _, event, text = tokens.last
    if event == :on_nl
      _, event, text = tokens[-2]
    end

    # ...
```

```
(defun calculate-indentation ()
  (if (= (line-number-at-pos) 1)
      0
    (save-excursion
      (goto-char (point-min))
      (let* ((start-with-period (looking-at "[ \t]*\."))
             (bol-pos (point))
             (base-indentation (beginning-of-indentation))
             (start-pos (point))
             (start-line (line-number-at-pos))
             (tokens (ripper-lex (buffer-substring-no-properties start-pos bol-pos)))
             (rtokens (reverse tokens))
             (event (nth 1 (car rtokens)))
             (text (nth 0 (car rtokens))))
        (when (eq event :on_nl)
          (setq event (nth (1 (nth 1 rtokens))))
          (setq text (nth (1 (nth 2 rtokens)))))
        ;; ...
        ))))
```

ねっ、そっくり

でしょ？[要出典]

たぶん皆様が感じたよりはそっくり
なのですが、RubyとLispの
構文レベルの相似性というよりも
TextbringerがEmacsを参考にし
たAPIを提供していることが大きい

ところで

Ruby作者のMatzが言語おたくであることは比較的有名

Rubyist Magazine - Rubyist Hotlinks 第1回 まつもとゆきひろ

https://magazine.rubyist.net/articles/0001/0001-Hotlinks.html

RubyKaigi 2013 レポート

まつもとゆきひろさん, Rubyに影響を与えた言語とRuby開発初期を語る。〜 RubyKaigi 2013 基調講演 1日目

2013年5月31日
にく

2013年5月30日〜6月1日の3日間，お台場にある東京国際交流館にてRubyKaigi 2013が開催されています。毎日1つある基調講演をそれぞれレポートします。

Gihyo.jp RubyKaigi 2013レポート

https://gihyo.jp/news/report/01/rubykaigi2013/0001

まつもとゆきひろ 言語のしくみ Kindle版

まつもとゆきひろ (著)

★★★★☆ 2個の評価

› その他（2）の形式およびエディションを表示する

Kindle版 (電子書籍)
￥2,926
獲得ポイント: 143pt

今すぐお読みいただけます: 無料アプリ

単行本
￥3,080
獲得ポイント: 31pt プライム

￥1,385 より 17 中古品の出品
￥3,080 より 4 新品

世界中で使われているプログラミング言語「Ruby」の作者、まつもとゆきひろ氏が「言語の作り方」を初めて真正面から解説する本です。

本書のために新言語「Streem」を作りました。2年をかけて新言語を実際にデザイン・実装した取り組みを、試行錯誤の過程も含めて詳しく解説しています。

‹ 続きを読む

amazon.co.jp - まつもとゆきひろ 言語のしくみ

https://www.amazon.co.jp/dp/B01N7JZXMD/

時計の針は

平成6年に戻る

```
;;; ruby-mode.el --- Major mode for editing Ruby files -*- lexical-binding: t -*-

;; Copyright (C) 1994-2019 Free Software Foundation, Inc.

;; Authors: Yukihiro Matsumoto
;;      Nobuyoshi Nakada
;; URL: http://www.emacswiki.org/cgi-bin/wiki/RubyMode
;; Created: Fri Feb  4 14:49:13 JST 1994
;; Keywords: languages ruby
;; Version: 1.2

;; This file is part of GNU Emacs.

;; GNU Emacs is free software: you can redistribute it and/or modify
;; it under the terms of the GNU General Public License as published by
;; the Free Software Foundation, either version 3 of the License, or
;; (at your option) any later version.

;; GNU Emacs is distributed in the hope that it will be useful,
;; but WITHOUT ANY WARRANTY; without even the implied warranty of
;; MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
;; GNU General Public License for more details.

;; You should have received a copy of the GNU General Public License
;; along with GNU Emacs.  If not, see <https://www.gnu.org/licenses/>.
```

“

Emacsで言語モードを自作

そこで再び懸念になるのがオートインデントです。当時のEmacsの言語モードで、オートインデントをしてくれるのはCのようなブロックを記号で表現するものが主流でした。Pascalなどの予約語を使ってブロック構造を表現する言語のモードはキー操作でインデントを深くしたり、浅くしたりするものばかりでした。これでは、オートインデントの「気持ち良さ」を失ってしまいます。

そこで数日間Emacs　Lispと格闘しました。正規表現などを使って簡易的にRubyの文法解析を行って、endのある文法でもオートインデントができるRubyモードのプロトタイプを作成しました。これにより、endのあるEiffelっぽい文法の言語でもオートインデントが可能であることが証明されたのです。これにより、安心してRubyの文法をendを使うものにできました。逆に言えば、このときオートインデントするRuby言語モードの開発に成功していなければ、今のRubyの文法はなかったということです。

”

–「まつもとゆきひろ 言語のしくみ」 2016 まつもとゆきひろ, 日経BP

https://books.google.co.jp/books?id=h9KjDwAAQBAJ&pg=PT67

RubyにSymbolとか
when/unless文とか
Kernel.#requireがあるのは
(Emacs)Lispの影響だと思う

Matzから離れて

RuboCop

rubocop-hq / rubocop

Sponsor | Used by 77.4k | Watch 211 | Star 10.4k | Fork 2.2k

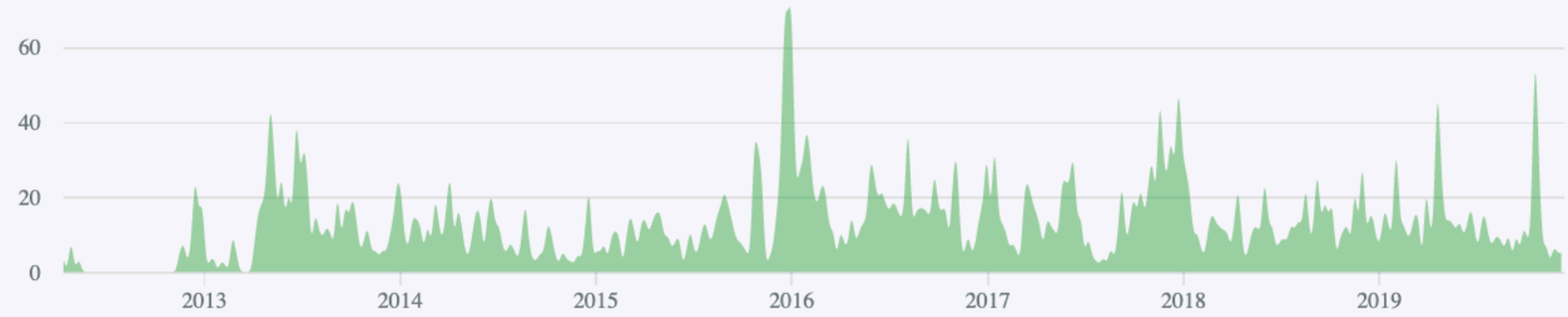
Code | Issues 122 | Pull requests 28 | Actions | Projects 0 | Wiki | Security | Insights

- Pulse
- **Contributors**
- Community
- Commits
- Code frequency
- Dependency graph
- Network
- Forks

Apr 15, 2012 – Dec 14, 2019

Contributions: **Commits**

Contributions to master, excluding merge commits



jonas054 #1

804 commits 81,538 ++ 52,035 --

bbatsov #2

456 commits 24,797 ++ 13,708 --

koic #3

436 commits 28,722 ++ 42,254 --

alexdowad #4

411 commits 19,070 ++ 8,586 --

The Clojure Style Guide

Introduction

> Role models are important.

— Officer Alex J. Murphy / RoboCop

Tip	You can find a beautiful version of this guide with much improved navigation at https://guide.clojure.style.

This Clojure style guide recommends best practices so that real-world Clojure programmers can write code that can be maintained by other real-world Clojure programmers. A style guide that reflects real-world usage gets used, and a style guide that holds to an ideal that has been rejected by the people it is supposed to help risks not getting used at all — no matter how good it is.

The guide is separated into several sections of related rules. I've tried to add the rationale behind the rules (if it's omitted, I've assumed that it's pretty obvious).

The Emacs Lisp Style Guide

> Role models are important.
>
> – Officer Alex J. Murphy / RoboCop

This Emacs Lisp style guide recommends best practices so that real-world Emacs Lisp programmers can write code that can be maintained by other real-world Emacs Lisp programmers. A style guide that reflects real-world usage gets used, and a style guide that holds to an ideal that has been rejected by the people it is supposed to help risks not getting used at all — no matter how good it is.

The guide is separated into several sections of related rules. I've tried to add the rationale behind the rules (if it's omitted, I've assumed that it's pretty obvious).

I didn't come up with all the rules out of nowhere; they are mostly based on my extensive career as a professional software engineer, feedback and suggestions from members of the Emacs Lisp community, and various highly regarded Emacs Lisp programming resources, such as "GNU Emacs Lisp Reference Manual".

Rubyから
Lispを学ぶ

つくって学ぶプログラミング言語 RubyによるScheme処理系の実装

渡辺昌寛
達人出版会

470円+税 PDF EPUB

プログラミングをより深く理解するための近道は、プログラミング言語を実装してみること。Schemeのサブセットを Rubyで実装していくことで、プログラムはどう実行されるのか、その基本がはっきり分かります。

概要 サンプル リンク用タグ

※本書はCC BYにより配布されています。上記の「買い物かごへ」ボタンからは有償で購入できます。無料で入手したい場合は、下記リンクよりダウンロードしてください。なお、有償版も無償版も内容は同一です。

- EPUB版
- PDF版

Random code snippets, projects and musings about software from Eric Kidd, a developer and entrepreneur. You're welcome to contact me!

Why Ruby is an acceptable LISP (2005)

Dec 03, 2005 • by Eric Kidd

Years ago, I looked at Ruby and decided to ignore it. Ruby wasn't as popular as Python, and it wasn't as powerful as LISP. So why should I bother?

Of course, we could turn those criteria around. What if Ruby were more popular than *LISP*, and more powerful than *Python*? Would that be enough to make Ruby interesting?

Before answering this question, we should decide what makes LISP so powerful. Paul Graham has written eloquently about LISP's virtues. But, for the sake of argument, I'd like to boil them down to two things:

1. LISP is a dense functional language.
2. LISP has programmatic macros.

As it turns out, Ruby compares well as a functional language, and it fakes macros better than I'd thought.

2010-11-02

なぜRubyは許容可能なLISPなのか

LISP Ruby

LISPの真実を読んでたら最後に出てきたので、かなり古い記事だけれども、Eric Kidd氏のWhy Ruby is an acceptable LISPを訳してみました。まつもとさんによる反応もあり、そのエントリの中で原文はほぼ要約されています。

一年前、私はRubyに注目してはいたものの、それを無視することにした。RubyはPythonほどポピュラーではないし、LISPほど強力というわけでもない。なのに何故気にかけなければならないというのか。

もちろん、これらの評価基準は考えなおすこともできる。もしRubyがLISPよりもポピュラーで、Pythonよりも強力だったらどうなるだろうか？*1 それはRubyを興味深いものにするに足るのではないか？

この疑問に答える前に、LISPを強力たらしめているものは何なのかを定義しておくべきだろう。Paul GrahamはLISPの美徳について雄弁に語ったが、ここでは議論のために次の2つに絞ることにしよう。

最後に私が5年前に書いたRubyコードを紹介して終ります

```
(Class.new do
  (define_singleton_method :let, ->(*){})
  (let
      [(display = ->(ary){
                 (puts \
                       %(\n\(my-lisps . ) +
                       %(\() +
                       (ary.map(&->(){
                          .to_s.gsub(%( ), %(\\ ))})
                                .send(:join, %( )) + %(\)\)))))}),
       (lisps = '(Arc Clojure Common\ Lisp Emacs\ Lisp Scheme TAO)'),
       (lisp-chooser = ->(){
                       (loop do
                             (let (= (.sample 3))) &&
                             ((.include? %(Matz\ Lisp).intern)?
                              (break ()):
                              (p ()))
                             (next) end)}.
                             (eval \
                              (?%+?i+ (lisps)) ||
                              (lisp-chooser)))]
  (display. (lisp-chooser))) end)
```