一つの(Emacs)Lisp、一つの世界

One Lisp in a lifetime.

Shibuya.lisp lispmeetup #75
2019年4月25日 #lispmeetup

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
  - 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

宣伝

4月
30

新元号決定！平成最後のLT大会＆PARTY

LTと共に新元号になった私

ハッシュタグ ： #engineers_lt

フォロー参加者

フォローブックマーク

募集内容	**前売券** 3000円（前払い）	先着順 9/40人
	スタッフ券 3000円（前払い）	先着順 10/10人

グループ　メンバーになる

エンジニアの登壇を応援する会

イベント数　12回
メンバー数　634人

開催前

2019/04/30(火)
20:00 ～ 2019/05/01(水) 02:00

Googleカレンダー　icsファイル

このイベントに参加できます

受付票を見る

※受付や入場方法は主催者の案内に従ってください。

申し込みキャンセル

開催日時が重複しているイベントに申し込んでいる場合、このイベントには申し込むことができません

募集期間

5月
8

東京Emacs勉強会 端午の節句

Emacs 26.2リリース記念

フォロー参加者

フォローブックマーク

募集内容		
	発表する (15分程度) 無料	先着順 6/6人
	設営・受付スタッフ 無料	先着順 3/4人
	一般参加 無料	先着順 24/40人

グループ

メンバーです

東京Emacs勉強会

イベント数 2回
メンバー数 63人

開催前

2019/05/08(水)
19:00 ～ 22:00

Googleカレンダー　icsファイル

このイベントに参加できます

受付票を見る

※受付や入場方法は主催者の案内に従ってください。

さて

Emacs

GNU Emacs を使ってる方？

Emacs Lispを
書いている方？

Emacs Lisp

シンボルの
名前空間で見ると
Lisp-2に分類される

関数がある
マクロがある
汎変数がある

なんでも
できるのでは
？？？？

そうして2012年
青年はSICPをEmacs
Lispで解こうとした

SICP読んだ方なら

おわかりですね？

Emacs Lispに末尾呼び出し最適化はない

そうして青年は Dr.RacketでSICPを解きはじめた…

（飽きっぽいので最後までは未だに解けてないです…）

Menu

- About
- Lispとは
- NScLisperとは
- Screenshots
- Download
- Event
- Members

Diary

リリカル☆Lisp開発日記

- Common Lispからweb browserを操る
- Mark-sweepのunmarkをサボる
- Common Lispのenvironmentがよく分からない
- 機械もすなる論理学を機械にやらせてみた

魔法言語 リリカル☆Lisp

About

★「魔法言語 リリカル☆Lisp」はノベルゲーム風のLispチュートリアルです。"アリサ"や"すずか"達と楽しくLispを学べます。全12話構成で各話の最後には練習問題が用意されています。Lisp処理系にはNScripter上で動作するLispインタプリタであるNScLisperを使用。別の処理系をインストールする必要はありません!!

Lispとは

★CやC++、Java、BASIC、Perl、Ruby、PHP、Python、ML、Haskellなどと同じプログラミング言語の一つです。マサチューセッツ工科大学のJohn McCarthy教授を中心とする研究グループによって開発され、1962年に発表されました。LispとはLISt Processingの略で名前通りリストの処理を得意とします。このことから人工知能の開発に多く用いられています。

NScLisperとは

★NScLisperは高橋直樹氏が開発したAVG作成用スクリプトエンジンNScripter上で動作するScheme-likeなLispインタプリタです。zickによって開発されました。外部拡張などは一切行っておらず、全てNScripter標準の命令によって実装されているところが特徴です。

リリカル☆Lispは

最後まで解きました

Lispとして見た

Emacs Lisp

ダイナミック

スコープ楽しい！

クロージャがなかった

（今はあります！）

まとめようと思ったけど山本和彦さんが音書いたブログがあった

そうして青年は Dr.RacketでSICPを解きはじめた…

2008-02-08

Emacs Lisp のダメなところ

Emacs Lisp

Emacs をこよなく愛する僕の目から、Emacs Lisp がダメだと思うところをまとめておきます。

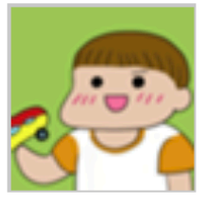
文化的な問題

Emacs Lisper の多くは、Lisp が好きで使っているのではなく、Emacs が好きだからしかたなく使っているのでしょう。本当は C で書きたいのに、無理して Lisp を利用している感じです。

そのため、Emacs に付いてくる Emacs Lisp のコードは、Lisp らしくないものがほとんどです。単に C での発想を Lisp で表現しています。

これらのコードは、読みこなせないぐらい関数が大きく、副作用のある部分とない部分が分離されていません。また高階関数を用いて、データ構造を走査するコードと実際に仕事をするコードを分離するという意識も低いようです。

山本和彦



id:kazu-yamamoto

子育て真っ最中のHaskeller。

検索

記事を検索

最新記事

関手、Applicative、Monad の法則

Emacs Lispの文化

MELPA (Milkypostman's Emacs Lisp Package Archive)

- **Up-to-date packages built on our servers from upstream source**
- **Installable in any Emacs with 'package.el'** - no local version-control tools needed
- **Curated** - no obsolete, renamed, forked or randomly hacked packages
- **Comprehensive** - more packages than any other archive
- **Automatic updates** - new commits result in new packages
- **Extensible** - contribute recipes via github, and we'll build the packages

Current List of 4,156 Packages 88,259,843 downloads to date

Enter filter terms

4156 matching package(s)

@tadsan 2018年08月20日に投稿 1163 views 編集する

MELPAにレシピを投稿するには (2018年版)

Emacs emacs-lisp melpa

5

この記事は以前書いた記事「MELPAにレシピを投稿するには」の内容を、本家の最新のアップデート[1]に併せて、別記事として翻訳したものです。

↓ ここから翻訳 *(途中)*

MELPAは「レシピ」によって構成され、レシピごとに一つの「パッケージ」を記述され、一つの専用リポジトリで管理されます。この文書ではMELPAに新しいパッケージの登録を提案するための方法を説明します。

A community-driven Emacs Lisp style guide

README.md

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".

The guide is still a work in progress; some sections are missing, others are incomplete, some rules are lacking examples, some rules don't have examples that illustrate them clearly enough. In due time these issues will be addressed — just keep them in mind for now.

A community coding style guide for the Clojure programming language

README.md

The Clojure Style Guide

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

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).

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 Clojure community, and various highly regarded Clojure programming resources, such as "Clojure Programming" and "The Joy of Clojure".

The guide is still a work in progress; some sections are missing, others are incomplete, some rules are lacking examples, some rules don't have examples that illustrate them clearly enough. In due time these issues will be addressed — just keep them in mind for now.

Emacsらしい

Lispとは

The long lost Emacs string manipulation library.

README.md

s.el

The long lost Emacs string manipulation library.

Installation

It's available on marmalade and Melpa:

```
M-x package-install s
```

Or you can just dump `s.el` in your load path somewhere.

Functions

Tweak whitespace

- s-trim `(s)`
- s-trim-left `(s)`
- s-trim-right `(s)`

バッファがファーストクラスオブジェクト

手続き的にカーソルを操作しながらバッファ内を操作できる

個人的には

とても楽しい

Emacs JP Slack teamの
スプラッシュメッセージ

“教養の言語としてLisp を教えるときに、純粋関数型言語の特徴だけをゴリゴリ押しまくるのはよい方針ではない。Lisp の雑草のような生命力の根源を見失わせる恐れがあるからだ。” by 竹内郁雄 『初めての人のためのLISP [増補改訂版]』

– tadsan

Emacsは雑草だ！！！！

私からは

以上です

補足

Emacs Lispの歴史的経緯をまじめに知りたければ

Evolution of Emacs Lisp

https://www-labs.iro.umontreal.ca/~monnier/