



# つくって学ぶプログラミング言語

RubyでSchemeの処理系を作るお話

渡辺昌寛

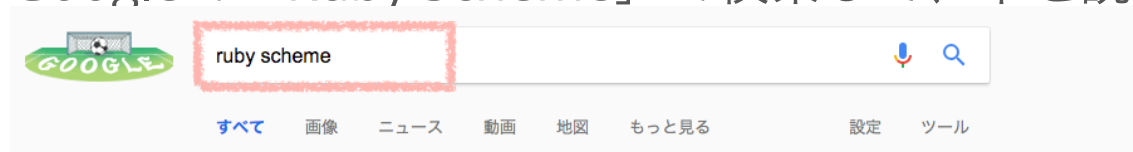
海浜幕張テック ミートアップ #2 2018/10/26

# 自己紹介

- ・ 名前: 渡辺昌寛
- ・ 勤務先: 品川
- ・ 業務: 研究開発的な部署で雑なプログラムを書きつづける
- ・ 業務経験のトピック: MIT Media Lab. 共同研究 (1.5年)
- ・ **幕張歴: 18年**

# TL;DR

- ・ プログラミング言語を理解する近道は、プログラミング言語を実装してみることに
- ・ RubyでSchemeというプログラミング言語のインタープリタを作る → プログラムはどう実行されるかわかる
- ・ Googleで「Ruby Scheme」で検索して、本を読んで理解しよう!



約 22,400,000 件 (0.36 秒)

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 達...

<https://tatsu-zine.com/books/scheme-in-ruby>

2013/04/15 - 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. 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 ...



# 何が、出力されるか、分かりますか？

JAVASCRIPT

```
fnArray = [];  
for (var i = 0; i < 5; i++) {  
    fnArray.push(function() { alert(i) });  
}  
for (var j = 0; j < 5; j++) { fnArray[j](); }
```

# 予想と合っていましたか？

どうしてこうなるか説明できますか？

```
fnArray = [];  
for (var i = 0; i < 5; i++) {  
    fnArray.push(function() { alert(i) });  
}  
for (var j = 0; j < 5; j++) { fnArray[j](); } // 5, 5, 5, 5, 5
```

JAVASCRIPT

# ちなみに...

0～4を表示したい場合

```
fnArray = [];  
for (var i = 0; i < 5; i++) {  
    fnArray.push(function(k){  
        return function(){ alert(k); }}(i));  
}  
for (var j = 0; j < 5; j++) { fnArray[j](); } // 0, 1, 2, 3, 4
```

JAVASCRIPT

# クロージャ

- ・ 上のプログラムは、クロージャを理解する必要がある
- ・ クロージャとは?
- ・ イマドキのプログラミング言語ではファーストクラスで使える

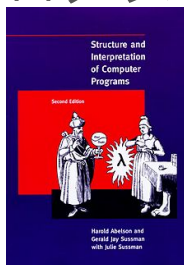
# 本を書いた経緯

- ・ 優れた技術者でも受けた教育の背景は多様
  - この辺(計算機科学)の知識がアヤしい人がいる
- ・ クロージャなど計算機科学の基本を知ってもらいたい
- ・ 最初は会社の勉強会用資料に
- ・ 達人出版会さんのご厚意で出版 🙏
  - しかもCC BYで!(後述)



# 本を読めば

- ・ プログラミング言語とは何かを深く理解することで、**プログラミングのレベルが上がる**。
- ・ “この言語が良い!”という時に、**シンタックス、ライブラリ、プログラミング言語固有の機能など、どのレベル**の機能に言及しているの**か**区別できる。
- ・ **プログラムがどのように実行されるか**理解できる。
- ・ **関数型言語の中心となる概念**を理解できる。
- ・ **λ式とクロージャの違い**が説明ができる。
- ・ 計算機科学の良書である**SICP**という本を読む準備ができる。
  - Structure and Interpretation of Computer Programs 2nd ed.: 訳本 計算機プログラムの構造と解釈 第2版



- ・ 友達に**“関数型言語の処理系を作ったことがある”**と言える 😊

# なぜプログラミング言語を理解するために処理系を作るのか?

- ・ **ものごとの本質を理解**するには、その**内部がどうなっているのかを理解**することが最も大切だと信じているから
- ・ **内部を理解する**ためには、**実際に作る**ことが一番
  - Scheme(のサブセットの処理系)をRubyで作る

# なぜSchemeか?

## なぜ関数型言語か?

- ・ 作成が簡単にも関わらず強力な機能を持っていること
- ・ なじみが少ないパラダイムである関数型のプログラミング言語を理解することにより、プログラミングの知識に幅を持てるようになる

# なぜRubyで作るのか？

- ・ 多くの人が親しんでいる手続き型言語
- ・ 機能も強力なため、本質的なことを簡潔に説明するのに役立つ

# その他: CC BYで提供

- ・ CC BY とは?
  - CC: Creative Commons
  - クリエイティブ・コモンズは、私たちのウェブサイトで**無料**でライセンスのセットを世界に提供することによって、作品を共有したいという気持ちを人々が表現する手助けをしようと考えています。(FAQより)
  - CC BYライセンスの要約:  
<https://creativecommons.org/licenses/by/3.0/deed.ja>
- ・ (渡辺によるテキトーな)意識
  - クレジット(氏名、作品タイトル、変更の有無など)を表示さえすれば、改変、再配布、商用利用が可能
- ・ 社内勉強会などなどでバンバン使ってください

# 反応



Twitter icon, user name, and location are blurred.

「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」が  
おもしろい。とりあえず第 2 章まで読め  
たけど、JavaScript 版も簡単に作れた。



Twitter icon, user name, and location are blurred.

SICP読むにあたりとても役に立った。特  
にクロージャールについてようやく理解で  
きた(と思いたい)。CC BY だけど感謝を  
込めて購入しました。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 - 達人出版会  
[tatsu-zine.com/books/scheme-i...](https://tatsu-zine.com/books/scheme-i...)

# まとめ(TL;DRの再掲)

- ・ プログラミング言語を理解する近道は、プログラミング言語を実装してみることに
- ・ RubyでSchemeというプログラミング言語のインタープリタを作る → プログラムはどう実行されるかわかる
- ・ Googleで「Ruby Scheme」で検索して、本を読んで理解しよう！



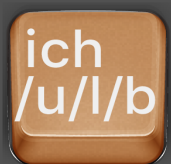
約 22,400,000 件 (0.36 秒)

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 達...

<https://tatsu-zine.com/books/scheme-in-ruby>

2013/04/15 - 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. 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 ...





ちなみに



# クロージャとは何か

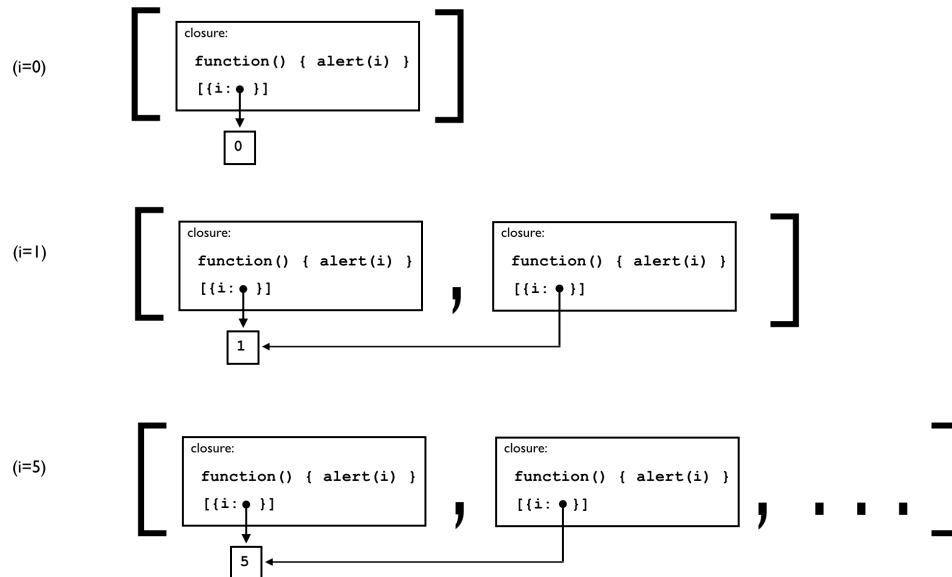
- ・ **λ式(コード)**と**環境**のペア
  - **環境**: 変数とそれに**束縛**された値の組のリスト
  - クロージャの環境は、λ式の**評価**時の環境
  - クロージャの**関数適用**は、クロージャ中の環境を仮引数を引数に束縛して拡張した上で、λ式を**評価**する
- ・ **評価**とは? **関数適用**とは? **束縛**とは?...
- ・ 本を読もう!

# すべて5が表示される場合

クロージャの環境のiが共通、かつ、副作用で値が変わる

JAVASCRIPT

```
fnArray = [];  
for (var i = 0; i < 5; i++) {  
    fnArray.push(function() { alert(i) });  
}  
for (var j = 0; j < 5; j++) { fnArray[j](); } // 5, 5, 5, 5, 5
```

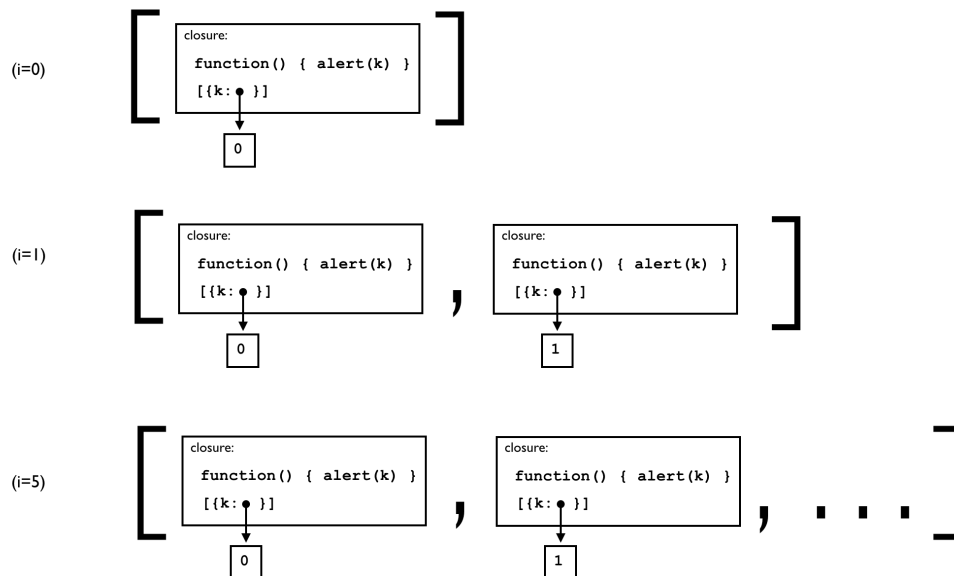


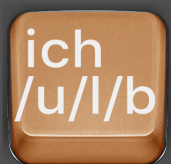
# 0～4が表示される場合

仮引数と値で環境を拡張 → 関数適用時の仮引数の値が各々で保持

JAVASCRIPT

```
fnArray = [];  
for (var i = 0; i < 5; i++) {  
    fnArray.push(function(k){  
        return function(){ alert(k); }}(i));  
}  
for (var j = 0; j < 5; j++) { fnArray[j](); } // 0, 1, 2, 3, 4
```





さらなるマニアへ

# Scheme in Scheme in Rubyにチャレンジ!

- ・ 作ったScheme処理系でSchemeの処理系を作る
  - セルフホスティングと呼ばれる
- ・ 関数型言語のプログラミング方法が学べる
- ・ なぜ、本のRubyプログラムがこんな書き方で書かれていたかが分かる
  - RubyとSchemeがほぼ1 vs 1に対応
- ・ Schemeの強力さが分かる
  - Rubyより短い
- ・ 本の最新版を [https://github.com/ichusrlocalbin/scheme\\_in\\_ruby](https://github.com/ichusrlocalbin/scheme_in_ruby) (本の**はじめに**の章の末尾にURLが記載されています) から取得 → 付録 A にコードが記載

<b>Scheme</b>
<b>Scheme</b>
<b>Ruby</b>

# Enjoy Hacking!



ichusrlocalbin@gmail.com

