

# ひまつぶし人生

前田修吾

株式会社ネットワーク応用通信研究所

# スマウルビー甲子園2025



- 3/20(金・祝)決勝大会

# 関ヶ原Ruby会議01



- 発表応募は西軍で!

# 自己紹介

---

- 前田修吾
- 1996年4月 理学部数学科→文学部哲学科
- 1997年2月 Rubyとの出会い

# 最近読んだ本

---

- 國分功一郎 『暇と退屈の倫理学』

# パスカルのことば(1)

人間の不幸などというものは、どれも人間が部屋にじっとしていられないがために起こる。部屋でじっとしていればいいのに、そうできない。そのためにわざわざ自分で不幸を招いている。

## パスカルのことば(2)

ウサギ狩りに行く人がいたらこうしてみなさい。「ウサギ狩りに行くのかい？ それなら、これやるよ」。そう言って、ウサギを手渡すのだ。

# パスカルの解決策

---

- 神への信仰

# 退屈の起源

---

- 定住革命
  - 定住生活→農耕のはじまりなどの変化
- 遊動生活では新しい環境に適応しなければならない
  - 退屈する暇がない
- 定住によって退屈が生まれた

# 暇と退屈

---

- 暇とは「何もすることのない、する必要のない時間」という客観的な条件
- 退屈とは「何かをしたいのにできないという感情や気分」という主観的な状態

# ハイデッガーの退屈

---

- 第一形式: 何かによって退屈させられること
  - 暇で退屈な状態
- 第二形式: 何かに際して退屈すること
  - 暇ではないが退屈な状態
- 第三形式: なんとなく退屈

# ハイデッガーの解決策

---

- 退屈の第三形式は、自由ということ
- 決断せよ

# 決断の奴隷になるということ

---

- 決断の瞬間とは一つの狂気である
  - セーレン・キルケゴール
- 狂気が必要な時もあるが、最初から求めるのは本末転倒
- 決断がもたらすのは心地よい奴隷状態

# 『暇と退屈の倫理学』の結論

---

- こうしなければ、ああしなければ、と思いつらう必要はない
- 贅沢を取り戻す
- 動物になる

# ウィリアム・モリス

---

革命が到来すれば、私たちは自由と暇を得る。そのときに大切なのは、その生活をどうやって飾るかだ

# アーツ・アンド・クラフツ運動

---

- 機械による大量生産の否定
- 手工芸の復興

# ひまつぶし人生

---

- これまでRuby関係でやってきたこと

# バグ報告・バグ修正

---

- バグを見つけたり直すのはたのしい
- 昔は簡単に見つかった
  - 「○○するとSEGVします」

# [ruby-list:2767] [BUG?] strip

---

- はじめて報告したバグ

```
print "ばつ".strip #=> 「つ」の2バイト目が削除される
```

# 原因

---

- isspace()の引数に負の値を渡していた

## [ruby-list:2794] Re: [BUG?] strip

---

つーか、やっぱり `isspace()` の問題だと思うなあ。 `isspace()` が負の数を受け付けないという仕様だとは聞いてないので、

## C89: 4.3 CHARACTER HANDLING

---

*In all cases the argument is an int, the value of which shall be representable as an unsigned char or shall equal the value of the macro EOF. If the argument has any other value, the behavior is undefined.*

# 既存メソッドの拡張

---

- 引数の追加など
- お手軽

## [Feature #21552]

---

- Ruby 4.0で実装したString#stripの拡張

```
p " \tfoo\t ".strip(" \t") #=> "foo"
```

- はじめてのバグ報告から28年ぶり

# 名前を考える

---

- 機能がOKでも名前がなかなか決まらない

## [ruby-dev:771] Re: printn means print and newline

---

前リクエストしたのですが、まつもとさんはPascalが嫌いということで却下でした。  
# putsならいいのかなあ。

## 他の言語を参考にする

---

- PascalやJavaより、C、Lisp、Smalltalkなどの名前が好まれがち
- writelnやprintlnは却下され、putsは採用
  - 適切な名前か疑問だが、短いのは便利

# deconstruct

---

- コンストラクタの逆関数
- 名前だけ提案
  - <https://shugo.net/jit/20120309.html#p01>

# deconstructの例

---

```
Measure = Data.define(:amount, :unit)
```

```
distance = Measure.new(10, 'km')
```

```
distance.deconstruct # => [10, "km"]
```

# パターンマッチで利用

---

```
case distance
in n, 'km' # 裏側で #deconstruct を呼ぶ
  puts "It is #{n} kilometers away"
else
  puts "Don't know how to handle it"
end
```

# 名前の由来

---

- 脱構築(déconstruction/deconstruction)
  - ジャック・デリダ

# 組み込みライブラリ機能

---

- callcc
- Refinements

# callcc

---

- Schemeのcall-with-current-continuation(call/cc)
- どういうものかわからなくてRubyに実装してみた

# callccの例

---

```
require 'continuation'  
n = callcc { |c|  
  $cont = c  
  1  
}  
p n  
$cont.call(n + 1) if n < 10
```

# Rubyの落とし方

---

- <https://magazine.rubyist.net/articles/0002/0002-RubyCore.html>

`callcc`は使わないようにしましょう。`callcc`を使うとメソッドを途中から何度も実行することができますが、実装がそのような場合を考慮していることは多くありません。

# Refinements

---

- 適用範囲を限定したモンキーパッチ
- Chad Fowlerによる命名

# Refinementsの例

---

```
module SmartDivision
  refine Integer do
    def /(other) = quo(other)
  end
end
```

```
p 1 / 2 #=> 1
```

```
using SmartDivision
```

```
p 1 / 2 #=> (1/2)
```

# 幻の機能

---

```
def with_smart_division(obj, &block)
  block.using(SmartDivision)
  block.call
end
```

```
p 1 / 2 #=> 0
with_smart_division do
  p 1 / 2 #=> (1/2)
end
```

# 文法の変更

---

- Hashの値の省略記法
- :::演算子

# Hashの値の省略記法

---

```
x = 1; y = 2;
```

```
p({x:, y:}) #=> {x: 1, y: 2}
```

# ボツになった案

---

```
x = 1; y = 2;  
# ECMAScript6風  
p({x, y}) #=> {x: 1, y: 2}
```

## :::演算子

---

- [Feature #6242] Ruby should support lists

*I've heard that Ruby is a LISP.  
LISP stands for "LISt Processing."  
Hence, Ruby should support lists.*

- エイプリルフールのネタ

# リストの例

---

```
S[1,2,3].inject(:+)    #=> 6
S[1,2,3]               #=> S[1, 2, 3]
0 ::: S[1,2,3]         #=> S[0, 1, 2, 3]
0 ::: 1 ::: 2 ::: nil #=> S[0, 1, 2]
```

## 他の人のネタ(1/3)

---

- [Feature #12240] circle number arguments by nobu
- [Feature #12241] super end by nobu
- [Feature #13393] New comment syntax by nobu

## 他の人のネタ(2/3)

---

- [Feature #15741] Ordinal parameters by mame
- [Feature #17769] Proposal: numeric coefficient syntax by mrkn
- [Feature #17768] Proposal: Downward assignments by mame

## 他の人のネタ(3/3)

---

- [Feature #19565] Ignore lower-case/upper-case Proposal by rubyFeedback
- [Feature #20404] 2pi by mame

# ライブラリ開発(1)

---

- curses
  - TUI用拡張ライブラリ
- net-ftp
  - FTPクライアントライブラリ
- net-imap
  - IMAPクライアントライブラリ

## ライブラリ開発(2)

---

- ruby-prof
  - プロファイラ
- immutable
  - 永続データ構造
- radd\_djur
  - Packrat parser combinator

## アプリケーション開発(2)

---

- mod\_ruby
  - Ruby実行用Apacheモジュール
- eruby
  - ERBみたいなツール
- ximapd
  - 検索ベースのIMAPサーバ

# アプリケーション開発(2)

---

- Textbringer
  - テキストエディタ。詳しくは福岡で
- Mournmail
  - Textbringerで動くメーラー
- textbringer-presentation
  - Textbringerで動くプレゼンツール

# 結論

---

- Rubyで生活を飾ろう