

---

# 技としての Perl

早川 真也 <hayakawa@edge.co.jp>

Shibuya.pm Technical Talks #4

2003/10/18

エッジ株式会社

## 概要

---

- 同形異義
- プログラミング言語としての本質的な違い

## 同形異議とは？

---

- 同じ表記で意味が異なるもの

## わかりやすい同形

---

- `foo () ;`
- `/foo/ ;`
- `print foo "Hi" ;`

## わかりやすい同形

---

- `*main:: {something} ;`
- `/ (bar) {3} / ;`
- `$foo->{bar} ;`
- `@hash{@keys} ;`
- `eval {something} ;`
- `LABEL {something}`
- `for {something}`
- `while {something}`
- `sub foo {somehref}`

## わかりにくい同形

---

- `foo bar;`

## 同形が多い理由

---

- Perl の文法は複雑？！

## ちょっとRubyと比較

---

- スキャナ

Perl : 約 7000行 (yylex(), etc..)

Ruby: 約 1200行 (yylex())

- パーサ

Perl : 約 1100行 (yyparse(), etc..)

Ruby: 約 5600行 (yyparse(), etc..)



## つまり

---

- Perl インタプリタはサイヤ人？！

メリット

---

**コンパイル時と実行時を細かく  
制御可能**

**トリッキーなコードを書ける**

デメリット

---

**ソースを読む際に  
実行時の様子を想定しにくい**

例

---

- 関数の宣言
- プロトタイプ

さいごに

---

言語は作り出されたものでなく  
作り出す働きである

W.Humboldt