

どうして言語は一つにならないの？ ～相対化して学ぶプログラミング言語論

Is there a "Best" programming language?
A comparative look at How they work



pixiv Inc.
USAMI Kenta

pixiv

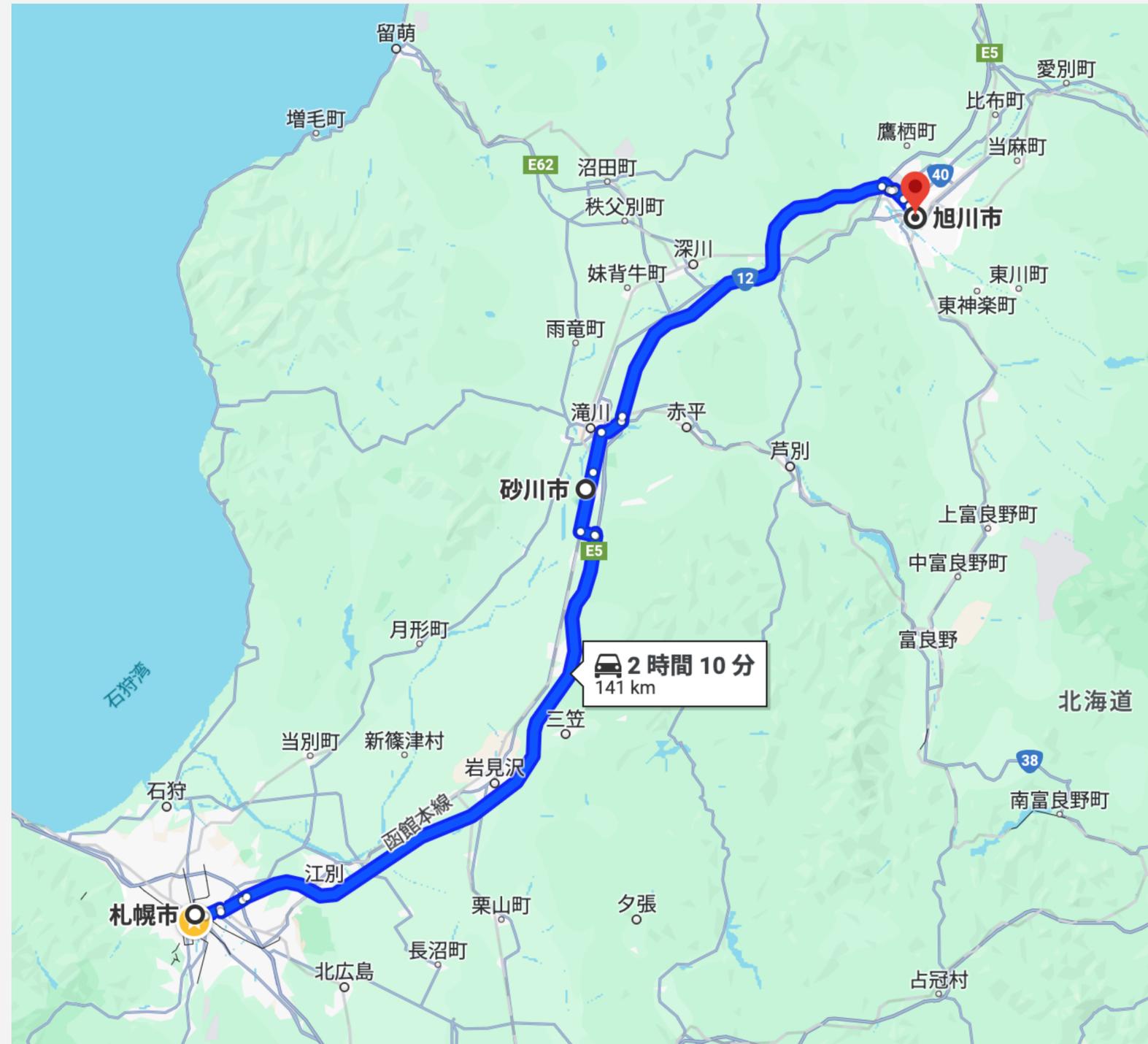
2025-07-25 #techramen25conf
旭川リサーチセンター - TechRAMEN 2025 Conference

お前誰よ



- うさみけんた (@tadsan) / Zonu.EXE / にゃんだーすわん
- ピクシブ株式会社 Platform Div > WebTechnology Team PHPer
 - 2012年末から現職、APIとかCIとかいろいろなところを見つめてきました
 - 最近チームが再編されてインフラっぽい仕事もしてます
- Emacs PHP Modeを開発しています (2017年-)
- プログラミング言語にちょっとこだわりのある素人 (spcamp2010)

出身地は札幌と旭川の間



PHPカンファレンス

純粋 vs 副作用 PHPはなぜ難しいのか？

Pure vs. Side Effects: Navigating PHP's Complexities



pixiv Inc.
USAMI Kenta

pixiv

2025-06-28 #phpcon
PHPConference Japan 2025

x

今年のTechRAMEN

なぜテキストエディタを 極めるのか

Why I Pursue Text Editors



pixiv Inc.
USAMI Kenta

pixiv

2024-07-26 #techramen24conf
TechRAMEN 2024 Conference

x

プログラミング言語史

PHPの20年とこれから

20 years of PHP and beyond



pixiv Inc.
USAMI Kenta

2023-09-15 ZORKS沖縄
(非公式)PHPカンファレンス沖縄2024 前夜祭LT会

x

あした後日祭やります

7月
27

ゆるい #techramen25conf 非公式後日祭 - コントリビ ュートまつり+

ゆるく OSS 活動と MDN Web Docs 日本語翻訳に入りこんでいこう + RDRA

主催 : tomio2480, tadsan, うーたん, zenzengood



ハッシュタグ : #ゆるあさ

[OSS Gate:終日]ビギナー枠

500円 (会場払い)

先着順

1/12人

[OSS Gate:終日]サポーター枠

500円 (会場払い)

先着順

2/4人

グループ

メンバーです

ゆるい勉強会

労力をかけないIT勉強会の運営がモットー



イベント数 **69回**

メンバー数 **249人**

開催前

2025/07/27(日)

10:00 ~ 18:00

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

募集期間

2025/07/04(金) 12:01 ~

2025/07/27(日) 18:00

今回のお題

どうして言語は一つにならないの？ ～ 相対化して学ぶプログラミング言語論



うさみけんた [@tadsan](#)

☆ 7

世の中には非常に多種多様なプログラミング言語があります。
また、注目を集める言語が毎年登場し、知っておくべき言語は増える一方です。

ある日、同僚に尋ねられました。
「なんでプログラミング言語っていっぱい種類があるの？ 最強の言語に統一しないの？」

言語を一個だけ覚えれば誰でもどんな場所でも動くコードが書けるようになる... とても素晴らしいことですが、本当にそんなにうまくいくのでしょうか？

このトークではプログラミング言語にまつわるさまざまなトピックについて、洗いざらいお話しします。

- プログラミング言語と処理系
- プログラミング言語とパラダイムと系譜
- プログラミング言語と型 (動的 vs 静的)
- プログラミング言語って誰が作っているの？ どうやって作るの？

会社の飲み会で
同僚に言われました

Q.「えっと、tadsanが
やってる言語って
なんだっけ？」

僕はプログラミング
言語が趣味みたいな
人間なので
いっぱいやっけます

僕はプログラミング
言語が趣味みたいな
人間なので
いっぱいやっています

A.「外部でよく発表
してるのはPHP」

Q.「なんで言語っていっぱいあるの？ そのうち一個にならない？」

A.「いい質問ですね！」

特に初心者は
覚えることが多い

Web開発は多言語を要求しがち

- バックエンド言語どれか
 - JavaScript(Node.js), Ruby, Python, Java, C#, PHP, ...etc.
- フロントエンド
 - HTML, JavaScript, TypeScript, CSS, ...etc.
- その他
 - SQL, 正規表現, シェルスクリプト ...etc.

どれか1つにならんか
と願いたくなる
気持ちはわかる

言語を1個だけ覚えたら
なんでも作れる！

ゲームも作れるし
システム開発もできる

そうになったら
いいのにな...

ところで

「言語」#とは

「自然言語」
人間が日常的に使う

「人工言語」
人間が目的をもって
設計した言語

Q.
(自然言語)何言語
知っていますか？

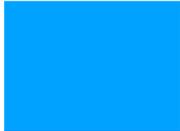
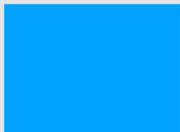
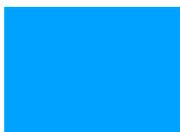
日本語、英語、中国語、韓国語
ドイツ語、フランス語、タイ語
ヒンディー語、マレー語…

言語って
結構いっぱいある

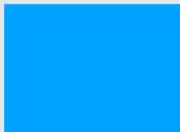
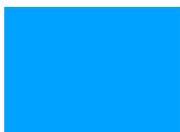
日本人は結構
「何か国語」「母国語」
って言っちやいがち

困 ≠ 言語

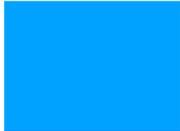
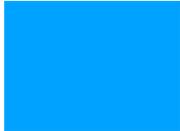
公用語	
日本	1 (事実上)
インド	22
EU加盟国	24

	公用語	エスノログ調べ (手話を除く)
日本	1 (事実上)	
インド	22	
EU加盟国	24	

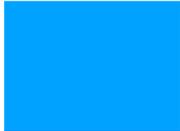
国際SILという団体による言語調査

	公用語	エスノローグ調べ (手話を除く)
日本	1 (事実上)	
インド	22	
EU加盟国	24	

国際SILという団体による言語調査

	公用語	エスノローグ調べ (手話を除く)
日本	1 (事実上)	
インド	22	717
EU加盟国	24	

国際SILという団体による言語調査

	公用語	エスノログ調べ (手話を除く)
日本	1 (事実上)	
インド	22	717
EU加盟国	24	153

国際SILという団体による言語調査

	公用語	エスノローグ調べ (手話を除く)
日本	1 (事実上)	13
インド	22	717
EU加盟国	24	153

Q.
(自然言語)何言語
しゃべれますか？

私は日本語しか
しゃべれませんが
＼(^o^)／

世界にプログラミング
言語は何種類あるで
しょうか

151種類！

…と断言してくれるオーキド博士がいてくれたらよかったです。… そのように明快な回答は出せない

(計算機言語)何言語
言えますか？

GitHubの言語検出

The screenshot shows the GitHub repository page for 'github-linguist / linguist'. The repository is public and has 171 issues, 58 pull requests, and 534 watchers. The main branch is selected, and there are 12 branches and 182 tags. The repository includes a README and a MIT license. The README content is as follows:

Linguist

Run Tests **passing**

[Open in GitHub Codespaces](#)

This library is used on GitHub.com to detect blob languages, ignore binary or vendored files, suppress generated files in diffs, and generate language breakdown graphs.

Documentation

- [How Linguist works](#)

GitHubの言語検出

594

The screenshot shows the GitHub repository page for 'github-linguist / linguist'. At the top, there is a search bar with the text 'Type / to search'. Below the search bar, there are navigation tabs for 'Code', 'Issues 171', 'Pull requests 58', 'Discussions', 'Actions', 'Security', and 'Insights'. The repository name 'linguist' is displayed with a 'Public' label. To the right, there are buttons for 'Edit Pins', 'Watch 534', and a 'Code' button. Below the repository name, there are buttons for 'main', '12 Branches', and '182 Tags'. A 'Go to file' search bar is also present. The main content area shows the 'README' and 'MIT license' files. The 'README' file is selected and displays the title 'Linguist'. Below the title, there is a 'Run Tests passing' status and a button to 'Open in GitHub Codespaces'. The description states: 'This library is used on GitHub.com to detect blob languages, ignore binary or vendored files, suppress generated files in diffs, and generate language breakdown graphs.' Under the 'Documentation' section, there is a link to 'How Linguist works'.

コンピュータ言語

- 人工的に作られた言語のうち、コンピュータで処理することを前提に設計された言語
 - データ記述言語 (HTML, XML, JSON, Markdown, ...etc.)
 - 問い合わせ言語 (SQL, SPARQL, ...etc.)
 - スタイルシート言語 (CSS, XSL, ...etc.)
 - プログラミング言語 (C言語, JavaScript, Ruby, Python, ...etc.)

このトークでは言語の種類
を厳密には区別しない…

が、分類する必要性が生じている時点で統一無理そう
という予感がしてくる

プログラムで全てを司る

- 汎用のプログラミング言語、だいたい何でも書ける
 - データも記述できる
 - 問い合わせもできる
 - スタイルシートも埋め込める
 - もちろんプログラムも書ける（それはそう）

データ記述言語もなんでもできる

- 例: GitHub Actions (YAML)
 - GitHubで処理させることをYAMLで記述する
 - 制御構造を持っていて、ほかのActionsを呼び出せる
- 構文としてはYAMLだが、これって独自のプログラミング言語では？

データと処理の世界を往復する

- Lispという言語はプログラムをS式というリスト形式で記述する
 - 関数は (format “string”) のような形式で記述する
 - マクロという言語機能によって、コンパイル過程でプログラムを生成できる
 - データとプログラムが同じ構造をとっている (同図像性)
 - eval という機能でリストをプログラムとして実行できる
- Julia, Elixirなどの言語はS式ではないが、この性質を受け継いでいる

```
(defn sum-to-n [n]
  "1からnまでの整数の総和を計算する."
  (reduce + (range 1 (inc n))))
```

```
-  
~  
~  
~
```

U:**- 2025-06-15-143920-sum.clj (4,1) [1] (Clojure

```
(defn sum-to-n [n]
  "1からnまでの整数の総和を計算する."
  (reduce + (range 1 (inc n))))
```

U:**- 2025-06-15-143920-sum.clj (4,1) [1] (Clojure

```
(defun sum-to-n (n)
  "1からnまでの整数の総和を計算する."
  (loop for i from 1 to n
        sum i))
```

U:--- 2025-06-15-144121-sum.l (2,35) [1] (Lisp Pr

```
(defn sum-to-n [n]
  "1からnまでの整数の総和を計算する."
  (reduce + (range 1 (inc n))))
```

U:**- 2025-06-15-143920-sum.clj (4,1) [1] (Clojure

```
(define (sum-to-n n)
  "1からnまでの整数の総和を計算する."
  (let loop ((i n) (acc 0))
    (if (zero? i)
        acc
        (loop (- i 1) (+ acc i)))))
```

U:**- 2025-06-15-144417-sum.scm (2,37) [1] (Scheme G

```
(defun sum-to-n (n)
  "1からnまでの整数の総和を計算する."
  (loop for i from 1 to n
        sum i))
```

U:--- 2025-06-15-144121-sum.l (2,35) [1] (Lisp Pr

```
(defn sum-to-n [n]
  "1からnまでの整数の総和を計算する."
  (reduce + (range 1 (inc n))))
```

U:**- 2025-06-15-143920-sum.clj (4,1) [1] (Clojure

```
(define (sum-to-n n)
  "1からnまでの整数の総和を計算する."
  (let loop ((i n) (acc 0))
    (if (zero? i)
        acc
        (loop (- i 1) (+ acc i)))))
```

U:**- 2025-06-15-144417-sum.scm (2,37) [1] (Scheme G

```
(defun sum-to-n (n)
  "1からnまでの整数の総和を計算する."
  (loop for i from 1 to n
        sum i))
```

U:--- 2025-06-15-144121-sum.l (2,35) [1] (Lisp Pr

```
(defun sum-to-n (n)
  "1からnまでの整数の総和を計算する."
  (let ((sum 0)
        (i 1))
    (while (<= i n)
      (setq sum (+ sum i))
      (setq i (1+ i)))
    sum))
```

F -:**- 2025-06-15-144224-sum.el (8,9) [1] (ELisp/d

eval

- evaluation (評価)
 - プログラムの実行中に文字列やリスト形式のプログラムを実行することで、柔軟な実行時処理を実現できる
 - 非常に強力だが、多用しすぎると処理が追いつかない意味不明なプログラムになりがち
 - 現代となっては(Lisp以外の世界では)使うなという風評になっている

なんでもできる
は
難しい

プログラムで全てを司る

- 汎用のプログラミング言語、だいたい何でも書ける
- 良いことも、悪いことも、なんでも記述できる
- 意図しない記述で無限ループに陥るとか
- 「大いなる力には大いなる責任が伴う」
- LispやRubyなどはこういう考えが前提にありそう

式言語 (expression language)

- ユーザーに計算式やロジックなどを入力させたいケースがある
- 汎用のプログラミング言語よりもシンプルな計算式や数学関数だけを持つ「式言語」を提供することで、プログラミング
- プログラミング言語そのものを入力させると任意コード実行(ACE: Arbitrary Code Execution)が成立しうる
- 悪意がなくても、うっかりで停止しないプログラムを書けてしまう

Excel数式

- スプレッドシートのセルに数式を書くと、計算結果が画面に表示される
 - 基本的には「命令」は書けなかった（過去形）
- 近年はExcelではPythonプログラムを書けたり、Google SpreadsheetsではJavaScriptを書けるようになった

Write once, run anywhereの夢

- Javaは「一度書けばどこでも動く」マルチプラットフォーム性を謳っていた
- 汎用的なライブラリなどは共通化できるが、アプリケーションとしては環境固有の実装が必要
- 現代でもReact Native(JavaScript), Flutter(Dart), KMP(Kotlin Multiplatform) などマルチプラットフォームの環境は多い
- WebAssembly(WASM)

プログラミング言語の パラダイム

なんちゃら型、なんちゃら指向

- 一般に普及しているプログラムは「手順」を順番に記述する（手続き型）
- C言語(手続き型)→C++, Java, C#(オブジェクト指向)
- どれもC言語を発展させたような言語と捉えられる(互換性はない)が、それぞれのアプローチは非常に違っている
- 関数を基本的な単位とするパラダイム（関数型）

PHPはどこでも65点がとれる言語

- 「現代におけるプロダクト開発とPHPを選定するワケ #phpkansai」
- By @potato4d (PHPカンファレンス関西2017)
- 小難しいことを考えずに簡単にWebアプリを書ける (HTML+ α) の言語
 - なんだかんだいってWEB+DB連携がさくっとできる言語ランタイムとしての存在感は健在
- 求められているのは素晴らしい言語ではなく簡単に使える言語