

# やわPHP カタPHP

AI Dente PHP: Finding the Perfect Resilience



pixiv Inc.  
USAMI Kenta

pixiv

# 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います
- 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)

# PHP界隈で型の話 始めて10年くらい

# pixiv inside [archive]



2016-11-11

PHP

## Phan静的解析がもたらす大PHP型検査時代

こんにちは、pixivでPHPをやってるうさみです。健全なコードベースは黙っても降ってこないの  
で、チーム全体で開発効率を高めるような改善をするのがお仕事です。

テキストエディタはmicro推しです ム(//><)ノ ☆

さる11月3日に大田区産業プラザ PiOで開催されたPHPカンファレンス 2016にて大怪獣に蹂躪さ  
れながらPhanについて30分のセッション発表をいたしましたので、その内容を紹介します！

### Phanとは

PhanはPHPの静的解析ツールです。開発元はハンドメイドのマーケットサービスを運営し、現在  
PHP作者のRasmus Lerdorf氏する米Etsy社です。もちろんRasmus Lerdorf氏も開発に参加してま  
す。

### pixiv insideとは

<https://inside.pixiv.blog/> に移転しました。こ  
ちらは2016年末までのアーカイブです。

✓ 読者です 236

[このブログについて](#)

### 月別アーカイブ

- ▼ 2016 (50)
- 2016 / 12 (27)
- 2016 / 11 (6)
- 2016 / 10 (1)
- 2016 / 9 (3)
- 2016 / 8 (1)
- 2016 / 7 (2)
- 2016 / 6 (1)
- 2016 / 5 (4)
- 2016 / 4 (1)
- 2016 / 2 (2)

レガシープロジェクトの  
ウィークポイントは  
圧倒的な「未定義挙動」

型... ドキュメント...

設計意図...

あらゆる情報が足りない

せめて型を固めておけば  
静的型付き言語に  
移植しやすくなるのでは…?

# そろそろ現実が見えてきた



# 今回のお題

PHPカンファレンス香川|2026

**採択** 2026/05/09 15:00～ 蘇鉄の間 レギュラートーク (30分)

## やわPHP、カタPHP



うさみけんた  tadsan

☆ 5  フィードバック

PHPは長く使われてきたプログラミング言語で、Webの非常に多くの場面で活用されてきました。

PHPの代替となる数多くの言語やフレームワークも提案されているなか、PHP自身やフレームワーク、エコシステムも絶えず改善され、ビジネスで使い続けられるだけの言語としての強度を保っています。

しかしながら、ツールの導入や「型」に囚われすぎ、痛めつけられることで、開発が硬直化することもままあるのではないのでしょうか。

このトークでは、ともすればAIの方がうまくPHPを書いてくれる現在において、自我を持って、楽しく自信をもってコシのあるPHPを書くための考え方についてお話しいたします。

型で消耗しないように

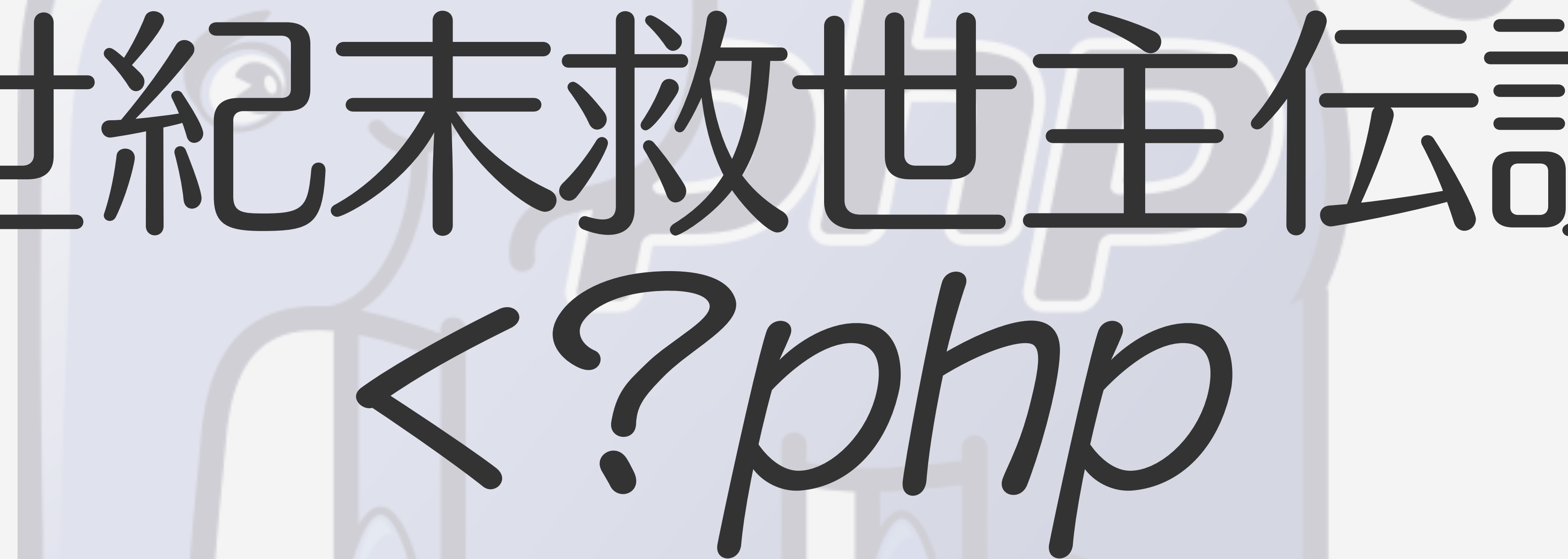
今回は具体的な  
tipsより  
考えの話がメイン



2022×年  
世界は生成AIの炎に  
包まれた！

あらゆるコードが生成され  
テストは自動化され  
すべてのコーディングが  
駆逐されたかに見えた

だがプログラマは  
死滅していなかった！



# 世紀末救世主伝説

## <?php



ほんとうに？

もう墓穴に足をつっこんでる  
自覚がないだけでは？



PHPStan Lv.maxで  
消耗してませんか？

生成AIによる開発は  
1年前とは信じられない  
ほど前進した

# 2025年2月



## CLINEに全部賭ける

2025/02/26に公開 3件

AI プログラミング Cline Idea

Cline を使い始めて2ヶ月ぐらい経った。

自分の直感として、Cline は真のイノベーションの入口であり、そして開けてはいけないパンドラの箱でもあったと思う。

ここでいう Cline は Cline型コーディングエージェントであり、広義には Devin / Cursor や Copilot Agent 等を含む話。だが、後述するように Cline でしか見えない世界がある。

その先の未来に、プログラマとしての自分はフルベットする、という話をする。



mizchi

フォロー



Programmer in the Loop

バッジを贈る

バッジを贈るとは →

### 目次

- 私たちが知っているプログラミングの終焉
- AIとプログラマが真の意味で同じ土俵に立つ

大規模言語モデル  
生成AI万歳！

業務時間中ほとんど  
コードを書いていない  
という話も珍しくない

人間が書かないなら  
別に良い言語が  
あるのでは…？

かっこいい言語には  
枚挙に暇がない

# 世界には良い言語はいっぱいある

Rust...

C#...

PowerShell...

JavaScript...

Ruby...

Java...

Common Lisp...

F#...

Python...

TypeScript...

Haskell...

Java...

Kotlin...

Swift...

GO...

Scheme...

Clojure...

Smalltalk...

今後使うべき  
プログラミング言語  
#とは

# 言語選択の 重要な要素

実行速度が速い？  
簡単に書ける？

いままででは採用が  
開発のボトルネック

○○○言語を書ける  
人が採用市場にいない

PHPを書ける人は  
割とあちこちに居る

書けるがキャリアアップの  
ためにPHP案件を好んで  
請ける人が減ってる…？

AIエージェントの力を  
借りれば別に詳しくなくても  
メンテできる時代になっ  
てしまった…

# 2025年2月



## Claude Code に向いているプログラミング言語

2026/03/05に公開 12件

- AI
- # プログラミング言語
- # ベンチマーク
- # llm
- Claude Code
- Tech

[English version →](#)

### TL;DR

Claude Code に 13 言語で簡易 git を実装させたところ、Ruby・Python・JavaScript が最速・最安・最安定でした。静的型付けの言語は 1.4~2.6 倍遅く、生成コストがかかる傾向がありました。



Yusuke Endoh

フォロー

World No.1 IOCCC player.

[http://iocco.org/winners.html#Yusuke\\_Endoh](http://iocco.org/winners.html#Yusuke_Endoh)

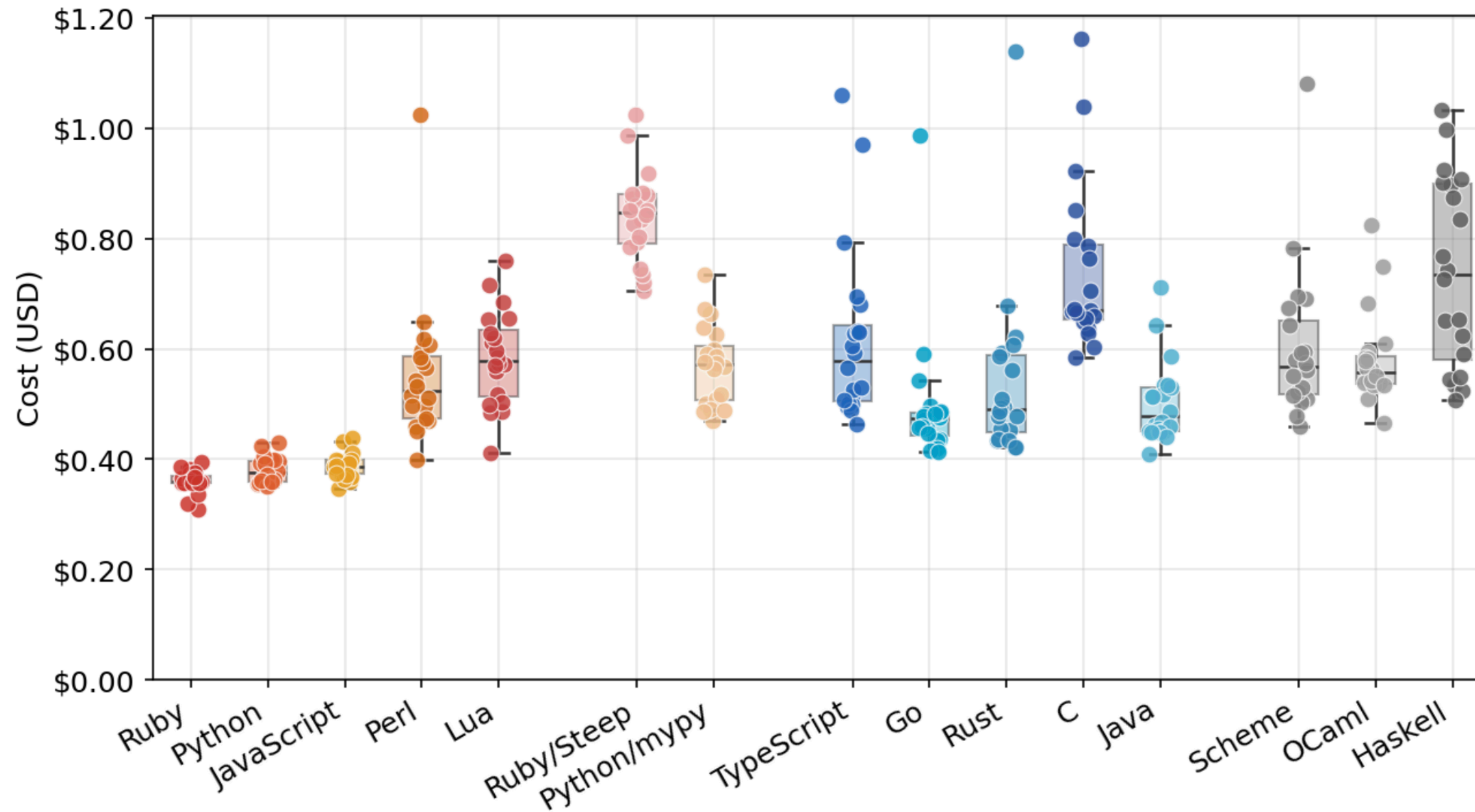
バッジを贈る

[バッジを贈るとは →](#)

# TL;DR

Claude Code に 13 言語で簡易 git を実装させたところ、Ruby・Python・JavaScript が最速・最安・最安定でした。静的型付けの言語は 1.4~2.6 倍遅く、生成コストがかかる傾向がありました。

Cost for Claude Code to Generate a Mini-Git (v1+v2, 20 trials)



# mametterさんの実験

- <https://github.com/mame/ai-coding-lang-bench>
- 簡易Gitの仕様書と受け入れテストを用意してClaudeCodeに生成させる
- 生成させるコードはv1とv2というマイルストーンに分ける
  - SPEC-v1: 基本機能(init, add, commit, log)
  - SPEC-v2: 拡張機能(status, diff, checkout, reset)
- Ruby, Pythonはプレーン、型あり版を別に計測する

# mametterさんの結果考察

- Ruby, Python, JavaScriptの群(型なし)は安定して低コスト
- Go, Rust, Javaの群も平均すると低コストだが、最悪の場合に上ぶれ
- Ruby, Pythonの型付き+TypeScriptは、型なしよりも安定しない
- そのほかの言語は分散が大きく、うまく行く場合と最悪の差が大きい

厳密な計測はできてないが  
PHP/PHPStanも  
似た傾向を示している

型は全部捨てる！  
PHPStanやめろ！

# これが21世紀の AIファースト開発



完



これでよかったの  
だろうか...

そもそもPHPは  
どんな場面で  
使われてきたのか

# 私見

重厚長大な  
コンパイル言語  
IDEのアンチテーゼ

もし言語を使えば  
シンプルなエディタで  
サクサク書けるよ！

Perl Python  
Ruby PHP

型宣言？  
そんな機能ないよ？

型なんて変数名見れば  
わかるだろ…  
常識的に考えて

スクリプト言語！  
アジャイル！

そういう時代も  
ありました

現実には甘くない…

# 2006年のPHP特集誌

amazon.co.jp  
prime

お届け先 宇佐美健太さん  
151-0053

本

Amazon.co.jpを検索



JP

本 > コンピュータ・IT > Web作成・開発 > Webプログラミング



## 超・極める! PHP

秋元 裕樹 (著) | 形式: 大型本

3.4 ★★★★★ (3)

旬のテクニック満載!PHPプロフェッショナルのための実践技術誌

PHPプログラマならまずこれを読み!

PHPプロフェッショナルが知りたいことを全部解決するために、業界随一の技術を持つ執筆陣の書き下ろし記事を集めて作りました。

特集1から3までまとめて読めば、他の本ではわからなかった“PHPを活用するためのツボ”を得ることができます。

特集◆1 大規模システム開発をモノにする!

手軽にWebアプリケーションを作成するための言語として人気の高いPHPを学び始めたはよいけれど、どうもスキルが伸びない、仕事で使うまでには腕が上達しない、と悩んでいる

著者をフォローする



すべてを表示

# PHPの すごい人たちが明かす PHPの これから

ここ数年、スクリプト言語がずいぶん元気だ。PHPも、個人がホームページを作成するための簡単言語から、バックエンドにデータベースを備えた大規模システムを開発するように、その立ち位置を変化させてきている。そこで、PHPの動向や実例に詳しい著者陣の中から5人に集まっていたいただき、今後の動向についてお話を伺った。

**藤本:**GREEも大きいほうだと思いますよ。内側はすごく複雑になってますけど。そのとき思ったんですけど、typoでひたすら苦しむ(笑)。

**全員:**うははは(笑)。

**藤本:**変数名とか、E-NOTICEで捕まえきれないのがたくさん出てくるんですよ。

**小山:**UnitTestしようよ(笑)。

**月宮:**それは、今までのシステムの変数名とか、そういうののtypoではなく？

**藤本:**いや、そういうのではなく。書き方にもよると思うんですけど、JavaっぽくPHPで大規模なのを書いていると「コンパイラがチェックしてくれるよね？」って思っているところが、通っちゃうので…で、E-NOTICEで捕まえきれないのが出てきちゃって、苦しむ。「こんなとき、ここにコンパイラがあればなあ」と思っちゃいますね。

正直つっこみどころ  
だらけだが...

われわれは20年後の  
技術を使いこなす  
チート未来人である

この記事が  
示唆するものは多い

欲しかったのは  
ガードレール

「ありえない」状態を  
食い止めるのが  
ガードレール

# 動的解析

実行してみても  
エラーが出ないか

# 静的解析

コードを分析して  
怪しいところがないか

ユニットテスト

VS

型チェック

「ありえない」状態を  
どちらかで食い止めたい

「ありえない」状態は  
すみやかにしばく

# 過去の定説

型なし言語は  
静的検査と相性が悪い

型が<sup>3</sup>ついている  
#とは

型推論規則に従って  
すべての項に矛盾なく  
型が割り当てられた状態  
…ですが何か？

## 型チェッカー

int

int

```
$a = 3;
```

```
$b = 2;
```

```
$c = pow($a, $b);
```

int

引数ok

引数ok

PHP処理系はそういう  
チェックをしないので  
「型なし」と呼ばれる

型チェックをしたとしても  
実用的な推論結果にならない

型チェッカー

mixed

string欲しいんだけど...

```
$json = $_POST['json'];  
$a = json_decode($json);  
var_dump($a['name']);
```

?array|int|float|string|bool

このキーあるの...?

# つまり

ネプチューンの!  
なんでそんなこと言ったの?TV

ホリケンが WHY?ボタン を押せば  
本人登場 or ホリケンロケ

犯人は20代から30代  
もしくは40代から50代かそれ以上の  
男もしくは女

そのため動的言語では  
テストの重要性が  
高かった

だが...今は違う！

ギョツ

# TypeScript PHPStan

単なる  
「型書けるようにした版」  
ではない

# 型理論 計算機科学の結晶

型は「ありうる」  
状態を記述する言語

適切に推論できれば  
テストを書くまでもなく  
変なコードが炙りだせる

むかしのことを  
思い出してみよう

型なんて変数名見れば  
わかるだろ…  
常識的に考えて

ちっちやいスクリプト  
／書き捨てるのコードを  
書く分には  
型なんていらなかった

型はコードの規模が  
長大になると効く

「型なんていらない」の  
サイズの閾値が  
大きく変わってしまった

コード生成ベンチマークは  
仕様と受け入れテストが  
最初からゴールまで明確

型なしRubyの  
生成コストが低いのは  
非常に理にかなっている

生成する段階では  
型はいらないというのは  
非常に理にかなってる

最近ふわっと考えてる仮説

生成されたコードに  
ガードレールとして  
追加で型を敷設する方が  
効率がいい？

## さらなる仮説

コード上に書く型は  
型宣言だけに留めた方が  
トークン効率が良いのでは…？

このあたりの話題は  
もうちょっと探究したい  
ので、そのうちどこかで

ともあれ  
普通にコードを書けば  
自然に型がついている  
状況が理想

さて

# 大規模言語モデル

得意不得意が  
はっきりしている

決定論的な処理は  
苦手だし  
圧倒的に非効率

決定論的

=

入力に対して答えが  
定められる問題

計算はAIに直接  
返答させるとはではなく  
コード生成させて実行

生成AIにスペースや  
改行位置を細かく指示  
するのは圧倒的非効率

フォーマッターを実行して  
常に同じスタイルに  
整形されるようにする

非決定的な問題を  
決定的な土俵に持ち込む

型推論規則に従って  
すべての項に矛盾なく  
型が割り当てられた状態  
…ですが何か？

型推論は典型的な  
決定論的な計算

型のことは  
型チェッカーに  
やらせる

# 静的解析ツール いろいろある

- Phan
- PHPStan
- Psalm
- Mago


拡張を書きましよう

# 型推論ロジックを記述できる

//Zenn



投稿する

 pixiv Publicationへの投稿

Edit



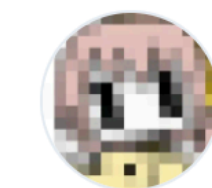
## キミにも作れるPHPStan拡張

2024/04/13に公開

 PHP  PHPStan  Tech

こんにちは！ PHPStanを活用していらっしゃいますでしょうか。

PHPStanは今日では言わずと知れた静的解析ツールですが、プラグイン機構を備えており、型付けのためのさまざまな機能を提供しています。PHPStanはできるだけコードを書かずPHPDocによる記述だけで多くのケースには型をつけられるようになっていますが、残念ながら本稿執筆時点のPHPStan 1.10系ではPHPDocだけですべてのニーズを満たすことはできないため、より深くPHPStanを利用するにはプラグインの理解が必要になります。



にゃんだーすわん



にゃーん

バッジを贈る

バッジを贈るとは →

目次

複雑な型を  
書きまくるのが  
型付けではない

PHPStanは  
普通にコードを書いて  
警告が増加しない  
レベルが適正

そこからレベルを  
上げるには  
拡張で解決したい

お使いのAIさんと  
相談しながらサクっと  
作れる時代になった

やるなら今

LLMにとって  
プログラミング言語と  
人間の言語は同列

流暢なコードは  
効率よく処理しやすい

コードに@varを  
書いてる場合ではない

理想的な変数名は  
名前から内容が  
自然に想像できて  
期待に反しない

コードは少ないほど  
見通しがよくなる

見通しがよいコードほど  
人間もAIも理解しやすい

ややこしい構造や  
紛らわしい名前は  
人間もAIも混乱する

# リーダブルコードが輝く時代

O'REILLY®  
Japan

[BLOG](#) [BOOKS](#) [EBOOKS](#) [LEARNING PLATFORM](#) [ORDER](#) [ABOUT](#)

## リーダブルコード

—より良いコードを書くためのシンプルで実践的なテクニック

Dustin Boswell、Trevor Foucher 著、角征典 訳



### TOPICS:

[Programming](#), [Business/Essay](#)

### 発行年月日:

2012年06月23日

### PRINT LENGTH:

260

### ISBN:

978-4-87311-565-8

### 原書:

[The Art of Readable Code](#)

### FORMAT:

Print PDF EPUB

オンライン学習プラットフォーム

購読可


Ebook

2,640円

[いますぐ読む](#)

[Ebookを購入する](#)

# 蘇鉄の間




ふにゃっとしない名前の付け方 ～哲学で  
茹で上げる、コシのある  
ソフトウェア設計～

by あくしも / @akshimo

PHPカンファレンス香川2026

**fortee** 

# 桐の間



## AIの揺らぎに“コシ”を与える 階層化品質設計

by wakaba / @effy\_staffs

PHPカンファレンス香川2026

