

PHPerはどこから来たのか PHPerは何者か PHPerはどこへ行くのか

Where do PHPers come from? Who are PHPers? Where are PHPers going?



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 pixiv事業本部 Webエンジニアリングチーム 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
 - 最近ではピクシブ百科事典(dic.pixiv.net)も開発しています
- 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)

お前誰よ



- 2007-2011 北海道で情報通信系の勉強をしていた
- 2010年から札幌でRubyコミュニティに出入りしてた
 - この頃からプログラミング言語論とか言語処理系に興味を持つ
- 2011年に札幌でPython勉強会などを運営していた
- 2012年から東京に引っ越した
- 2016年くらいからPHP系のカンファレンス発表など

さて

2024年は
いろいろありました

2024年に開催される全国のPHPカンファレンスのまとめ



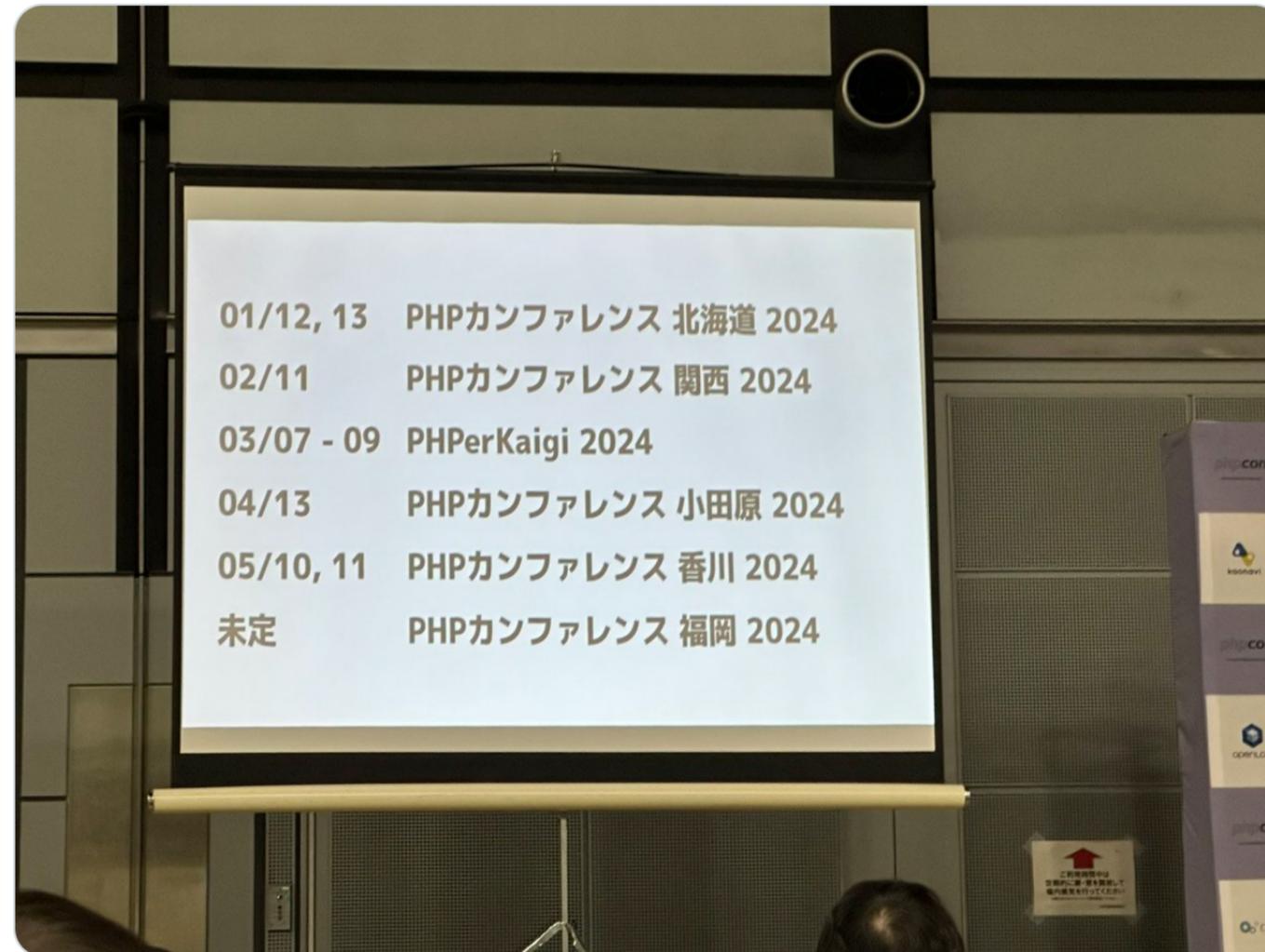
ことみん

@kotomin_m · フォローする



このスライド助かるw

[#phpcon](#) [#phpcon2023](#)



午後5:55 · 2023年10月8日



76



返信



リンクをコピー

どうして
こんなことにな
ったのか

話は
1993年に
さかかのぼる

その前に

本日のあらすじ

1. PHPはどこからきたのか
2. (日本の)PHPerは何者か
3. PHPerはどこへ行くのか

はじまります

PHPは
どこから来たのか

PHPを
作ったのは
誰でしようか

出典: フリー百科事典『ウィキペディア (Wikipedia)』

ラスマス・ラードフ (Rasmus Lerdorf, 1968年11月22日 -) はデンマーク系カナダ人のプログラマーで、プログラミング言語PHPの最初のバージョンであるPHP/FIの開発者。また、[アンディ・ガトマンズ](#)、[ゼーブ・スラスキー](#)によるPHPの後継バージョンの開発にも携わり、その発展に寄与した。

出生地はグリーンランド。1993年にカナダのウォータールー大学においてシステムデザイン工学分野での応用科学学士号 (BASc) を取得し、同大学を卒業。2002年9月より2009年11月6日までYahoo!の技術者として働く。

外部リンク [[ソースを編集](#)]

- lerdorf.com - 個人サイト



ラスマス・ラードフ



ウィキクォートにラスマス・ラードフに関する引用句集があります。

[表](#)・[話](#)・[編](#)・[歴](#)

PHP

[\[表示\]](#)

[典拠管理データベース](#)

[\[表示\]](#)

カテゴリ: [カナダのプログラマー](#) | [1968年生](#) | [存命人物](#) | [ケケタリク出身の人物](#)

出典: フリー百科事典『ウィキペディア (Wikipedia)』より引用
<https://ja.wikipedia.org/wiki/ラスマス・ラードフ> 2021-07-01T14:10:55版

[はてな匿名ダイアリー](#) > anond:20100427231539

< <http://anond.hatelabo.jp/20100...> | [フィクションに興味がある](#) >

2010-04-27

■ 伝説のPHP作者「Rasmus Lerdorf」名言集を聞くと嫌PHP厨がファビョる

- 今のPHPを作ったのは、何十人もの開発者ですよ。私は1人目の開発者だったに過ぎません。
- 問題を解くのが好きなだけで、プログラミングは大嫌いです。
- いかにプログラミングを避けるかを考えていたら、コードを再利用するためのツールとしてPHPができました。
- PHPは、歯ブラシみたいなものですね。毎日使うものですけど、だから何でしょう？誰が歯ブラシの本なんて読みたがります？
- パーザを書くのは苦手です。本当にダメなんです。今でもね。

出典: 伝説のPHP作者「Rasmus Lerdorf」名言集を聞くと嫌PHP厨がファビョる
<https://web.archive.org/web/20100501140915/http://anond.hatelabo.jp/20100427231539>

伝説の喫煙者「黒木 灰次郎」名言集を聞くと嫌煙がファビョる 1 user

引用元：<http://yutori7.2ch.net/test/read.cgi/news4vip/1271431628/>

1 名前： 以下、名無しにかわりましてVIPがお送りします 投稿日： 2010/04/17(土) 00:27:08.66 ID:jCpGzqAR0

タバコを吸うと料理の味がわからなくなる？

タバコを吸わない人にタバコの味はわかりませんよ。

2 名前： 以下、名無しにかわりましてVIPがお送りします 投稿日： 2010/04/17(土) 00:28:35.40 ID:jCpGzqAR0

僕は空気よりタバコを吸ってる時間の方が長い

5 名前： 以下、名無しにかわりましてVIPがお

一日に何本吸ってるか？

何キ口吸ってるかならわかりますけど

32 名前： 以下、名無しにかわりましてVIPがお送りします 投稿日：

あれ、これってバスタブなんですか？灰皿だと思ってました。

37 名前： 以下、名無しにかわりましてVIPがお送りします 投稿日：

喫煙者と言えないですよ。肺しか黒くなってないうちは。

38 名前： 以下、名無しにかわりましてVIPがお送りします 投稿日：

死んだら地獄に行きたいですね。天国ってきっと禁煙でしょ？

≡ ラスマス・ラードフ

🗺️ 3の言語版 ▾

本文 [トーク](#)

[閲覧](#) [編集](#) [履歴表示](#) [ツールボックス](#) ▾

ラスマス・ラードフ (Rasmus Lerdorf, 1968年11月22日 -) はデンマーク系カナダ人のプログラマーで、プログラミング言語PHPの最初のバージョンであるPHP/FIの開発者。

 ウィキペディアにも **ラスマス・ラードフ** の記事があります。

出典が明らかなもの [\[編集\]](#)

[はてな匿名ダイアリー](#) に記載された英語版の翻訳 [🔗](#) より移植

- 今のPHPを作ったのは、何十人もの開発者ですよ。私は1人目の開発者だったに過ぎません。
 - [sitepoint.com](#) [🔗](#)
- 問題を解くのが好きなだけで、プログラミングは大嫌いです。
 - [sitepoint.com](#) [🔗](#)
- いかにかプログラミングを避けるかを考えていたら、コードを再利用するためのツールとしてPHPができました。
 - [Itconversations.com](#) [🔗](#)
- PHPは、歯ブラシみたいなものですね。毎日使うものですけど、だから何でしょう？誰が歯ブラシの本なんて読みたがりません？
 - [sitepoint.com](#) [🔗](#)
- パーザを書くのは苦手です。本当にダメなんです。今でもね。
 - [Itconversations.com](#) [🔗](#)
- PHPには「protected属性」も「仮想メソッド」もありますよ。情報学科の教官が「重要だ」っていうやつは何でもね。僕自身は、こんなものどうでもいいと思ってますけど。
 - [Itconversations.com](#) [🔗](#)
- プログラミングを好む人がいるのは知ってますが、全く理解できないですね。
 - [Itconversations.com](#) [🔗](#)
- 僕はホンモノのプログラマーではありませんから、やっつけ仕事ですよ。ホンモノのプログラマーは、「動いてるように見えるけど、メモリリークだらけじゃないか。直す必要があるかもね」なんて言うでしょう？僕なら、10リクエストごとにApacheを再起動しますね。
 - [Itconversations.com](#) [🔗](#)



PHPは多くの人から疎まれ、
まじめなプログラミング言語として
考察対象とされていない節がある

正しい

だが、そうじゃない



Rasmus Lerdorf

Yahoo

PHP on Hormones

MySQL Conference

44 minutes, 20.5mb, recorded 2007-04-26

Topics: [Software Development](#)



[Download Audio](#)

In 1993, when Rasmus first saw the Mosaic Web browser, he knew that the Internet would be the platform of choice. But his employer, a Brazilian company, did not pay heed so he quit to return to Canada to do consulting work. During this six-month period, he found himself repetitively writing the same CGI programs in C. To avoid repetition, he collected his library of C programs and added a template parser that parsed HTML and made calls to his C routines. Thus was born the first version of PHP.



Rasmus believes there are four kinds of programmers. First, the pragmatic ones who are just after solving their own problems. The second kind finds programming as a means of self-expression, like an artist finds self-expression in his art-work. The third are the real programmers who enjoy programming for its own sake because it creates a hormone called oxytocin in them, and the fourth are the open source zealots who wish to change the world. He claims to be of the first kind. He programs to solve his problem and then moves on. He confesses that he created PHP purely to serve his own interest, to solve his own set of problems. He made the source publicly available so others could benefit from it. That set the ball rolling. Today, PHP runs a considerable number of some of the largest websites on the planet.

Hear the story of the evolution of PHP from being a purely procedural language to its current state of a full-fledged object oriented language, from its creator Rasmus Lerdorf. In this presentation, Rasmus also talks about the performance of PHP, profiling, security issues and vulnerabilities that websites are prone to, how to tackle them to some extent, and about his love for API. The [slides for this session](#) contain code snippets of the old and new PHP versions and also of the Flickr and Yahoo! Maps API examples.

Rasmus Lerdorf is the original creator of the PHP programming language, the mod_info Apache module, and the ANSI92 SQL-defying LIMIT clause in mSQL 1.x which has now crept into both MySQL and PostgreSQL.

Prior to joining Yahoo! as an infrastructure engineer in 2002, he was at a string of companies including Linuxcare, IBM, and Bell Canada working on internet technologies.

Resources

- [Slides for the session](#)
- [Rasmus Lerdorf's website](#)

Rasmus LerdorfはMySQLとPostgreSQLの両方に導入されているSQL標準(ANSI92 SQL)を無視したmSQLのLIMIT句のオリジナル作者です

This free podcast is from our [MySQL Conference](#) series.

皆さん、おはようございます。こんな時間に起きてくださってありがとうございます。Rasmus は PHP の父です。これはすごいことです。私たちが話しているのは、地球上でかなりの数の大規模な Web サイトを運営している PHP です。宣伝活動が盛んであるにもかかわらず、他の言語にも予算やマーケティング担当者などが集まっています。

PHP は言語として独自の地位を確立しており、オープンソースで見られる最もダイナミックかつクリエイティブな要素の 1 つであると考えられます。それでは、これ以上は省略して、Rasmus を紹介しましょう。

[拍手]



PHP on Hormones

MySQL Conference

Apr.26, 2007. Santa Clara

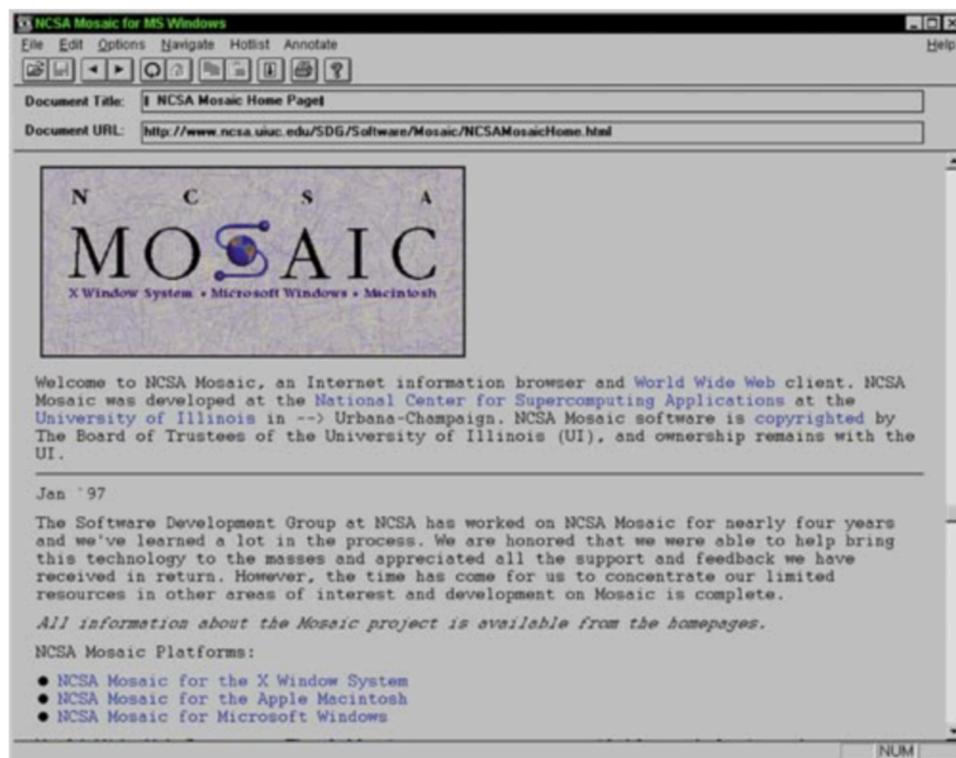
Rasmus Lerdorf <rasmus@php.net>

<http://talks.php.net/show/mysql07key>

そして、初めて Mosaic Browser を見たとき、私はブラジルの会社のオーナーに電話をかけて、「みんな、今すぐ会社全体を変えて、インターネット以外のことはやらないようにしなきゃいけない。このブラウザは世界を席巻するだろうから」と言いました。彼らは同意しなかったので、私は辞めました。



1993



私はカナダに戻り、ウェブコンサルティングを始めました。当時のウェブページはこんな感じでした。



1993

```
<HTML>
  <HEAD>
    <TITLE>My Personal Home Page</TITLE>
  </HEAD>
  <BODY>
    This is my cool page<P>
    And look at my counter<P>
    <IMG SRC="/cgi-bin/counter.pl">
  </BODY>
</HTML>
```

それらの背後にあるコード。背後にあるわけではなく、こんな感じです。ホームページがあり、誰もがページの下部にくだらないカウンターを持っていました。

彼らは、「これだけの訪問数、これはちょっとしたグラフィックです」と言いました。私は通常 Perl で書いていたので、このような小さなcgi-binカウンターでした。本当に、あまり印象的ではありませんでした。そして、それらのカウンターは、それをそこに置いた人以外には意味がありませんでした。

当時、私が雇われたのは、動的なウェブページを書いたり、バックエンドのデータベースや、企業や大学がかつてオンラインに公開していたあらゆる情報を取得したりするためでした。彼らは、その構築を手伝うために私を雇ったのです。そして、私は同じコードを何度も何度も書くことになりました。

私は C で多くの CGI プログラムを書きました。そしてしばらくして、同じ C コードを何度も書き続けることに本当に疲れてしまいました。

そこで、過去 6 か月ほどの間に書いた C コードをすべて収集し、ライブラリにまとめました。そして、HTML テンプレートを解析して C コードを呼び出す、非常にシンプルなテンプレート パーサーを考案しました。

これは基本的に 1994 年頃の PHP バージョン 1 です。



1994

```
<!--getenv HTTP_USER_AGENT-->
<!--ifsubstr $exec_result Mozilla-->
Hey, you are using Netscape!<p>
<!--endif-->

<!--sql database select * from table where user='$username'-->

<!--ifless $numentries 1-->
Sorry, that record does not exist<p>
<!--endif exit-->

Welcome <!--$user-->!<p>
You have <!--$index:0--> credits left in your account.<p>

<!--include /text/footer.html-->
```

今のところ、PHP の見た目は現在のものとあまり似ていません。しかし、見れば何をしているのかは大体わかります。

たとえば `getenv`、ユーザー エージェントを取得するには、Mozilla かどうかを確認します。すると、彼は「ねえ、あなたは Netscape を使っていますね」と言いました。

そこには SQL クエリがあります。最初のバージョンにも SQL クエリがありました。当時、私はオーストラリアの David Hughes という人が作成した Mini SQL というデータベースを使用していました。

SQL クエリも少し使用しました。ロジックも少し使用しました。少ない場合は、中括弧がないことがわかります。

セミコロンはありません。非常にシンプルなテンプレート言語です。

当時は、基本的にこれを使用していたのは私だけでした。気がくぐれに変更を加えることができました。毎日、言語を大幅に変更していました。そして、それほど時間はかかりませんでした。

1993年時点で存在したデータベースソフトウェア。
フリーソフトウェアではないがソースコードが
公開されていて、MySQLなどに影響を及ぼした。

1995年、1995年中頃までに、PHPは次のようになりました。これは、実際のところ、今日のPHPとそれほど変わりません。



1995

```
<?
$name = "bob";
$db = "db";
$result = mysql($db,"select * from table where firstname='$name'");
$num = mysql_numrows($result);
echo "$num records found!<p>";
$i=0;
while($i<$num);
    echo mysql_result($result,$i,"fullname");
    echo "<br>";
    echo mysql_result($result,$i,"address");
    echo "<br>";
    $i++;
endwhile;
>
```

ここではいくつかの違いがあることが簡単にわかります。中括弧がないことに気がつくでしょう。

この `while` ループは、後ろにセミコロンが付いているので、構文エラーのように見えます。しかし、私はパーサーを書くのが本当に苦手でした。今でもパーサーを書くのは本当に苦手です。

`while` つまり、`and while . if and`、などと書かれたトークンがあると、読みやすくなります `if`。つまり、どの構造を閉じているのかが実際にわかりました。つまり、中括弧はまったく必要ありませんでした。

2005

```
<?php
class db {
    protected static $dbh = false;

    function connect() {
        self::$dbh = new PDO('mysql:host=localhost;dbname=test','user','pass');
        self::$dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    }
}

class items extends db {
    function load($name) {
        if(!self::$dbh) $this->connect();
        try {
            if(!self::$dbh) $this->connect();
            $stmt = self::$dbh->prepare("SELECT * FROM items WHERE firstname=:name
                                         ORDER by ctime desc");

            $ret = $stmt->execute(array('name'=>$name));
        } catch (PDOException $e) {
            die($e->getMessage());
        }
    }
}
```

今から 10 年後、いや、それより 12 年後だと思えます。しかし、2005 年の時点では、同じコードがこのような状態になっています。これが 10 年間の開発に値するかどうかは、私にはよくわかりません。

しかし、コンピュータサイエンスの人々は、これですべてがオブジェクト指向になったことに満足しています。これはクールだからです。保護されたプロパティのようなものがあります。抽象メソッドがあります。コンピュータサイエンスの先生が本当に使用すべきだと言ったものがすべてあります。実際のところ、私はこのようなくだらないことにはまったく関心がありません。

私は問題を解決するツールを作ることに関心があります。プログラミングは大嫌いです。プログラミングを避けるために PHP を書きました。できるだけ早く問題を解決したいのです。そして、私が PHP を書いた方法は、私が知っていることに基づいていることに気づきました。そして、他の人がすでに知っているだろうと思っていたことに基づいており、学習曲線が非常に暗くなっています。

今日の大学を出た若者は、これが彼らの知識です。彼らはオブジェクト指向プログラミングを知っています。これが彼らにとって馴染み深いものです。手続き型プログラミングとは何かを説明しなければならない人たちがいました。

今日の大学を出た若者は、これが彼らの知識です。彼らはオブジェクト指向プログラミングを知っています。これが彼らにとって馴染み深いものです。手続き型プログラミングとは何かを説明しなければならない人たちがいました。

私は彼らに関数呼び出しを見せました。それで、クラスはどこにあるのですか？これはどこから来るのですか？つまり、私はこの手続き型呼び出しを理解していません。

それはただそこにあり、理解できないことをしているだけです。私にとっては、それはちょっと目を見張るものでした。そして、PHPが少し変化し、現代のオブジェクト指向言語に似たものへと進化したのは、人々がそのようなプログラミング方法を期待しているからです。

私はまさにこのスライドの最初のタイプ、つまり純粋に自己利益を追求するタイプです。問題を解決するツールが必要でした。私の興味はツールにはあまりありません。私の興味は問題を解決することにあります。

当時私を雇ってくれたさまざまな人たち、中でも大きなトロント大学は、Web 駆動型の大規模なダイヤルアップシステムの構築を私に依頼しました。彼らは私がどのようにそれを実現するかは気にしませんでした。彼らが求めていたのは、ソリューションだけでした。

ですから、途中でツールをオープンソース化したいと思ったとしても、それはまったく問題ありません。PHP の多くは、実際にはトロント大学向けに書かれたものです。ですから、私にとっては、それは純粋に自己利益のためでした。

オープンソースにしたのは、他の人たちが私に「どうやってそれをやったの？」と聞いてきたからです。私は「私はこうやってやった。これが私のツールだ」と答えました。彼らはそれを採用し、私にパッチを送り始めました。

トロント大学は、私が驚くべき速さでコードを書いていたことを素晴らしいと考えました。寝ている間もバグを修正していました。毎朝メールボックスを見ると、世界中から、自分が抱えていることすら知らなかったバグの修正が3～4件届いている、と気づくのです。素晴らしいシステムだと思いました。

同じ問題を抱える世界中の人たちと集まってみたらどうでしょう。そして、私たち全員が集まって、この共通の問題を解決するツールに取り組みます。そして、私たち一人ひとりが、自分の小さな世界の中では優秀に見えました。なぜなら、私たちはこのツールを非常に速く構築し、世界中の人々がそれに取り組んでいるという規模を利用していたからです。

ですから、これはソフトウェア開発、つまり私が必要としていたツールの開発にとって本当に良いモデルだと思いました。

他の人には別の考えがあります。つまり、他の人は自己表現のために貢献しているのです。実際にプログラミングが好きな人もいます。なぜ彼らがプログラミングが好きなのかは理解できません。

しかし、プログラミングが好きな人は、画家が絵を描いて他の人に見せるのと同じように、自分を表現したいと思っています。ミュージシャンは、それが自分を表現する方法なので、人前で演奏したいと思っています。プログラマーは、自分のコードを他の人に見せたいと思っています。

- 「やあ、私は優秀なプログラマーだ。」
- 「このコードを見てください。」
- 「この問題を私が解決したこの複雑な方法を見てください。」
- 「私が考え出したこのクールなアルゴリズムを見てください。」

そして、それが彼らの自己表現の仕方なのです。そして、私はそれが素晴らしいことだと思います。なぜなら、本物のプログラマーがいなければ、PHP は今日存在していなかったでしょうから。

そして、それが彼らの自己表現の仕方なのです。そして、私はそれが素晴らしいことだと思います。なぜなら、本物のプログラマーがいなければ、PHP は今日存在していなかったでしょうから。

私は本物のプログラマーではありません。動作するまでいろいろと試して、それから先に進みます。

本物のプログラマーはそこに座って「はい、動作します」と言うでしょう。しかし、メモリがあらゆる場所でリークしています。おそらく、これを修正する必要があります。

10 リクエストごとに Apache を再起動するだけにしようと思います。リークを修正しましょう。私は細かいことにこだわるタイプではありません。つまり、動作させるだけです。いくつかアイデアを思いつきます。

他の人たちがこれを本当に役に立つものにしてくれます。

ホルモン、これはオキシトシンというとても素晴らしいホルモンです。基本的には自然の信頼ホルモンです。そして、他の人とあまりうまく交流できない子供やオタクがたくさんいます。しかし、これらの子供は他の人と交流する必要があります。そして、他の人と交流すると、オキシトシンというホルモンが分泌され、気分が良くなります。これは、男性と女性の両方で、出産時やオーガズム時に分泌されるのと同じホルモンです。そして、他の人と交流すると、このホルモンも分泌されます。

たとえば、オタクたちが World of Warcraft をプレイしているのを目にするでしょう。彼らをオタクと呼ぶつもりはありませんが、内向的な子供たちが World of Warcraft をプレイしているのをよく見かけます。これが特徴の1つです。これらの大規模でインタラクティブなオンラインゲームでは、子供たちは実際に他の人と交流します。そして、それが彼らのオキシトシンを分泌させます。

そして、ゲームにそれほど興味がない他の子供たちは、実際にオープンソースプロジェクトに参加して貢献し、そのようにして他の人と話します。それが彼らの仲間との交流の方法です。

彼らはバーに行ったり、友達と野球をしたりしていると思います。コンピューターの前に座ってオープンソースの仕事をしています。そして、世界を良くするためだけに何かをする変人もいます。

たとえば、私は世界をより良くするためにオープンソースに取り組むつもりです。オープンソースの多くは世界をより良くすると思います。これは素晴らしい副次効果です。しかし、PHP で世界をより良くしようと決めたと言ったら、それは本当に嘘になります。

いいえ。ツールが必要でした。他の人はそれが便利だと思いました。素晴らしい。それで、これはあなたが気にする何かとどのように関係しているのでしょうか？

オープンソース プロジェクトを運営する上で、この言葉はまさに真実であると思います。すべてを自分のことだけに考えていては、まともなオープンソース プロジェクトを運営することはできません。オープンソース プロジェクトに貢献する人々を中心に据える必要があります。

このプロジェクトに関わっている人たちが自分自身についてどう感じているかを考えなければなりません。ある程度の所有権を放棄しなければなりません。これは彼らのプロジェクトであり、みんなで一緒に取り組んでいるのだという意識を彼らに与えなければなりません。そして、それは私にとって当初最も困難なことの1つでした。なぜなら、それは私の仕事だったからです。

PHP は私の宝物でした。コントロールを放棄し、基本的に一步下がって、誰に対しても特別な権限を持つことなく、他のみんなと同じように PHP に貢献していました。

もし誰かが私に反対したら、私たち3人と2人が「いやいや、私たちはこうしなければなりません」と言ったら、私は完全に反対しました。

それは問題ではありません。私が始めたからといって、私がそれに関して大きな発言権を持つわけではありません。そしてそれは最初よりも少し大変でした。しかし、私は本当に怠け者です。

そして、もし他の人々がそのことについて本当に強い感情を抱いているなら、私は彼らと戦うのが面倒くさすぎるのです。皆さんが少しは気にしているかもしれない世界でも同じことが言えると思います。あちこちにあるこれらのサイトはすべて、同じ特徴を示していると思います。

Q. つまわりPHPって
何だったの？

A. 異常に行動力があるRasmus
さんが自分のためにPHPを作っ
て、何回か書き直した後に利己的
な動機でソースコード公開したら
世界的に開発が盛り上がった

 Edit



PHP Early Days (誕生から4まで)

2023/09/17に公開

 PHP  # php4  Tech

この記事は何か

某社の某雑誌に寄稿しようと思って書いた記事の初稿... だったのですが、テーマじゃないところに紙幅を割きすぎて本来のテーマを大幅に逸脱した上に、書いてから三年ほど置いてしまったものです (執筆自体はまだ諦めてません)。

ちょうど「LLイベント」ことLearn Languages 2023というイベントで「PHPの20年とこれから」としてPHPの歴史を話したので、せっかくなのでこの期に公開しましょう。

 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

- この記事は何か
- ▼ 何かという話について

イスラエルの大学生
ZeevとAndiが仲間になっ
て大革新を遂げる



≡ 優しい終身の独裁者

文 18の言語版

ページ ノート

閲覧 ソースを編集 履歴表示 ☆ ツール

出典: フリー百科事典『ウィキペディア (Wikipedia)』

優しい終身の独裁者（やさしいしゅうしんのどくさいしゃ、*Benevolent Dictator For Life*、省略形**BDFL**、より直訳的には「慈悲深き終身の**独裁官**」）とは、**オープンソース**ソフトウェア開発プロジェクトの少数の**リーダー**に与えられる**称号**である。一般的には、コミュニティ内で論議、論争が発生した際に最終的な仲裁を行う権利を持つプロジェクト創設者であることが多い。独裁者という言葉から勘違いされることがあるが、あくまで論争が発生したときに仲裁を行うのであって、ソフトウェアの仕様の最終決定者のことではなく、ソフトウェアの権利を独占している者のことでもない。

この用語は1995年に誕生し、**Pythonプログラミング言語**作者の**ガイド・ヴァンロッサム**を指していた^{[1][2]}。ヴァンロッサムは、**CNRI**（**英語版**）に雇用されて間もなく、Python開発とワークショップの監督を行う準公式グループを設立することを企画するミーティングについて**電子メール**でやり取りを行っていた。その電子メールに返信した**ケン・マンハイマー**（**英語版**）のメールにこの用語が登場する^[1]。

BDFLは、オープンソースのリーダーに対するより良く知られた用語、「**優しい独裁者(benevolent dictator)**」（**英語版**）^[3]と混同すべきではない。こちらは**エリック・レイモンド**のエッセイ、「**ノウアスフィアの開墾**」（**英語版**）で普及した^[4]。**ハッカー文化**にまつわる話題のひとつとして、レイモンドは、いかにして、オープンソースの性質のせいで「独裁制」というものが慈悲深くなければならないのかを詳しく調べている。なぜなら、意見が完全に違ってしまえば、新しいリーダーがプロジェクトを**フォーク**することにつながるからである。

優しい終身の独裁者と時折見なされる人物の例 [ソースを編集]

出典: フリー百科事典『ウィキペディア(Wikipedia)』より引用
<https://ja.wikipedia.org/wiki/ラスマス・ラードフ> 2021-07-01T14:10:55版

Rasmusは一般的な意味でのBDDFLではない



513



199



 @tadsan (園島 めめ)

P++: 銀河に平和をもたらすための奇策と決着

PHP p++

最終更新日 2019年08月15日 投稿日 2019年08月14日 45634 views

PHP 8から、PHPは「PHP」と「P++」という2つの言語を提供するようになる

というキャッチーな紹介をする [P++: 静的型付けをめざすPHP](#) という記事がそれなりに話題になり、このニュースは目覚しく革新的な内容で、多くのひとの目を引き付けました。

PHP RFC: Fix up BCMath Number Class / Change GMP bool cast behavior

- Version: 1
- Date: 2024-06-30
- Author: Saki Takamachi (saki@php.net)
- Status: Voting
- First Published at: https://wiki.php.net/rfc/fix_up_bcmath_number_class



Introduction

Regarding [RFC: Support object type in BCMath](#), I received several comments and suggestions, so I created this RFC to respond to them.

Also, in those discussions, there were suggestions regarding the behavior of casting GMP classes to bool, and for procedural efficiency, I will combine these into one RFC. (Each vote will be taken separately.)

Proposal

Fix up BCMath Number Class

Real name	Yes	No
crell (crell)	✓	
derick (derick)	✓	
ericmann (ericmann)	✓	
galvao (galvao)	✓	
girgias (girgias)	✓	
jimw (jimw)	✓	
kguest (kguest)	✓	
petk (petk)	✓	
pierrick (pierrick)	✓	
ramsey (ramsey)	✓	
rasmus (rasmus)		
saki (saki)		
sergey (sergey)	✓	
timwolla (timwolla)	✓	
zeriyoshi (zeriyoshi)	✓	
Count:	15	

ramsey (ramsey)
rasmus (rasmus)
saki (saki)

(日本の)

PHPerは何者か

PHPのユーザーは
他の言語よりも
多様だと思う

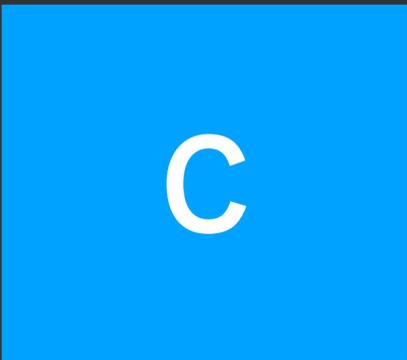
A

```
<!DOCTYPE html>
<html><head><title>Hello, PHP!</title></head>
<p>
  <?php if (date('H') < 19): ?>
    こんにちは
  <?php else: ?>
   こんばんは
  <?php endif; ?>
</p>
<p>今は<?= htmlspecialchars(date('H')) ?>時です</p>
```

B

```
<?php require 'bootstrap.php';

$db = mysqli_connect();
$stmt = mysqli_prepare($db, 'SELECT * FROM books');
mysqli_stmt_execute($stmt);
print '<h1>本の一覧</h1>';
print '<ul>';
foreach ($stmt as $s) {
    print '<li>'; print $s->name; print '</li>';
}
print '</ul>';
```

C

```
<?php declare(strict_types=1);  
  
namespace Foo\Http\Controller;  
  
class BooksController extends BaseController {  
    public function index() {  
        $books = Books::getAll();  
  
        $this->render(compact("books"));  
    }  
}
```

前提が「プログラミング」
「システム開発」
ではないユーザーも多い

 Edit



PHP Early Days (誕生から4まで)

2023/09/17に公開

 PHP  # php4  Tech

この記事は何か

某社の某雑誌に寄稿しようと思って書いた記事の初稿... だったのですが、テーマじゃないところに紙幅を割きすぎて本来のテーマを大幅に逸脱した上に、書いてから三年ほど置いてしまったものです (執筆自体はまだ諦めてません)。

ちょうど「LLイベント」ことLearn Languages 2023というイベントで「PHPの20年とこれから」としてPHPの歴史を話したので、せっかくなのでこの期に公開しましょう。

 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

 この記事は何か

 何かという話について

なお、この頃(1996～1997年)までには日本にもPHP/FIユーザーコミュニティが成立していたようで、廣川類さんの「[PHP日本語ページ](#)」が既に公開されていました。[Internet Archive Wayback Machine](#)の最古のアーカイブは1996年4月時点ですが、PHP/FIのQ&AやPHP 2.0や日本語パッチリリース情報について言及があります。

! 廣川さんは現在のmbstringモジュールの母体となったlibmbflの開発・メンテナンスに関わっていたほか、「[PHPの今とこれから](#)」として2011年から現在まで毎年PHPカンファレンスでPHPのアップデート情報を伝えています。これはユーザー視点からの期待感を含めたリアルタイムでのPHP開発状況が記録された一級の史料となっています。



Rui Hirokawa

27 SlideShares

91 Followers

65 Followings

Follow

...

Tags

- php
- php7
- php4
- php8
- phpcon2013
- phpcon2023
- php-con
- phpcon2019
- phpcon2003
- phpcon2010
- phpcon2009
- phpcon2008
- phpcon2007
- phpcon2006

See more

Presentations Documents Infographics

Latest Most Popular



PHPの今とこれから2023

PHP Conference Japan 2023
October 8, 2023
廣川 類 (日本PHPユーザ会)

PHPの今とこれから2023

9 months ago • 1790 Views



PHPの今とこれから2022

PHP Conference Japan 2022
September 24-25, 2022
廣川 類 (日本PHPユーザ会)

PHPの今とこれから2022

1 year ago • 1179 Views



PHPの今とこれから2021

PHP Conference Japan 2021
October 2, 2021 11:00-11:40
廣川 類 (日本PHPユーザ会)

Discord PHP Conference Japan 2021 DAY-1-TRACK1 #track1-k-php2021

PHPの今とこれから2021

2 years ago • 1243 Views



PHPの今とこれから2020

PHP Conference Tokyo 2020
December 12, 2020
廣川 類 (日本PHPユーザ会)

PHPの今とこれから2020

3 years ago • 2836 Views



PHPの今とこれから2019

PHP Conference Tokyo 2019
December 1, 2019
廣川 類 (日本PHPユーザ会)

PHPの今とこれから2019

4 years ago • 15192 Views



日本PHPユーザ会の紹介

PHPカンファレンス2003
2003年8月30日 日本PHPユーザー会

日本PHPユーザ会の紹介

5 years ago • 646 Views

[T-9]



「PHPの中の人によるパネルディスカッション」

"Under the Hood of PHP"
Panel Discussion by Core PHP developers

Panelist: Rasmus Lerdorf, Masaki Fujimoto, Yasuo Ohgaki, Moriyoichi Koizumi, Masahiro Takagi, Yoshio Hanawa
Moderator: Rui Hirokawa

PHP Conference Japan 2010
25 September 2010 16:50-17:35

PHPの中の人によるパネルデ...

5 years ago • 340 Views



PHPの今とこれから 2009

PHPカンファレンス2009
2009年9月5日(土)
日本PHPユーザ会 廣川 類

PHPの今とこれから2009

5 years ago • 444 Views



「PHPの今とこれから 2008」

PHPカンファレンス2008
2008年7月21日(土)
日本PHPユーザ会 廣川 類

PHPの今とこれから2008

5 years ago • 322 Views

PHP 4とPHPコミュニティの拡大

ともあれ、PHP 4では大幅なパフォーマンス改善や機能追加が達成されました。この時期の2000年には日本および世界でPHPカンファレンスが開催されるようになった時期でもあります。

phpcon PHPカンファレンス2024
@phpcon · Follow

先日開催された [#phpworld](#) にて、世界初のPHPのカンファレンスが日本のPHPカンファレンスだと正式に認定されました！ [#phpcon](#)

3:51 PM · Nov 16, 2019

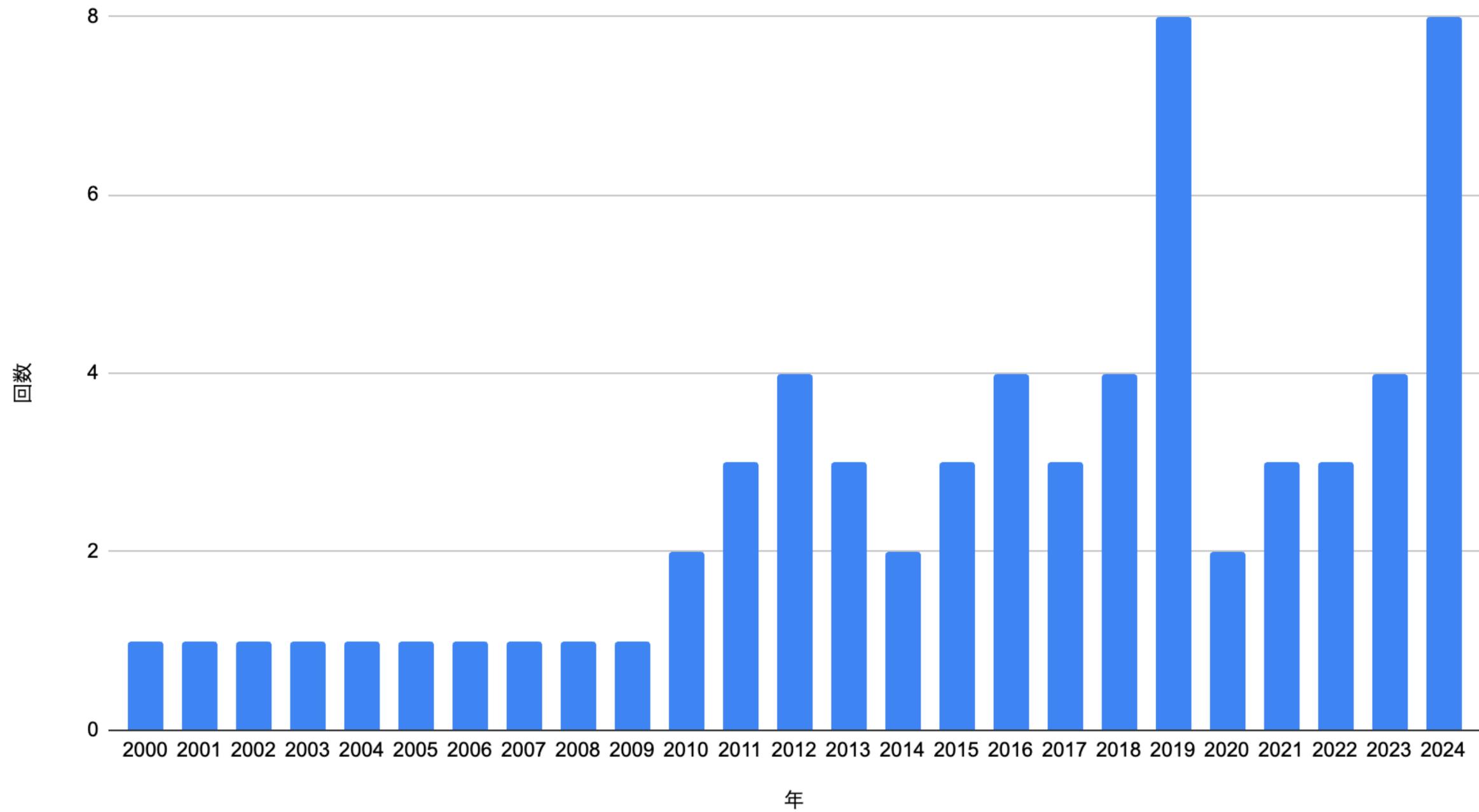
74 Reply Copy link

[Read more on X](#)

Q. 日本でPHP系の
カンファレンスは何回
開催されたのでしょうか

A. 数えかたにもよるが、
60~100回くらい
(WordCampをカウントするかで変化)

PHPカンファレンス開催回数／年



40	PHPカンファレンス仙台2019	https://phpcon-sendai.n	2019-01-26	2019	日本	東北
41	Laravel JP Conference 2019	https://web.archive.org/	2019-02-16	2019	日本	関東
42	PHPerKaigi 2019	https://phperkaigi.jp/201	2019-03-29	2019	日本	関東
43	PHPカンファレンス福岡2019	https://phpcon.fukuoka.	2019-06-29	2019	日本	九州
44	PHPカンファレンス北海道2019	https://2019.phpcon.hok	2019-09-21	2019	日本	北海道
45	PHPカンファレンス沖縄2019	https://web.archive.org/	2019-10-12	2019	日本	沖縄
46	CakeFest 2019 Tokyo	https://www.cakefest.or	2019-11-07	2019	日本	関東
47	PHPカンファレンス2019	https://phpcon.php.gr.jp	2019-12-01	2019	日本	関東

60	PHPカンファレンス北海道2024	https://phpcon.hokkaido	2024-01-12	2024	日本	北海道
61	PHPカンファレンス関西2024	https://2024.kphpug.jp/	2024-02-11	2024	日本	関西
62	PHPerKaigi 2024	https://phperkaigi.jp/202	2024-03-07	2024	日本	関東
63	PHPカンファレンス小田原2024	https://phpcon-odawara	2024-04-13	2024	日本	関東
64	PHPカンファレンス香川2024	https://phpcon.kagawa.j	2024-05-10	2024	日本	四国
65	PHPカンファレンス福岡2024	https://phpcon.fukuoka.	2024-06-22	2024	日本	九州
66	PHPカンファレンス沖縄2024	https://phpcon.okinawa.	2024-09-28	2024	日本	沖縄
67	PHPカンファレンス2024	https://phpcon.php.gr.jp	2024-12-22	2024	日本	関東

地域Ruby会議(Regional RubyKaigi)

“regional” wasn’t going to mean “provincial”—that regional conferences could be top-notch events—and that hope has been fulfilled beyond what we could possibly have wished for.

— [D.A.Black](#)

地域Ruby会議(Regional RubyKaigi;リージョなるるびーかいぎ)は、[RubyKaigi](#)のようなイベントをいろんな地域でやっってしまうというプロジェクトです。[日本Rubyの会](#)が後援しています。

地域Ruby会議

未確定情報を含む、今後の開催情報は、日本Ruby会の公式Wikiの[これから開催される地域Ruby会議](#)のページも参照してください。

Date	Name	Report			
			2019-12-14(Sat)	平成Ruby会議01	レポート
2024-10-05(Sat)	松江Ruby会議11	レポートはまだありません	2019-11-30(Sat)	鹿児島Ruby会議01	レポート
2024-09-07(Sat)	福岡Rubyist会議04	レポートはまだありません	2019-11-03(Sun)	富山Ruby会議01	レポート
			2019-09-15(Sun)	大阪Ruby会議02	レポート
			2019-07-06(Sat)	TamaRuby会議01	レポート
			2019-06-29(Sat)	TokyuRuby会議13	レポート
			2019-06-29(Sat)	とちぎRuby会議08	レポート
			2019-06-08(Sat)	名古屋Ruby会議04	レポート

Q. どうしてこうなったのか

A. コロナ禍明けで
直接会って会話する楽しさを
思い出してしまった

A. 自分も発表したり
イベントを開催したいと
思う人口の裾野が広がった

PHPカンファレンス北海道2024	https://phpcon.hokkaido.jp/	2024-01-12	2024	日本	北海道	復活 + 新人委員長
PHPカンファレンス関西2024	https://2024.kphpug.jp/	2024-02-11	2024	日本	関西	復活 + 新人委員長
PHPerKaigi 2024	https://phperkaigi.jp/2024/	2024-03-07	2024	日本	関東	継続
PHPカンファレンス小田原2024	https://phpcon-odawara.jp/	2024-04-13	2024	日本	関東	新規開催
PHPカンファレンス香川2024	https://phpcon.kagawa.jp/	2024-05-10	2024	日本	四国	新規開催
PHPカンファレンス福岡2024	https://phpcon.fukuoka.jp/	2024-06-22	2024	日本	九州	継続 + 新人委員長
PHPカンファレンス沖縄2024	https://phpcon.okinawa.jp/	2024-09-28	2024	日本	沖縄	継続
PHPカンファレンス2024	https://phpcon.php.gr.jp/	2024-12-22	2024	日本	関東	継続 + 新人委員長
WordCamp Tokyo 2022	https://www.wordcamp.org/			日本	関東	2022-09-22

YAPC
RubyKaigi
PyCon JP
etc...

言語固有のコミュニティの
事情がいろいろある

PHPerは
どこへ行くのか

よくも悪くも
コミュニティの盛り上がりで
カンファレンス発表の
ハードルが上がってしまった

コミュニケーションの場は
カンファレンスだけじゃない



18



10



▲ PHP Advent Calendar 2023 1日目

@tadsan (園島 めめ) in pixiv Inc.

PHPイベントの世界

PHP PHPカンファレンス

最終更新日 2023年12月02日 投稿日 2023年12月01日 2168 views

こんにちは！ PHP Advent Calendarです！ メリークリスマス！

ところで、みなさんはIT技術イベントに参加したことはありますか。世の中ではさまざまな趣旨・さまざまな規模のイベントが開催されています。PHP界隈においても、やはりいろいろなイベントが企画されているところです。

この記事では、このようなイベントについて紹介いたします。



PHP 📡

29752
記事

87018
フォロワー

記事をフォロー中

質問をフォロー中

ユーザーランキング

週間 月間 年間

- 👤 @mamimami0709 (mami)
23 Contributions
- 👤 @suin (suin@TypeScriptが好き)
22 Contributions
- 👤 @keitaMax
19 Contributions
- 4 🔄 @yukimaruyama (Maru)
15 Contributions
- 5 🟢 @Jim_Jin (Jimmy Jin)
14 Contributions

📖 PHPとは?

- 公式サイト: [PHP: Hypertext Preprocessor](#)
- 公式リファレンス: [PHP: Documentation](#)
- Wikipedia: [PHP \(プログラミング言語\) - Wikipedia](#)

記事 質問

👤 @hanaosan
2024年07月20日

【PHP】 PHP7.4から8.1にバージョンアップしてみた

PHP CakePHP PHPUnit バージョンアップ

♡ 0



👤 @soradore_
2024年07月20日

【PHP】 Enumでの検証

PHP

♡ 0



🏠 paizaQiita記事投稿キャンペーン 「プログラミング問題をやってみて書いたコードを投稿しよう！」

👤 @MelonPanUryuu (大塚 健一)
2024年07月19日

【PHP】 Fizz Buzz (paizaランク C 相当)をやってみた

会って触れ合う場は
多い方がいい



Kansai Dev Garage

イベント

メンバー

資料

B! 0

いいね! 0

✕ ポスト

グループのメンバーです

メール通知設定

イベント (7件)



開催前 2024/08/17 (土) 13:30~

2/15人

良いコードの書き方勉強会 in 京都

SAW 他

京都府京都市南区東九条北烏丸町24-22 MEETS201号室



開催前 2024/07/24 (水) 19:00~

3/18人

もくモグ会@奈良/オンライン同時開催

yasuyuki_higa 他

奈良県奈良市内待原町50



2024/07/19 (金) 19:00~

13/24人

PHP勉強会 in 神戸 (PHP"オレ"カンファレンス 非公式前夜祭)

SAW 他

兵庫県神戸市中央区東町115番地



2024/06/19 (水) 19:30~

11/14人

コンテナ・インフラ勉強会 in 大阪

SAW 他

大阪府大阪市北区大深町3-1 グランフロント大阪 タワーC 16階



2024/05/25 (土) 13:30~

8/15人

Laravel勉強会 in 奈良

SAW 他

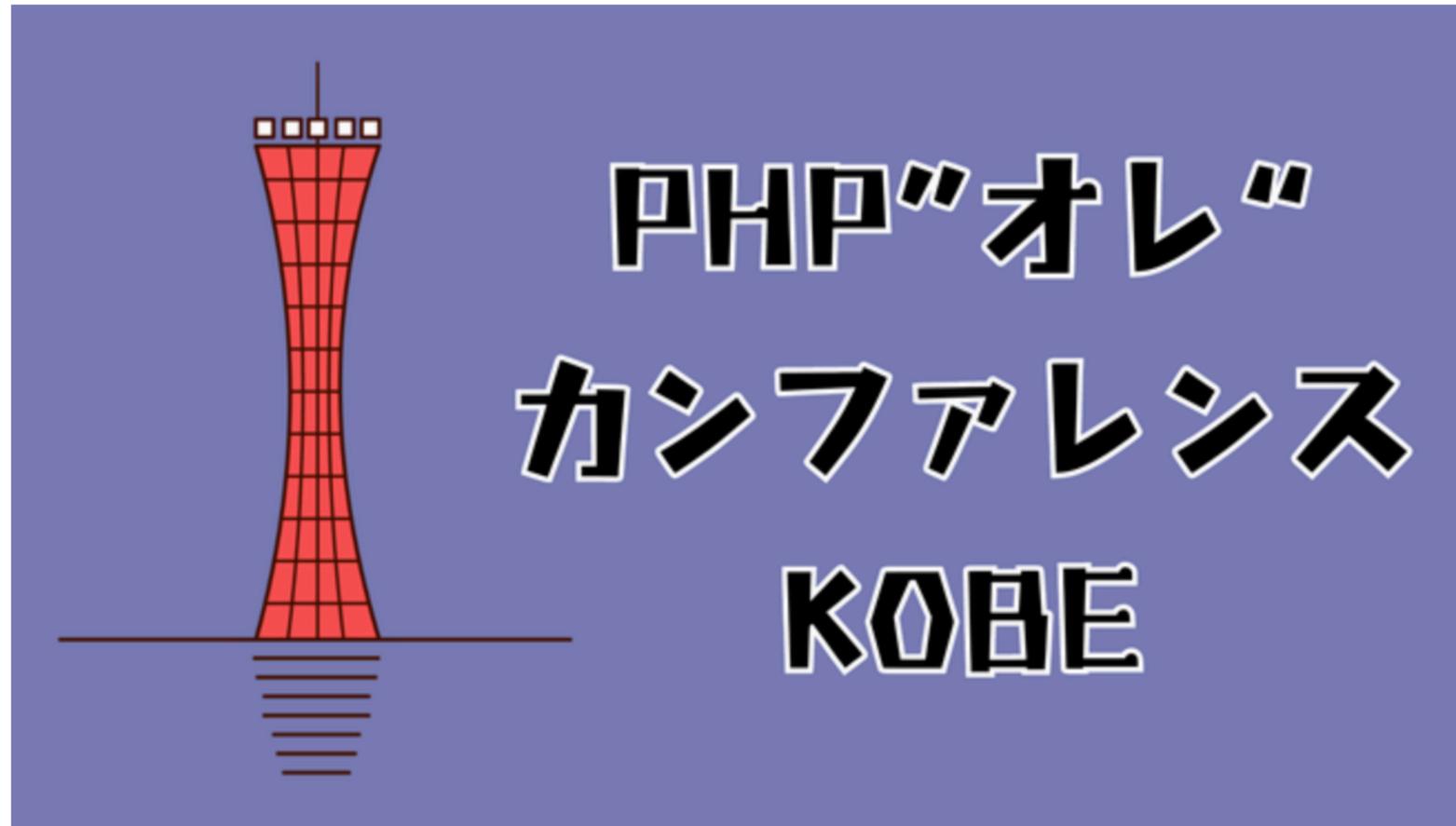
奈良県奈良市杉ヶ町23番地



7月
20

PHP"オレ"カンファレンス神戸 #1

オレたちの、オレたちによる、オレたちのためのぺちこん



ハッシュタグ : #orekobe

登壇枠(15分)

無料

先着順 (抽選終了)

6/6人

LT枠(5分)

先着順 (抽選終了)

グループ

メンバーになる

blessing software

神戸を中心に関西でtech LT会を開催しています



イベント数 **19回**

メンバー数 **122人**

開催中

2024/07/20(土)

09:00 ~ 19:00

Googleカレンダー icsファイル

募集期間が終了しました

開催日時が重複しているイベントに申し込んでいる場合、このイベントには申し込むことができません

募集期間

2024/04/08(月) 17:31 ~

2024/07/20(土) 09:00

イベントへのお問い合わせ

情報を他者に教えられる
レベルに整理することで
自分の理解を深めたい

個人的な思想：
直接的な勉強より
多角的な視点を得たい

ABOUT

PHPPerKaigi (ペチパーカイギ) とは？

PHPPerKaigi (ペチパーカイギ) は、PHPer、つまり、現在PHPを使用している方、過去にPHPを使用していた方、これからPHPを使いたいと思っている方、そしてPHPが大好きな方たちが、**技術的なノウハウとPHP愛を共有するためのイベント**です。

今年の開催はオフライン会場を軸としたオフライン・オンラインハイブリッド開催です。みなさんのご都合に合う参加方法をお選び頂けます。

2024年3月7日 (木) ~ 3月9日 (土) はPHPerたちのお祭りです！ 歴戦の勇者みたいな方ももちろん、初心者の方にも、全てのPHPerが楽しめるイベントです。スケジュール、空けておいて下さい！

開催に関連するニュースは公式X (旧Twitter) で告知をします。是非フォローしておいてください。

PHPPerKaigi公式X (旧Twitter) : [@phperkaigi](#)

PHP & 技術
ではなく
PHP愛 || 技術

発表内容を「勉強」として
完全理解することには
こだわらる必要はない

記憶のフックを作って
いつか役立ててほしい

PHPにお世話になっている
からといって全員が
PHPエコシステムに貢献し
ないといけないことはない

技術コミュニティを
楽しみましょう