Analyzing PHP

pixiv Inc.
USAMI Kenta

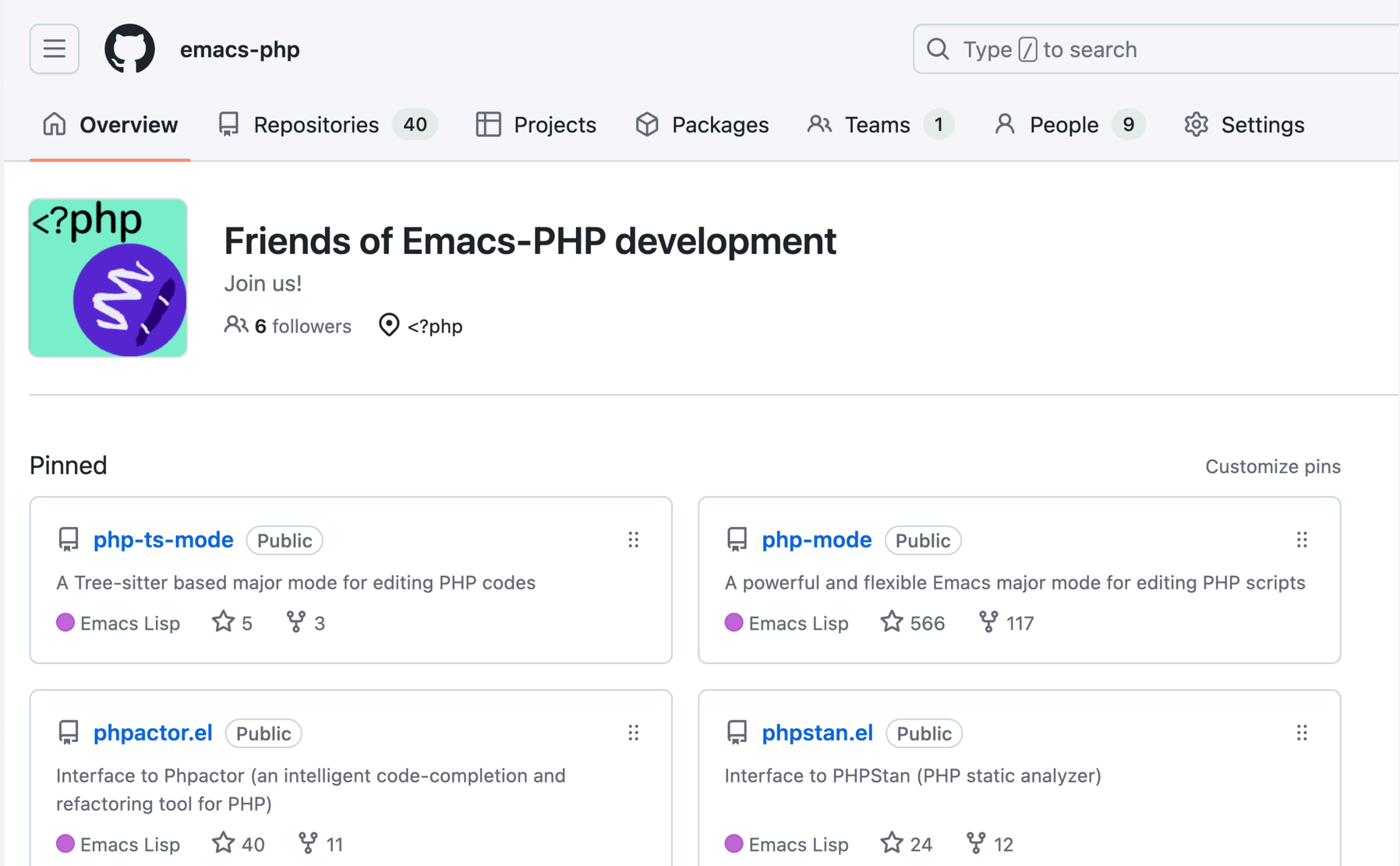
pixiv

2023-08-30
TechFeed Expert Night

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）

emacs-php



pixiv

PHPカンファレンス沖縄2023

pixiv

LLイベント

Learn Languages

Learn Languages 2023

About

Learn Languagesは様々なプログラミング言語について学びたい技術者のためのイベントです。2003年からほぼ毎年開催しています。

(Open Developers Conference (ODC) 2023 のトラックとして開催します)

URL	https://ll.jus.or.jp/2023/
ハッシュタグ	#ll2023jp

Session

Perl, PHP, Python, Rubyの20年とこれから

2003年に本イベントの第1回で扱ったプログラミング言語はPerl, PHP, Python, Rubyの4つでした。

今回は再びこれら4つの言語を取り上げ、今までの20年を振り返りこれからの展望を語っていただきたいと思います。

タイムテーブル:

13:00 - 13:20 Perl
13:25 - 13:45 PHP
14:00 - 14:20 Python
14:25 - 14:45 Ruby
15:00 - 15:45 この20年を振り返る座談会

History

このイベントは2003年に*LL Saturday*として開始しました。LLはLightweight Languageの略で、*LL Saturday*ではスクリプト言語であるPerl, PHP, Python, Rubyについて取り上げました。

2017年には様々な言語を学ぼうという主旨で、**Learn Languages**という名称にリニューアルしました。2020年は新型コロナウイルス感染拡大のため開催を断念しました。

今までのアーカイブは以下を参照してください。

Learn Languages 2022
Learn Languages 2021
Learn Languages 2019

pixiv

PHPって
どんな言語？

みなさんの想像する PHPとは

脆弱性 ゆるふわ 弱い型 動的
クソザコ 型なし 意味不明 弱い
自動変換 貧弱 Perlっぽい 適当

PHPについてのパブリックイメージ

~~脆弱性~~ ~~ゆるふわ~~ ~~弱い型~~ 動的

クソザコ ~~型なし~~ 意味不明 ~~弱い~~

自動変換 貧弱 ~~Perlっぽい~~ ~~適当~~

PHPについての認識は概ね間違い

ソースコードで見ると

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>
```

pixiv

B

```php
<?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>';
```

pixiv

C

```php
<?php declare(strict_types=1);

namespace Foo\Http\Controller;

class BooksController extends BaseController {
    public function index() {
        $books = Books::getAll();

        $this->render(compact("books"));
    }
}
```

pixiv

ＰＨＰの原作者

ラスマスかく語りき

2007年 MySQL Conference

“We have things like protected properties. We have abstract methods. We have all this stuff that your computer science teacher told you you should be using. I don't care about this crap at all.”

–Rasmus Lerdorf

https://en.wikiquote.org/wiki/Rasmus_Lerdorf

2007年 MySQL Conference

“PHPにはprotectedプロパティも
抽象メソッドもありますよ。計算機科学の
教授が「使え」と言ってるものは全部。
そんなことは~~クソ~~興味ないですけど”

–Rasmus Lerdorf (tadsanによる超訳)

変数に$があって
Perlっぽくて

C言語っぽい
関数と
-> があって

雰囲気Javaっぽい
オブジェクト指向で

PHPは見る人の心を映す

pixiv

整理しましょう

PHPは汎用プログラミング言語

- PHPはHTTP(≒Webサービス)を言語レベルでサポートしている
  - ほかの言語ではCGIライブラリとしてサポートしている機能が組み込み
- 各Webサーバとの連携は**SAPI**(Server API)として定義されている
  - どのサーバー上でもCGI風の動作をする
  - コマンドラインスクリプトとして実行するモード(CLI)もSAPIのひとつ

Perl

```
use DBI;

$db = DBI->connect('...');
$stmt = $db->prepare('SELECT * FROM books');
$stmt->execute;
print '<h1>本の一覧</h1>'; print '<ul>';
for ($i = 0; $I < $sth->rows; $i++) {
    @row = $sth->fetchrow_array;
    print '<li>'; print $row[0]; print '</li>';
}
print '</ul>';
```

pixiv

B = CGIスタイル

```
<?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>';
```

pixiv

現在でも動くCGI

言語としてのPHP

- よくあるC言語風の制御構文と標準関数をもったスクリプト言語
  - if, else, switch, while, for, **goto**
    - PHPのgotoは綺麗なgoto
- 関数定義、クラス定義(class, interface, trait, enum)
- クラスや関数の再定義(オープンクラス・モンキーパッチ)はできない

PHPの実行フロー

- いわゆるVM(仮想機械)型のインタプリタ
  - RubyをはじめPerl, Pythonなど現代的な言語は基本的にこれ
- 字句解析→構文解析→オペコードコンパイル

Zend Engine

文A 17の言語版

ページ　ノート　　閲覧　ソースを編集　履歴表示　★　その他

出典: フリー百科事典『ウィキペディア（Wikipedia）』

Zend EngineはPHPのインタプリタ（レジスタマシンベースの仮想機械）である。PHP Licenseに類似したZend Engine Licenseが適用される自由なソフトウェアとして、The PHP Groupによって開発・公開されている。

概要 [ソースを編集]

PHPはもともとラスマス・ラードフによって開発されたソフトウェアであったが、PHP 2(PHP/FI)までほぼ1人で開発を行っていた[1]。PHP 3を作るに当たり、イスラエル工科大学の学生であったアンディ・ガトマンズ（Andi Gutmans）とゼーブ・スラスキー（Zeev Surask）が構文解析部の開発へ加わった[1]。PHP 4では、この2人によってパーサが完全に作り直され、2人の名前から**Zend Engine**と名付けられた[2]。その後、二人はゼンド・テクノロジーズを創業し、PHPをベースとしたWebアプリケーションの開発を行っている。また、PHP自体もZend Engineがリリースされた1999年以降、急速に活躍の場を広げていった[2]。

機能 [ソースを編集]

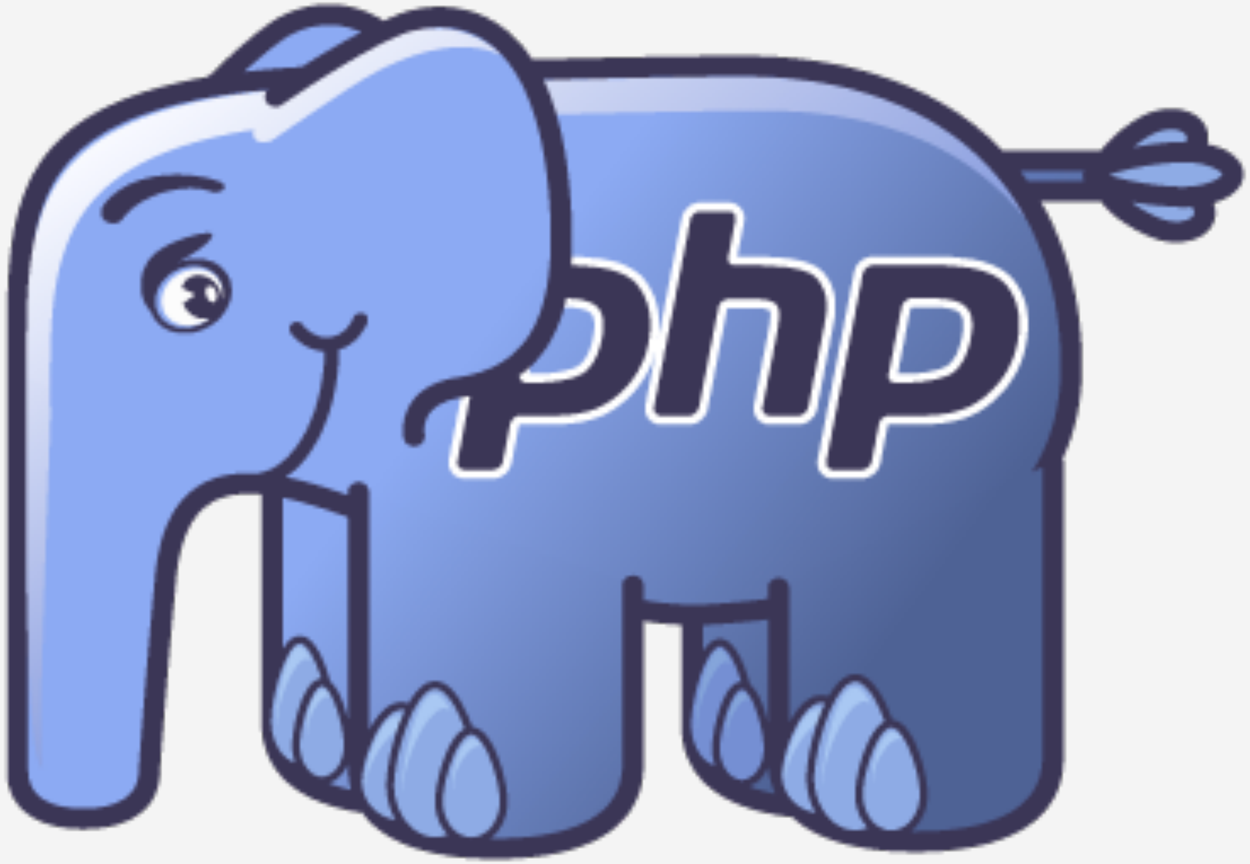
PHP 3以前では、実行のたびにソースコードを翻訳し実行を行っていた[3]。一方、Zend Engineでは、ファイル単位で中間表現へと翻訳を行い、それを実行することで、PHP 4/PHP 3の比で10倍以上という、大幅な速度向上を実現した[4]。また、Zend Engineではモジュール化が行われた[3]ほか、APIが公開されており[5]、第三者がPHP向けの拡張モジュールを開発することも可能である。

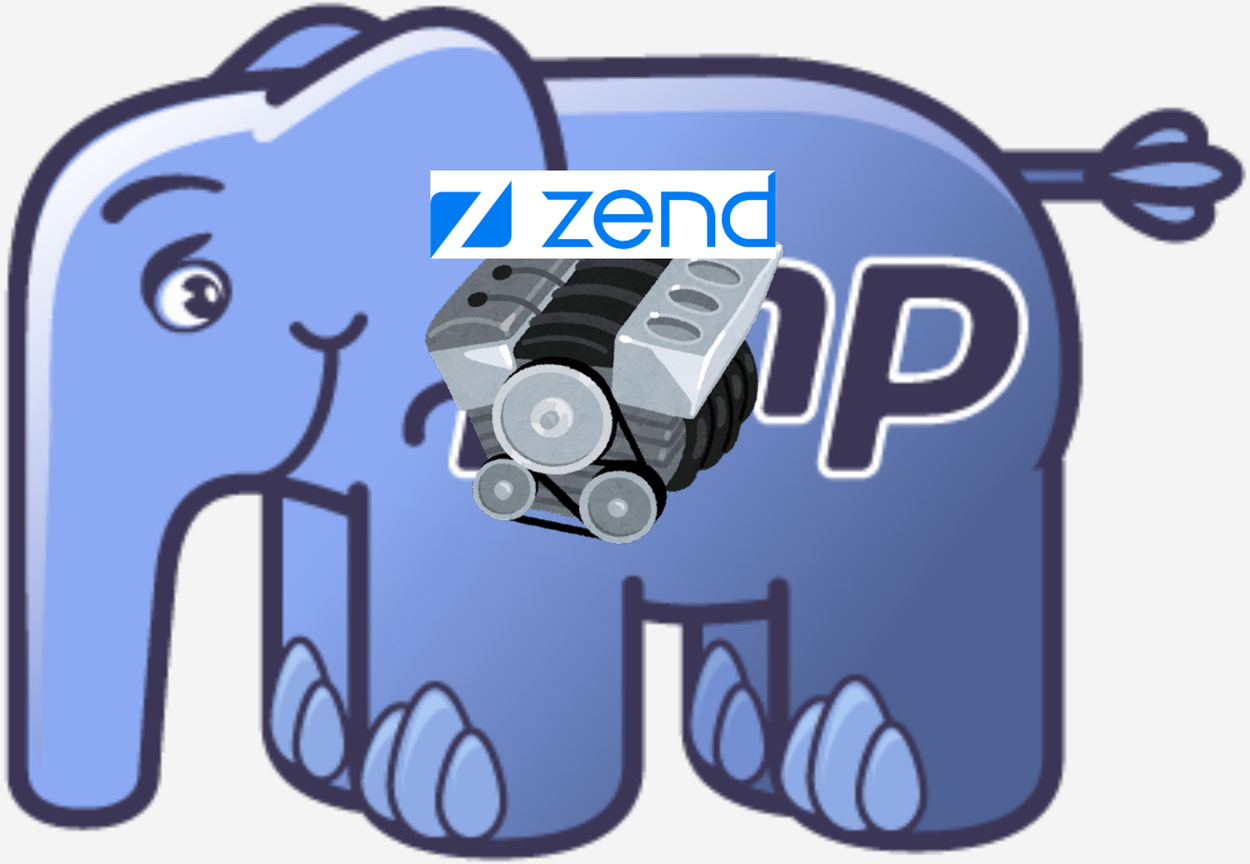
Zend Engine

作者	アンディ・ガトマンズ、ゼーブ・スラスキー
開発元	ゼンド・テクノロジーズ
初版	1999年7月19日 (23年前)
最新版	4.2.0 / 2022年12月8日 (9日前)
最新評価版	4.3.0-dev
リポジトリ	github.com/php/php-src
プログラミング言語	C言語
サポート状況	開発中
種別	インタプリタ
ライセンス	Zend Engine License
公式サイト	www.zend.com/products/zend_engine

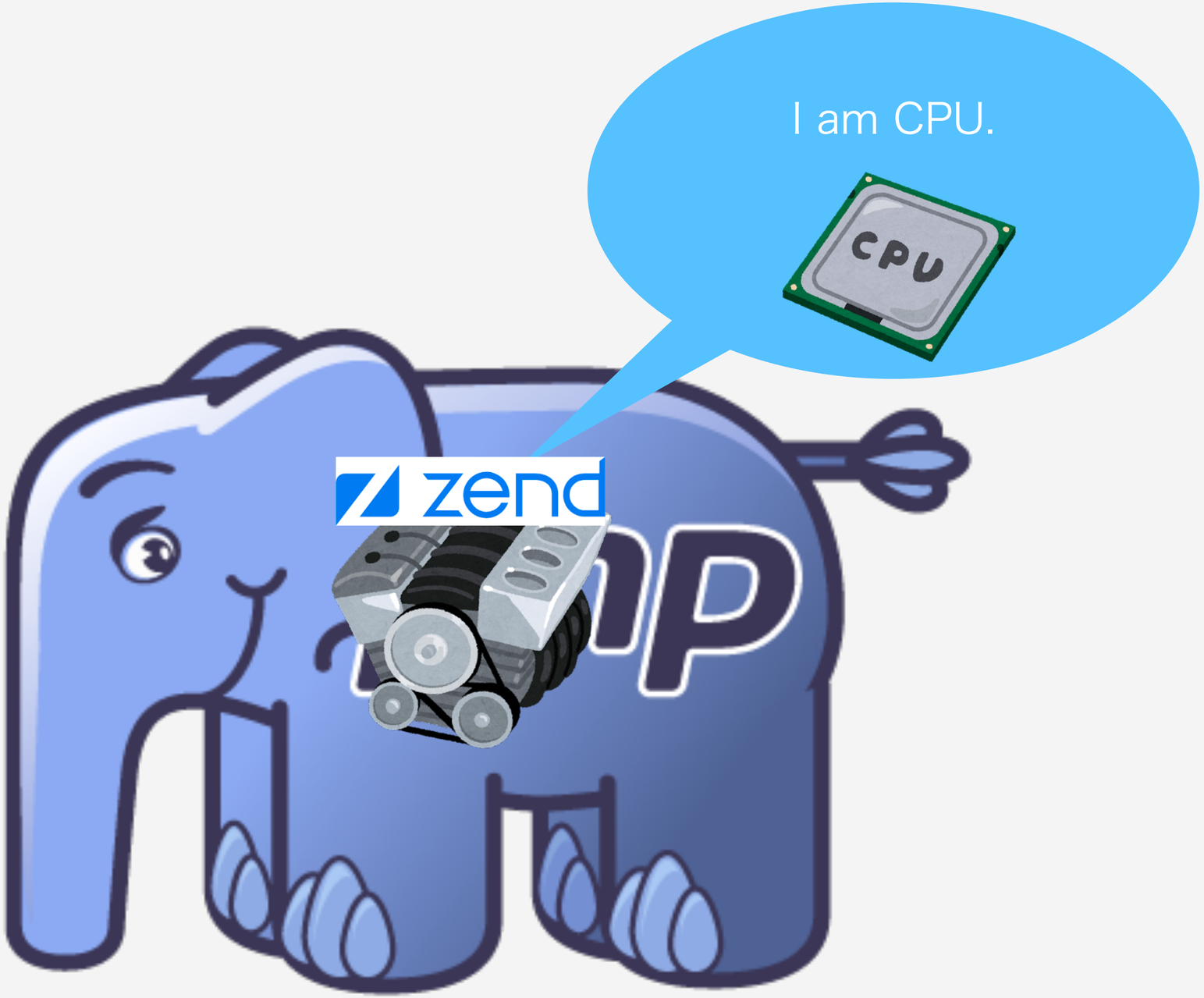
テンプレートを表示

仮想機械？


php



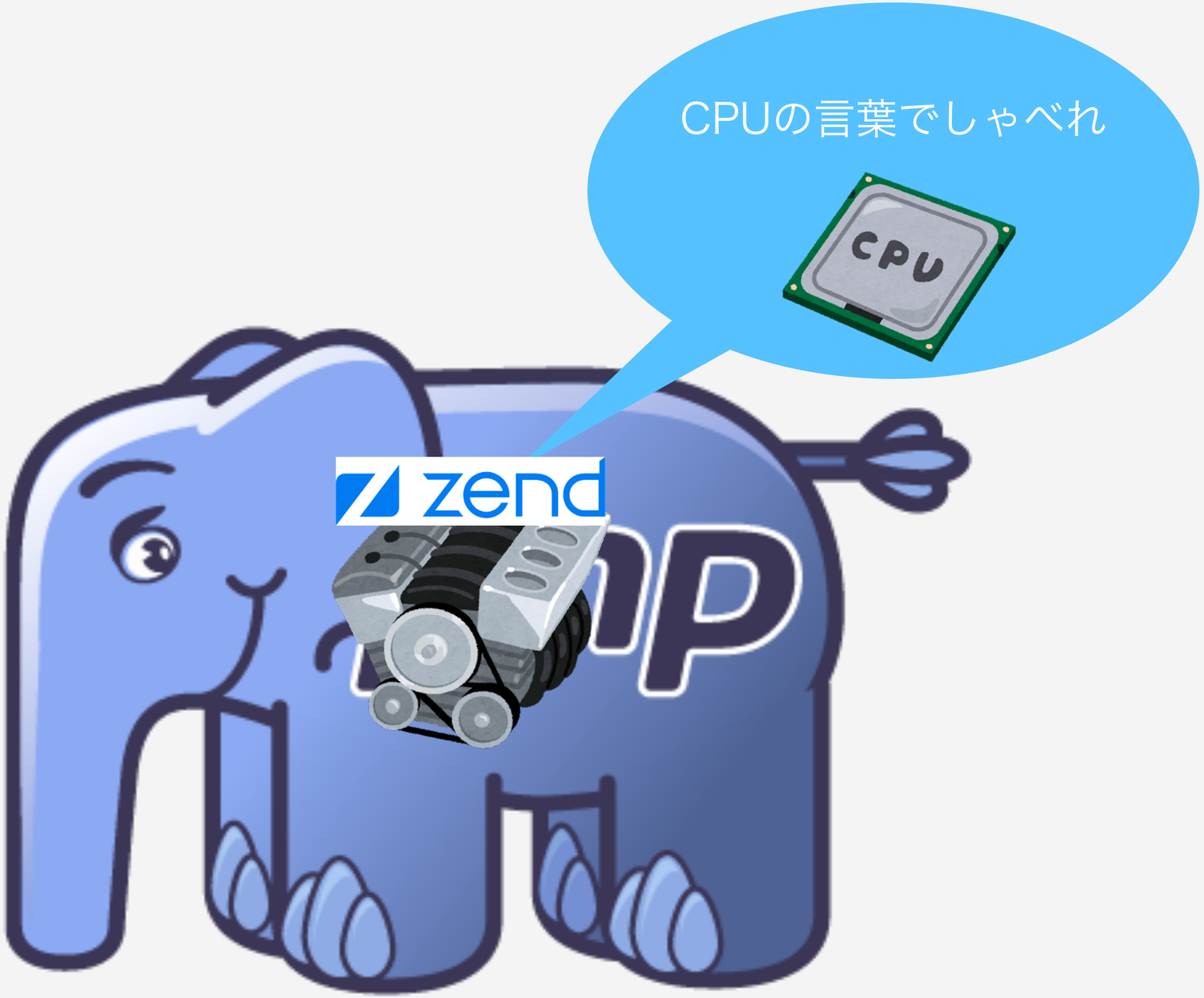
pixiv



pixiv

つまり？

自分のことをCPU だと思い込んだ象



pixiv

Zend Engine

オペコード

opcode

PHPはオペコードにコンパイルされる

```
<?php

echo 1;
```

```
ECHO                                    1
```


pixiv

PHPはオペコードにコンパイルされる

```
<?php

echo 'foo';
```

```
ECHO                    'foo'
```


pixiv

PHPはオペコードにコンパイルされる

```
<?php

print 'foo';
```

```
ECHO                    'foo'
```


pixiv

PHPはオペコードにコンパイルされる

```
<?php

echo 1 + 2;
```

```
ECHO                    3
```


pixiv

PHPはオペコードにコンパイルされる

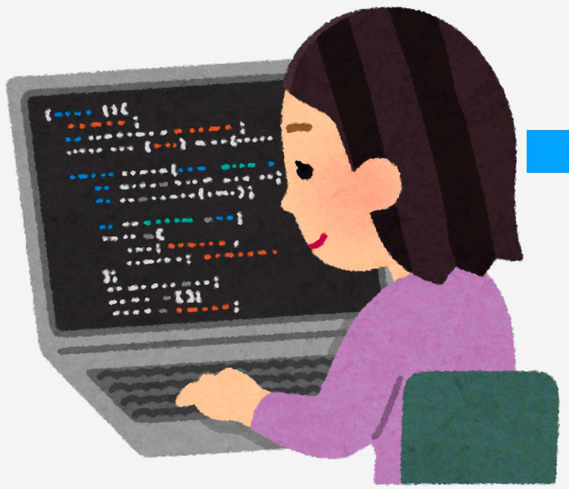
```
<?php
$a = 1;
echo $a + 2;
```

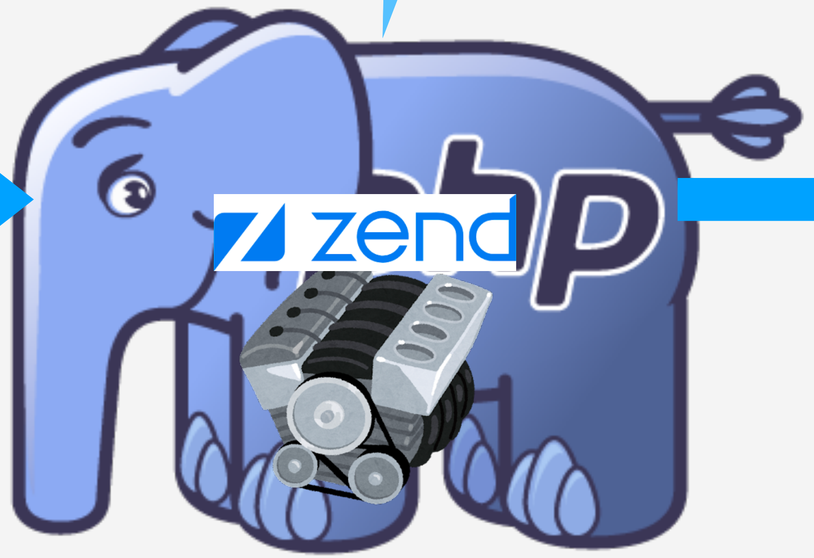
```
ASSIGN                  !0, 1
ADD              ~2     !0, 2
ECHO                        ~2
```


pixiv

<?php echo "foo";

ECHO 'foo'





pixiv

なんで
こんなことするの

PHP
高度な構文解析が必要な高級なシンタックス

opcode
構文解析不要な
シンプルな命令セット

分けて開発することで最適化しやすくなる

PHPと型宣言

- いわゆる動的型付け(dynamic typing)のランタイム
- 関数やプロパティ定義構文に型宣言があり、構文上で宣言されたものは実行時に確実にその型であることが保障される
  - ただし array などのコンテナは中に何が入っているか明示できない
- 現状では変数には型がない（なんでも入る）

PHPとDocComment型注釈

- クラスや関数定義の真上のコメント内 /** ... */に型について注釈を書く
- もちろんコメントなのでPHP実行中には解釈されないが、
  PhpStormのようなIDEや静的解析ツールなどが参照してくれる
- **array<Book>** や **array{name: string}** のようにランタイムでは
  保障されない型が書ける

静的解析しやすい特徴

- クラスや関数が再定義されることは基本的にない
- Java風のクラス継承（class, interface）
- 関数呼び出しは全て（）がつく、文末は必ず；

PHPの進化は型宣言の進化

いまやPHPは静的型付きと言っても過言ではない

（本当か…？）

型がついていない関数(PHP5)

```
function add($a, $b) {
    return $a + $b;
}
```


pixiv

スカラー型宣言(PHP7)

```php
function add(int $a, int $b): int {
    return $a + $b;
}
```

int + intって
本当にintなの？

pixiv

広い値をとるにはfloatが必要

```
function add(float $a, float $b): float {
    return $a + $b;
}
```

ひとつの解決策では
あるが… 不必要にfloat
を強制するのか

pixiv

PHPDocの型注釈(アノテーション)

```php
/**
 * @param int|float $a
 * @param int|float $b
 * @return int|float
 */
function add($a, $b) {
    return $a + $b;
}
```

pixiv

ユニオン型宣言 (PHP8.0)

```php
function add(int|float $a, int|float $b): int|float {
    return $a + $b;
}
```


pixiv

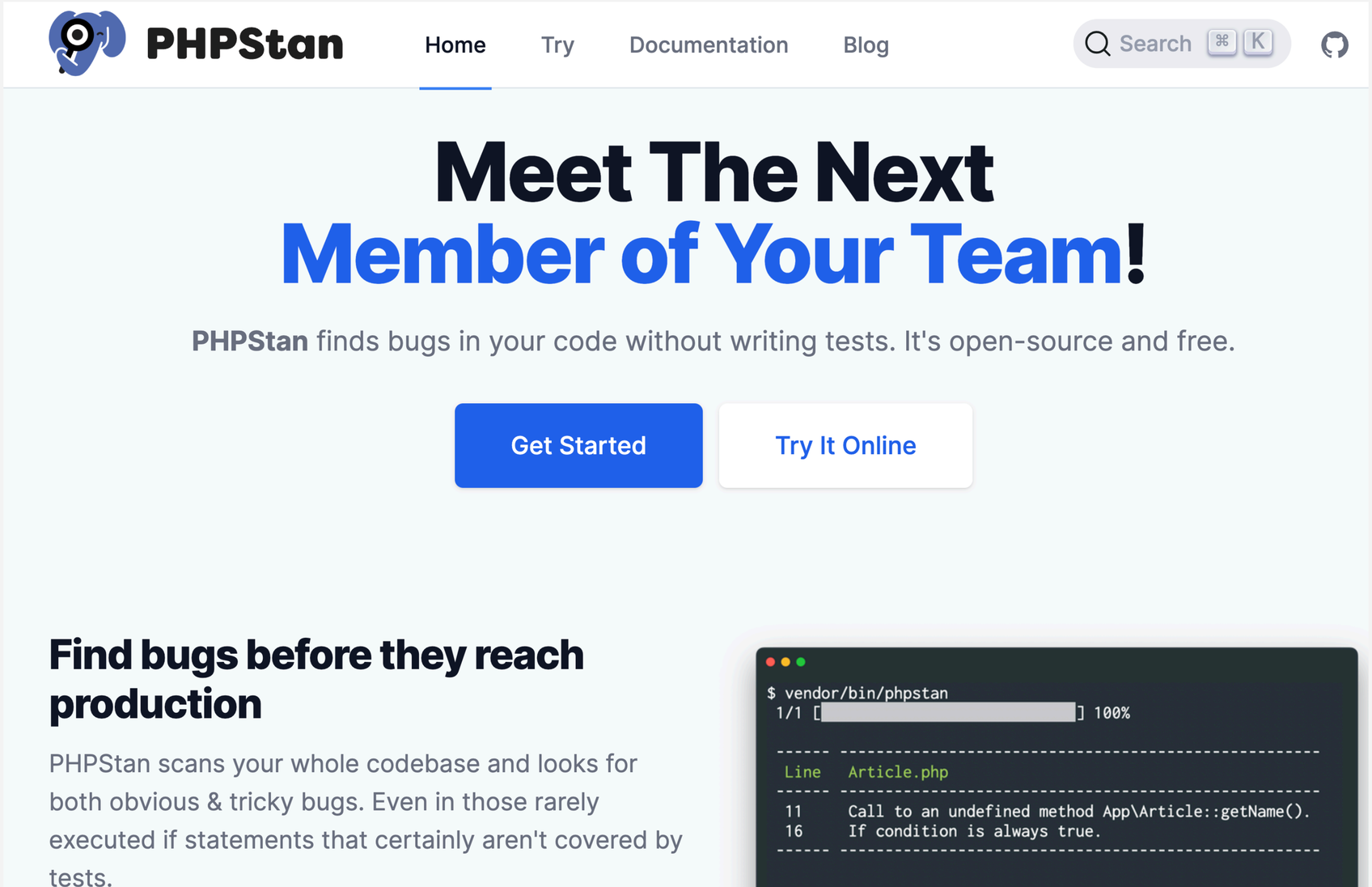
LL言語としてのPHP vs エディタ

- 2000年代のCGIやLL言語（Lightweight Language）という用語がよく使われていた頃は「重厚鈍重なIDEが必要な言語」のアンチテーゼとしてシンプルなスクリプト言語とエディタが好まれていた（ように思う）
- 時は流れ、Webアプリケーションは巨大になり、ファイルは増え、シンプルなエディタでは開発のオーバーヘッドが無視しがたくなってきた
- PhpStorm（JetBrains社のIDE）が静的解析とインテリジェントな編集機能を提供してくれるようになった

PHPと静的解析

- PHP組み込みで提供されているのは php -l（syntax check）のみ
- 「静的解析ツール」と呼ばれうるものは2000年代からあった
  - ドキュメント生成、コーディングスタイルのチェック
- 2010年代後半になってPhpStormの普及を呼び水にして、
  型システムや変数スコープを正確に扱えるツールが開発され始めた

PHP Static Analysis Tool



pixiv

PHPStanとは

- 2016年から開発されているPHPの静的解析ツール
  - Ondřej Mirtesさんの個人プロジェクト、2021年からフルタイム開発
- 開発当初は純粋な静的解析ではなく実行時リフレクションを用いることで高速な解析を実現していた
  - 現在は静的解析がデフォルトで、レガシープロジェクトに導入しやすくなった
- その他のPHP静的解析ツールにはPsalm, Phan, Qodana（PhpStorm）

PHPStanの提供する型

- PHPの組み込み型、クラス名、インターフェイス名
- 修飾型: non-empty-string, non-empty-array
- ユニオン型: A|B
- ジェネリクス: array<T>, ArrayObject<T>
- 定数型: 123, ”foo”, Foo::BAR

pixiv

PHP処理系

- ソースコード: https://github.com/php/php-src
- 処理系のソースコードはC言語で書かれている(一部のみC++)
  - パーサーはYacc(Bison)で記述されている
  - 基礎知識は『Rubyソースコード完全解説』の知識が役に立つ
- 議論はメーリングリストがメインだが、GitHub issueやPull Requestも受け付けてくれる

PHPの開発体制

- Request for Comments https://wiki.php.net/rfc
  - 言語機能の改廃提案と投票プロセス、リリース周期が明文化
- 機能提案する人は仕様をまとめてMLで議論、実装も責任を持って用意する
- PHP原作者のRasmus Lardorfは開発者としての1票しか行使しない
- PHP 7.0（2015年）以降、11月末〜12月初頭にリリースが定着
  - 品質保障のためのfeature freeze、RC版なども定められた

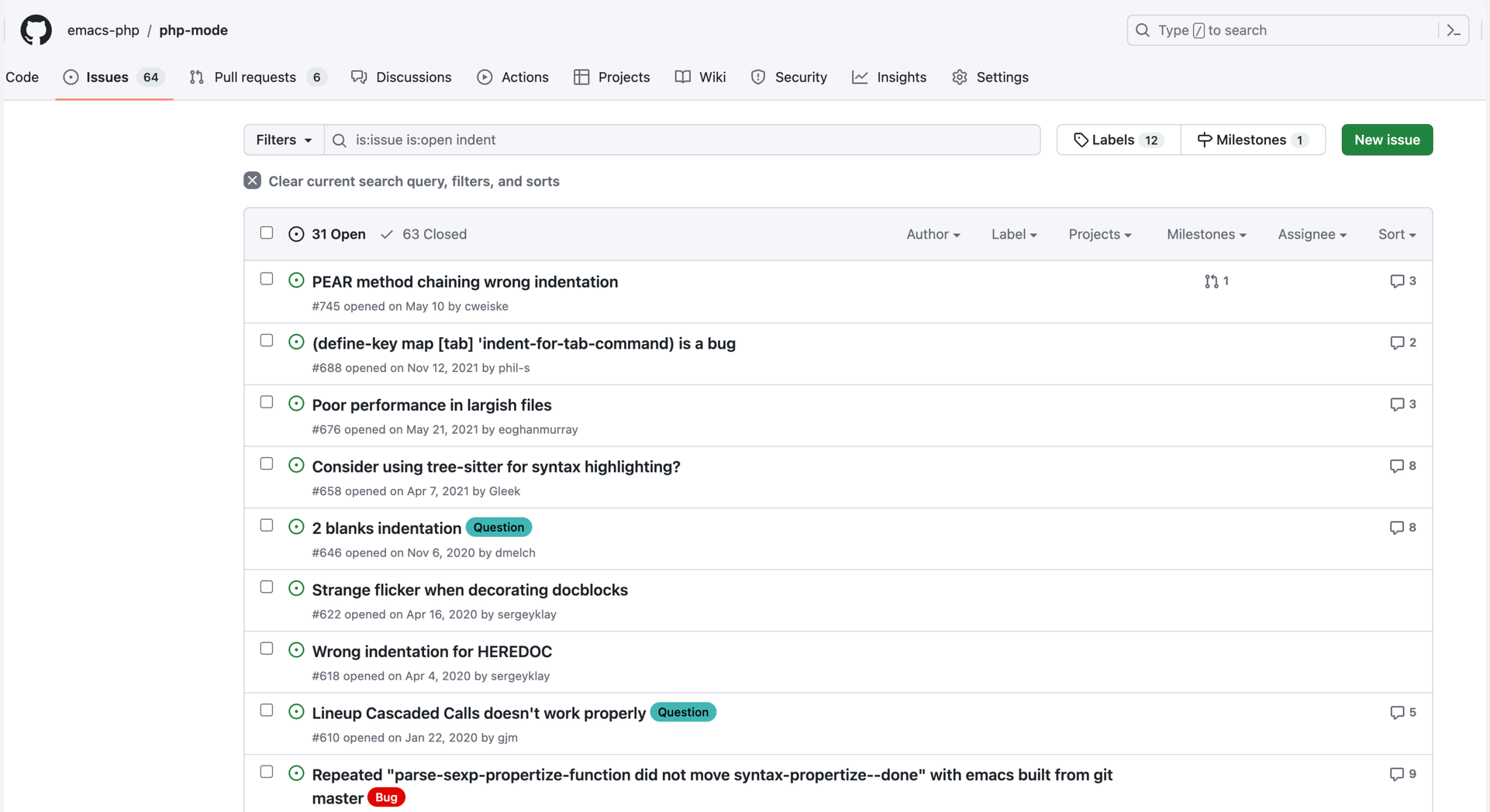
EmacsとPHP

- 実はEmacsは標準でPHP編集機能を持っていない
- サードパーティパッケージとしてPHP Modeを開発している
  - Emacsが提供するCc Modeという機能を基盤にしている
  - 「Cっぽい言語」をサポートするための巨大なフレームワークのようなもの
- ほかの言語にある機能はキーワードを設定するだけ
  - ない機能は… 地道にサポートするしかない

Emacs PHP Modeのサポート範囲

- HTMLテンプレートのようなPHPコードはサポートしていない
  - 表示できない、まともにインデントできない
    - C言語族用のフレームワークがベースになっているため
  - 別のHTMLテンプレート用のモードを使うことを推奨

php-modeの未解決issue



pixiv

最近のEmacs

- Emacs 29でLSPクライアントとTree-sitterが統合されるようになった
  - 既にいくつかの言語についてTree-sitterベースの編集モードが同梱
- Tree-sitterはエラー耐性のあるincremental parsing library
  - 言語ごとのtreesitter-xxxのようなライブラリをインストールする必要

php-ts-mode

- Emacs本体に同梱をされることを目標に開発中のパッケージ
- tree-sitter-phpが提供してくれる構文木をベースに色付け・インデント
  - ライブラリをロードしたら簡単に提供できる… かと思ったが、
    それほど簡単ではなかった
- 既存のphp-modeとの相互運用や基本的な問題を解決して、
  9月後半にリリースすることを目標に準備しています