

『えこひいき』 される関数たち

Functions Specialized by Zend Engine



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 エンジニア
 - 最近はピクシブ百科事典(dic.pixiv.net)を開発しています
- Emacs Lisper, PHPer
 - Emacs PHP Modeを開発しています(2017年-)

さて

PHP Advent Calendar 2022

参加者 20人 購読者 87人 395

購読中

カレンダー1

日	月	火	水	木	金	土
27	28	29	30	1 @tadsan Attributesで実現するPHP8時代のバリデータ 編集する	2 @noritakalzumi PHP で Postgres の boolean 型を安全に扱う	3 @naopusyu Composerコマンドを使ってパッケージの脆弱性をチェックする
4 @tadsan ZendEngineにえこひいきされる標準関数たち(前篇) 編集する	5 @tadsan えこひいきされた関数について 編集する	6 @77web@github レガシーシステムに書けるテストについて	7 +	8 @rana_kualu 【PHP8.2】PHP8.2がリリースされたので新機能全部やる	9 @ockeghem 徳丸本2版の実習環境を Docker に移植した話	10 @I-AM-RAILWAY... 代わりに投稿
11 @yoshikyoto 何か書きます 代わりに投稿	12 @rana_kualu 【PHP Next】PHPにcli amp関数がほしい	13 @takaram PHP8.2で非推奨になったエンコーディングたち	14 @hanhan1978 PHP と IEEE 754 の話	15 @tadsan 名前空間をさっくり理解する 編集する	16 @tzmfreedom symfony/pantherによるJavaScriptを含めた自動テスト	17 @tanakahisateru ちようぜつ設計とは
18 @il-m-yamagishi Laravel のリクエストだけでなく、レスポンスもバリデートしよう	19 @ippey_s GitHub CodespacesでリモートPHP開発環境を整えてPhpStormで使う3ステップ	20 @I-AM-RAILWAY... 代わりに投稿	21 @I-AM-RAILWAY...	22 @I-AM-RAILWAY...	23 @I-AM-RAILWAY...	24 @mako5656 PHPUnitについて何か書きます！
25 @mumumuorg [翻訳記事] PHP のリリースマネージャーについて	26	27	28	29	30	31



16



3



...

PHP Advent Calendar 2022 4日目



@tadsan

投稿日 2022年12月18日 3573 views

ZendEngineにえこひいきされる標準関数たち (前篇)



PHP, ZendEngine



こんにちは！ えこひいきしてますか？ 僕はしないようにしてます！

ところでPHPとかいう言語の標準ランタイム (つまりは、みなさんが普段使ってる `php` コマンド) はPHP スクリプトを直接実行するのではなく、実行前にコンパイラがPHPスクリプトをオペコードと呼ばれる中間表現(バイトコード)にコンパイルしてから、ZendEngineと呼ばれるVM(仮想機械)がオペコードを実行する構成になっていますよね。

PHPスクリプトで関数呼び出しをすると、あたりまえですが関数呼び出しをする命令にコンパイルされます。これはまったく不自然ではないことです。ところが、PHPの標準関数にはその前提に従わない、つまりは^{えこひいき}特別扱いされているものがあります。

今回の記事はそんなえこひいきされた関数たちを眺めていきましょう ム(〃><)ノ 🎉

Zend Engineって
なあに

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)まではほぼ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 [\[ソースを編集\]](#)

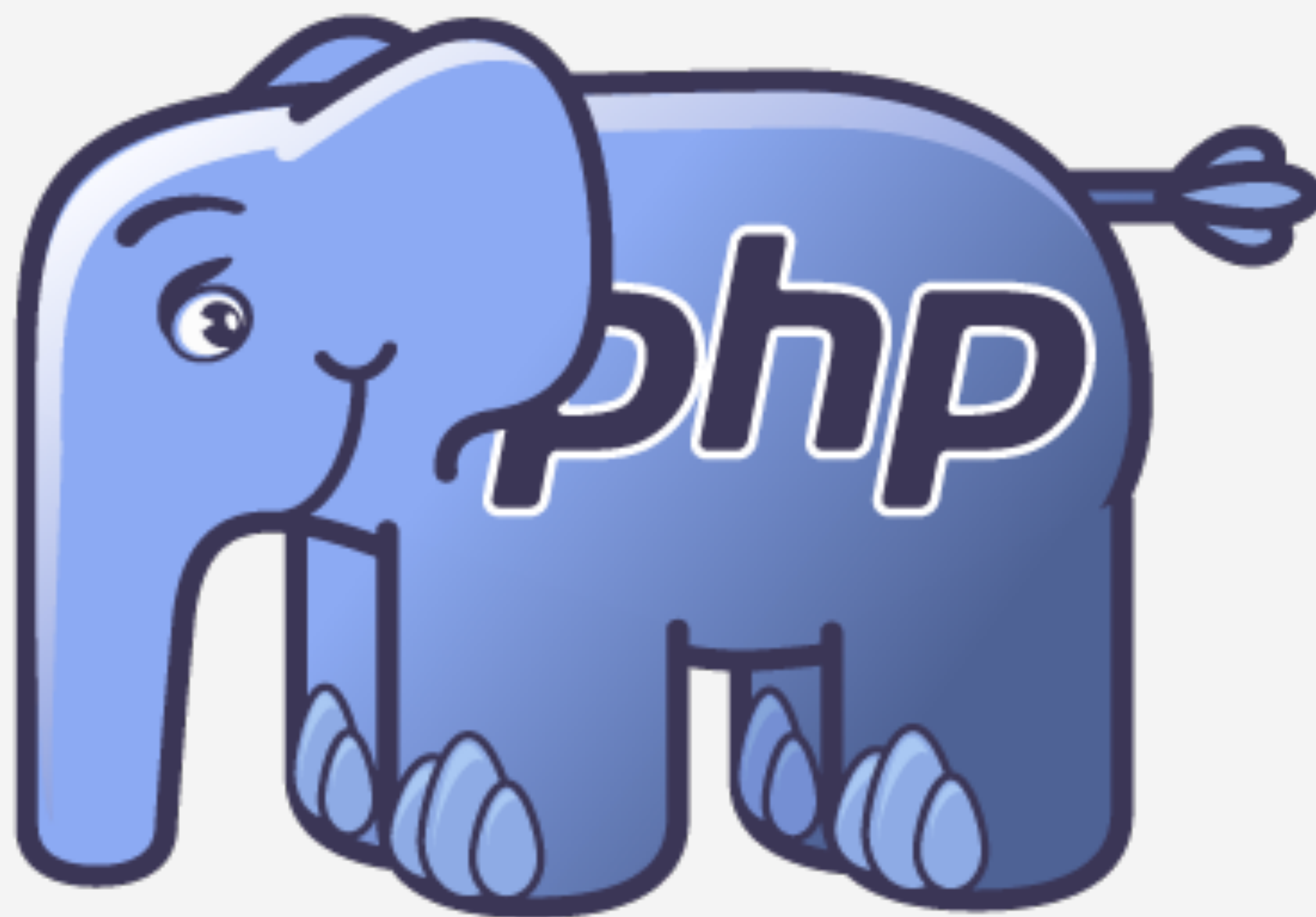
PHP 3以前では、実行のたびにソースコードを翻訳し実行を行っていた^[3]。一方、Zend Engineでは、ファイル単位で中間表現へと翻訳を行い、それを実行することで、PHP 4/PHP 3の比で10倍以上という、大幅な速度向上を実現した^[4]。また、Zend Engineではモジュール化が行われた^[3]ほか、APIが公開されており^[5]、第三者がPHP向けの拡張モジュールを開発することも可能である。

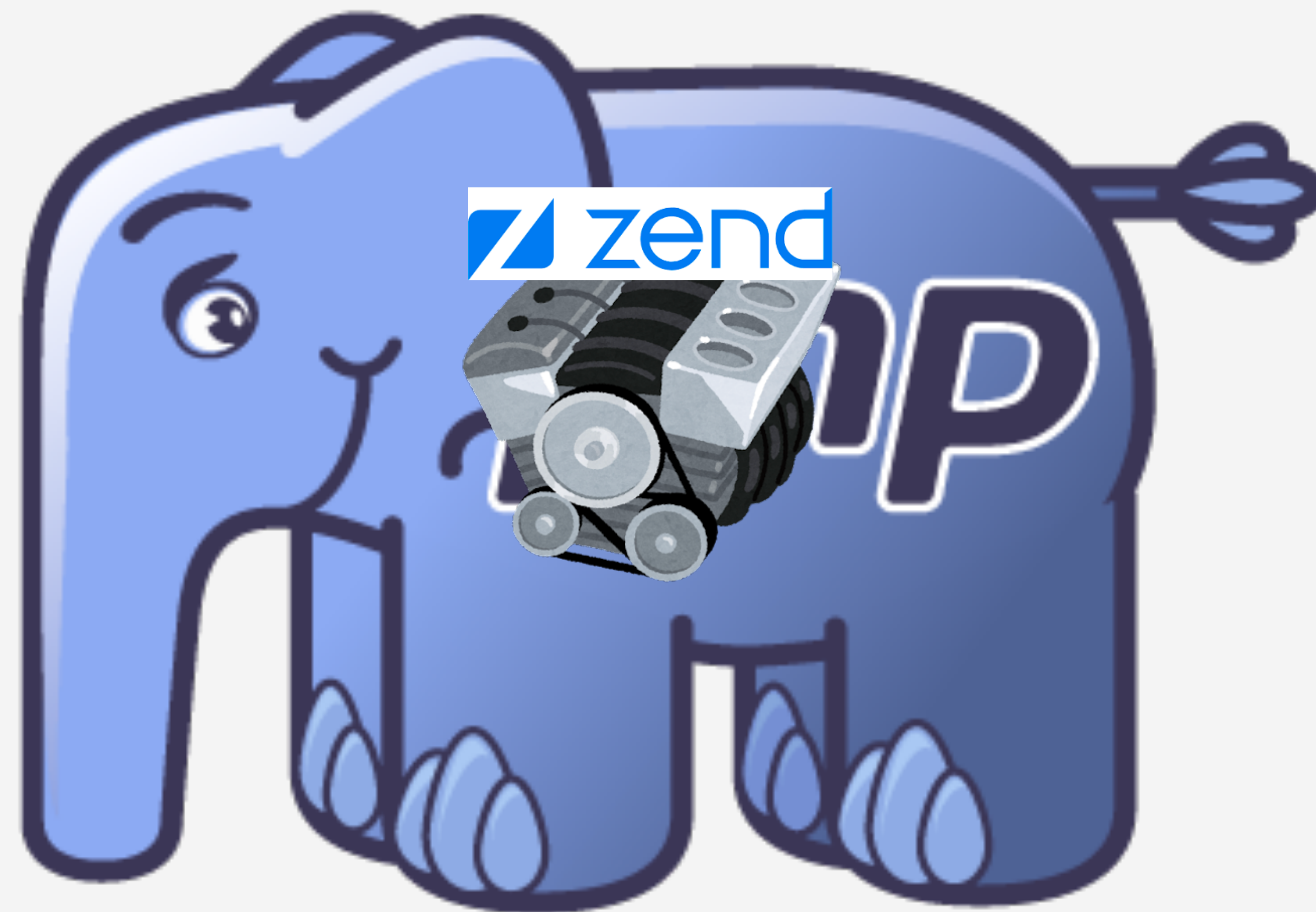
Zend Engine

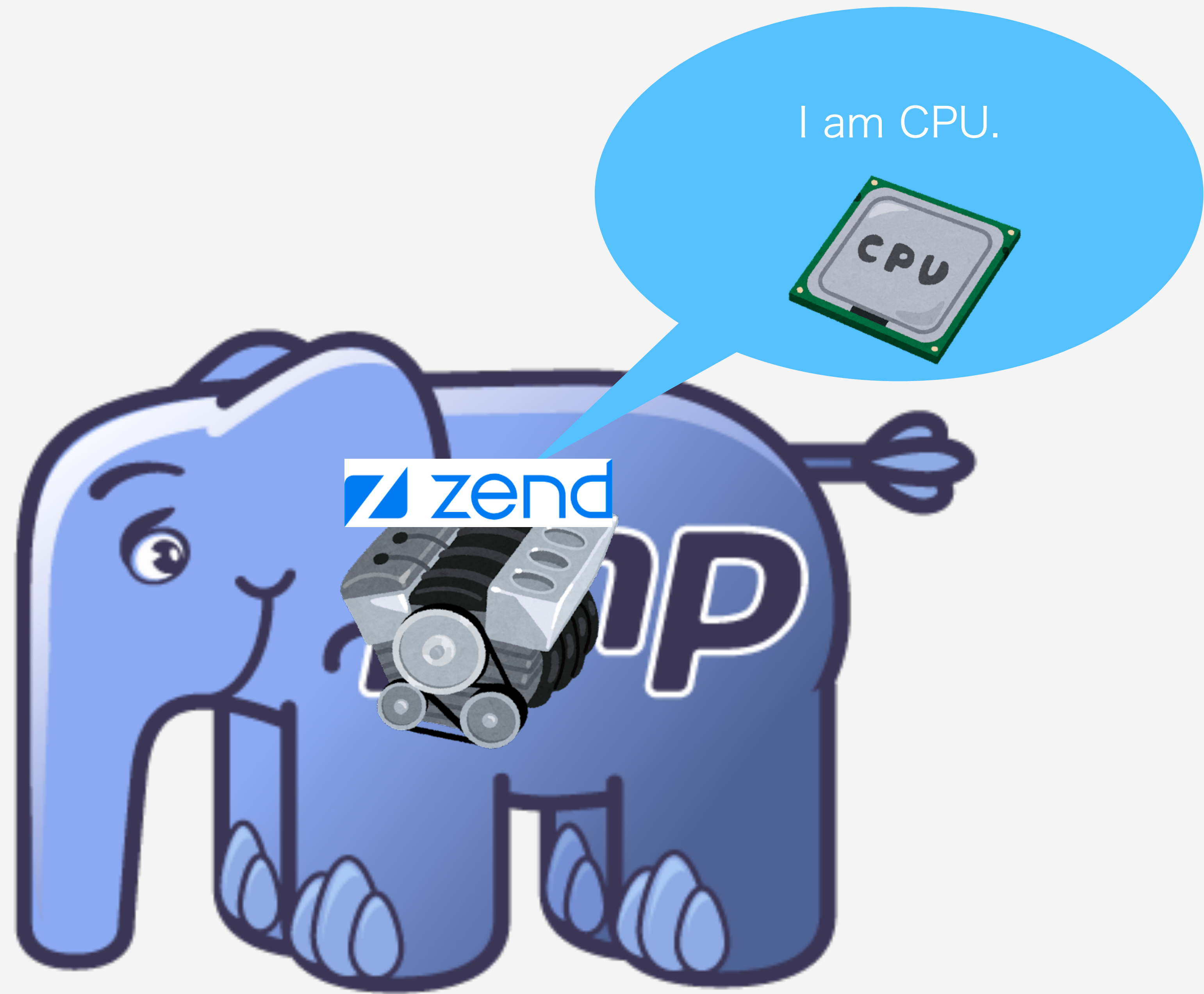
作者	アンディ・ガトマンズ、ゼーブ・スラスキー
開発元	ゼンド・テクノロジーズ
初版	1999年7月19日 (23年前)
最新版	4.2.0 / 2022年12月8日 (9日前)
最新評価版	4.3.0-dev
リポジトリ	github.com/php/php-src 
プログラミング言語	C言語
サポート状況	開発中
種別	インタプリタ
ライセンス	Zend Engine License
公式サイト	www.zend.com/products/zend_engine 

[テンプレートを表示](#)

仮想機械？

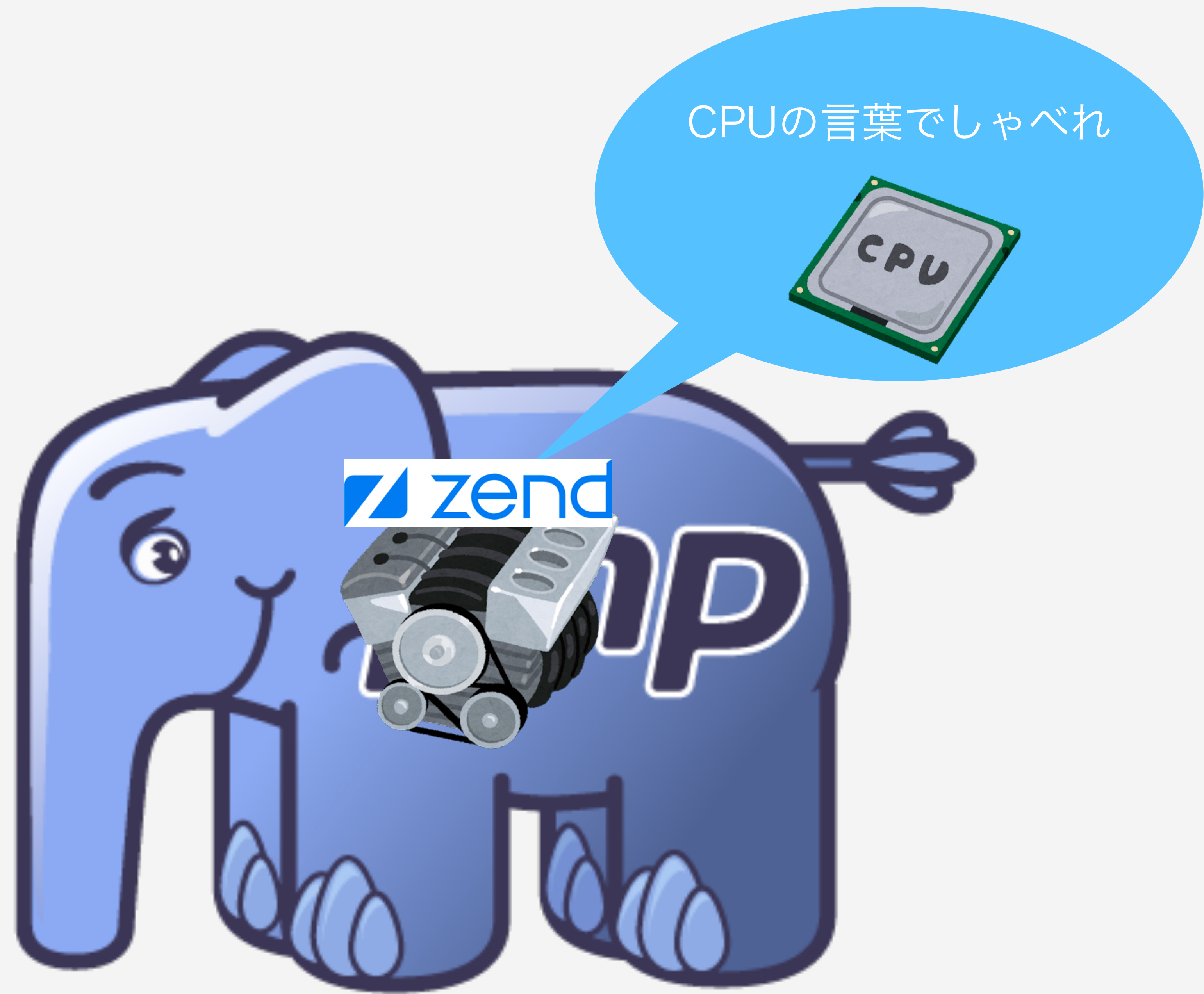






つまり？

自分のことをCPU
だと思い込んだ象



Zend Engine

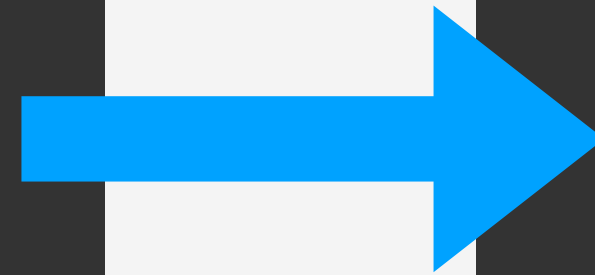
オペコード

opcode

PHPはオペコードにコンパイルされる

```
<?php
```

```
echo 1;
```



```
ECHO
```

```
1
```

PHPはオペコードにコンパイルされる

```
<?php
```

```
echo 'foo';
```



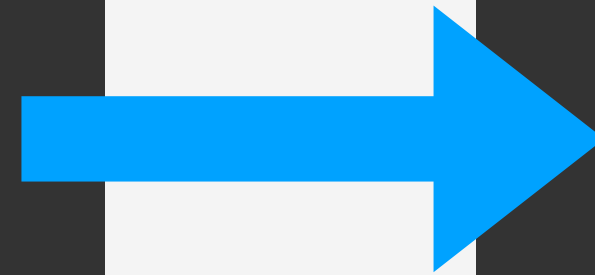
```
ECHO
```

```
'foo'
```

PHPはオペコードにコンパイルされる

```
<?php
```

```
print 'foo';
```



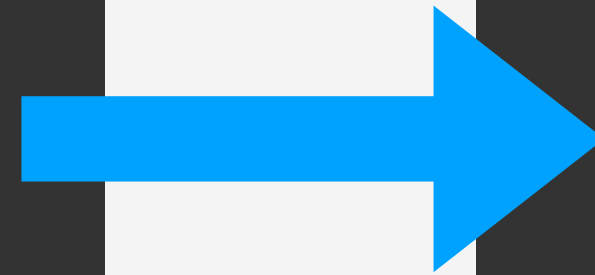
```
ECHO
```

```
'foo'
```

PHPはオペコードにコンパイルされる

```
<?php
```

```
echo 1 + 2;
```



```
ECHO
```

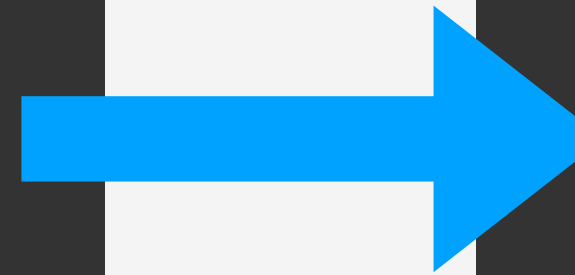
```
3
```

PHPはオペコードにコンパイルされる

```
<?php
```

```
$a = 1;
```

```
echo $a + 2;
```



```
ASSIGN
```

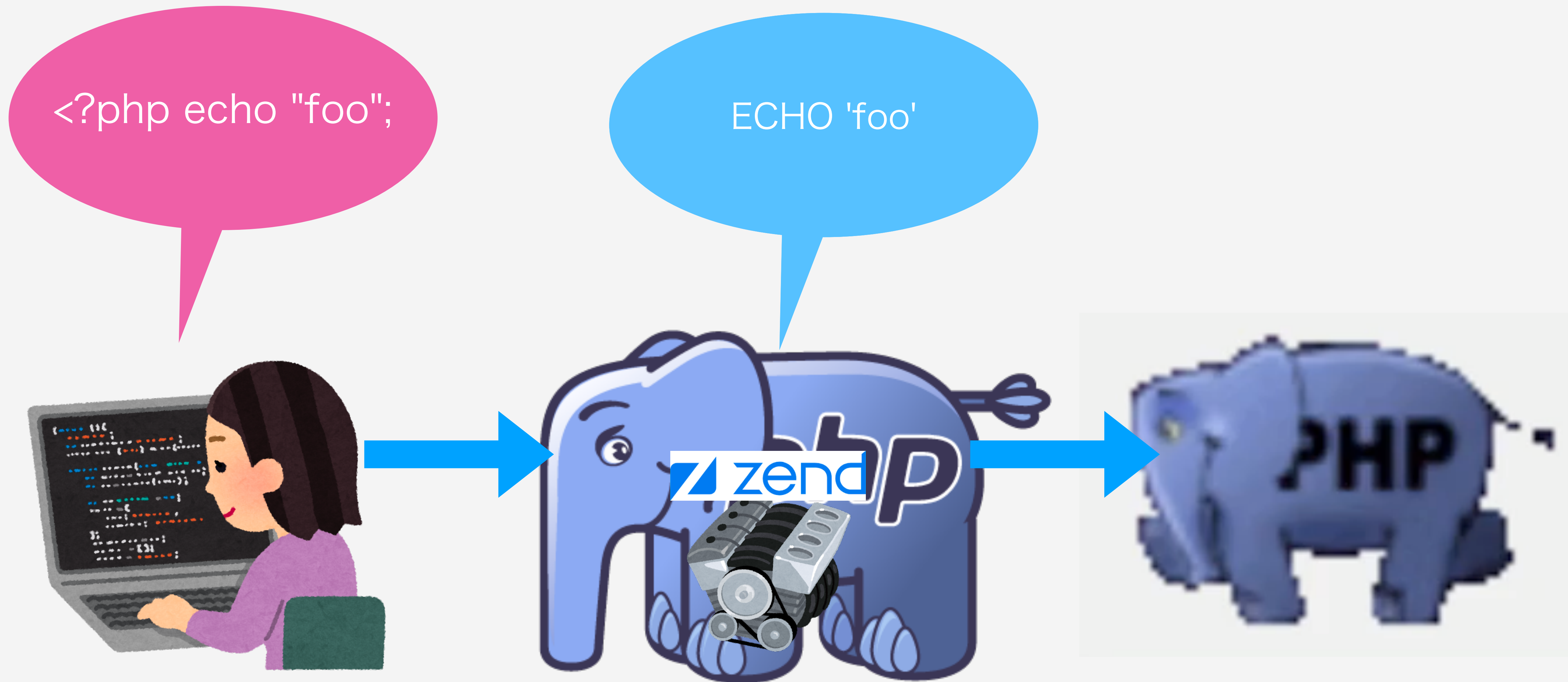
```
!0, 1
```

```
ADD
```

```
~2 !0, 2
```

```
ECHO
```

```
~2
```



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

これを極めれば
パフォーマンス改善に
役立つのか？



16



3



...

PHP Advent Calendar 2022 4日目



@tadsan

投稿日 2022年12月18日 3573 views

ZendEngineにえこひいきされる標準関数たち (前篇)



PHP, ZendEngine



こんにちは！ えこひいきしてますか？ 僕はしないようにしてます！

ところでPHPとかいう言語の標準ランタイム (つまりは、みなさんが普段使ってる `php` コマンド) はPHP スクリプトを直接実行するのではなく、実行前にコンパイラがPHPスクリプトをオペコードと呼ばれる中間表現(バイトコード)にコンパイルしてから、ZendEngineと呼ばれるVM(仮想機械)がオペコードを実行する構成になっていますよね。

PHPスクリプトで関数呼び出しをすると、あたりまえですが関数呼び出しをする命令にコンパイルされます。これはまったく不自然ではないことです。ところが、PHPの標準関数にはその前提に従わない、つまりは^{えこひいき}特別扱いされているものがあります。

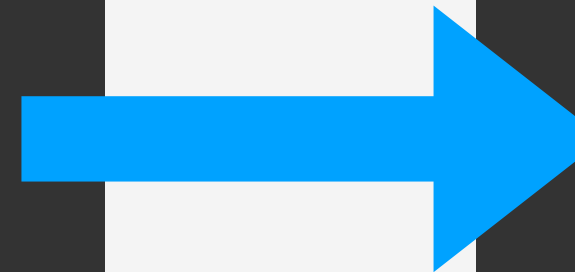
今回の記事はそんなえこひいきされた関数たちを眺めていきましょう ム(〃><)ノ 🎉

ここまでが前振り

関数呼び出し

```
<?php
```

```
var_dump( "foo" );
```



```
INIT_FCALL      'var_dump'  
SEND_VAL        'foo'  
DO_ICALL
```

関数呼び出しは
関数呼び出しに

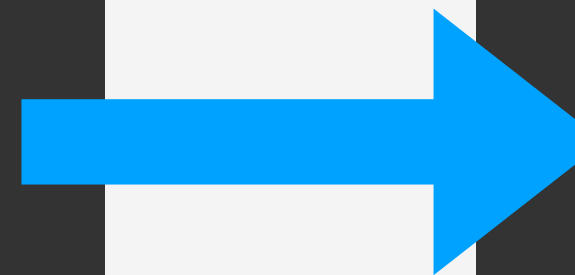
なるとは限らない

関数呼び出し

! ?

```
<?php
```

```
echo  
is_int( 'foo' );
```



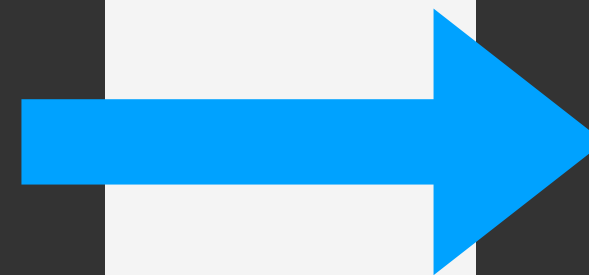
```
TYPE_CHECK 16 ~0 'foo'  
ECHO ~0
```

関数呼び出し

! ?

```
<?php
```

```
echo  
strlen( 'foo' );
```



ECHO

3

まとめ

『えこひいき』 とは何だったのか

- PHPの処理系の心臓部は“Zend Engine”と呼ばれる仮想機械
 - 実行時にOPCode(Zend Engineのマシン語)にコンパイルする
 - 現代的なスクリプト言語はほとんど同じような構造になっている
- PHPコードはOPCodeにコンパイルされるが一對一で変換されず、一部の関数呼び出しは特別な最適化が施される
- 一般的なアプリケーションではあまりパフォーマンスに寄与しないかも