入力＋検査＝型安全

Make PHP type safe by validating input

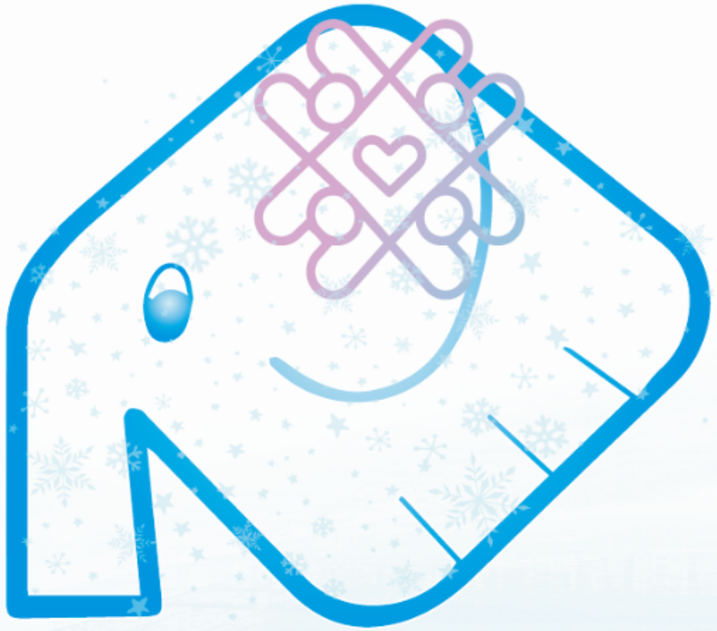
pixiv Inc.
USAMI Kenta

pixiv

2023-09-16
PHPカンファレンス沖縄2023

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）



PHP Conference Hokkaido 2024

12-13/Jan/2024

現在スポンサー募集中です！

各種お知らせは公式サイト/公式X(Twitter)アカウント
から行いますのでフォローをよろしくお願いします！

公式サイト: https://phpcon.hokkaido.jp/
公式X(Twitter)アカウント: @phpcondo

公式サイト

公式アカウント

さて

型、つけてますか？

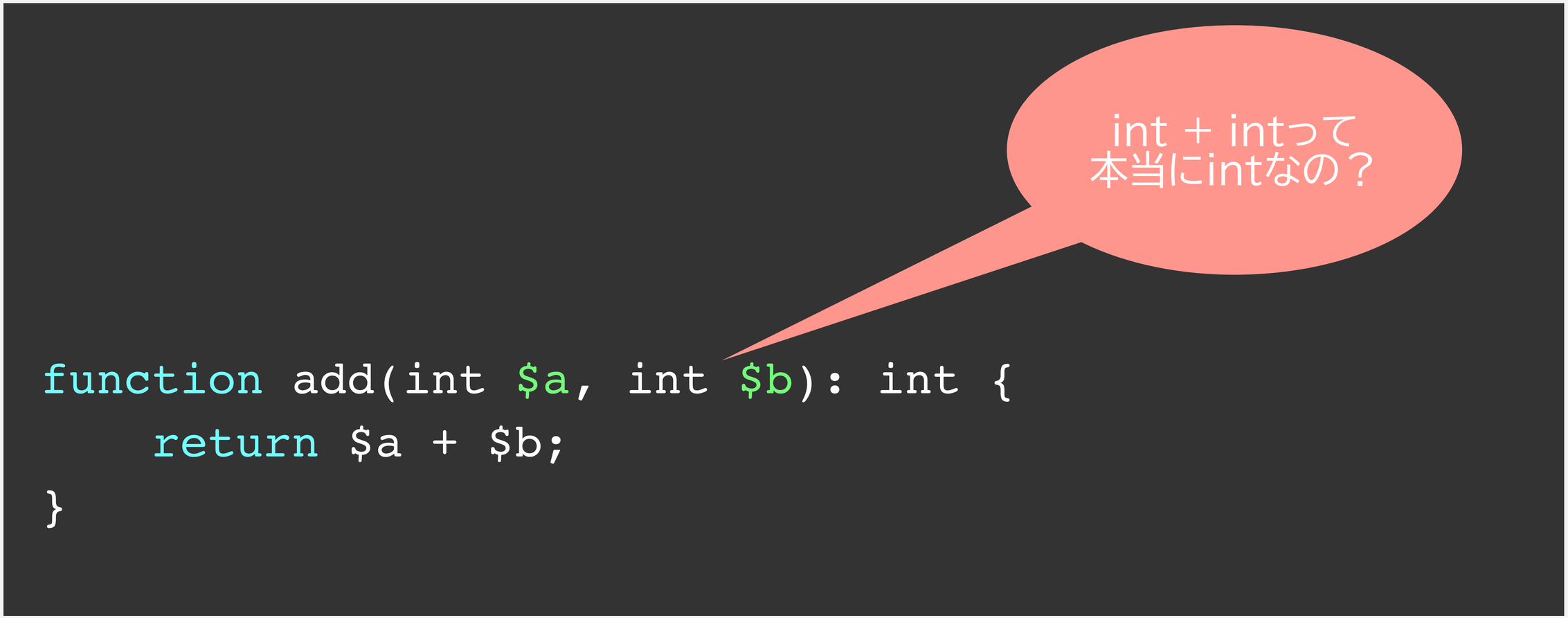
PHPの進化は型宣言の進化

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;
}
```


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;
}
```

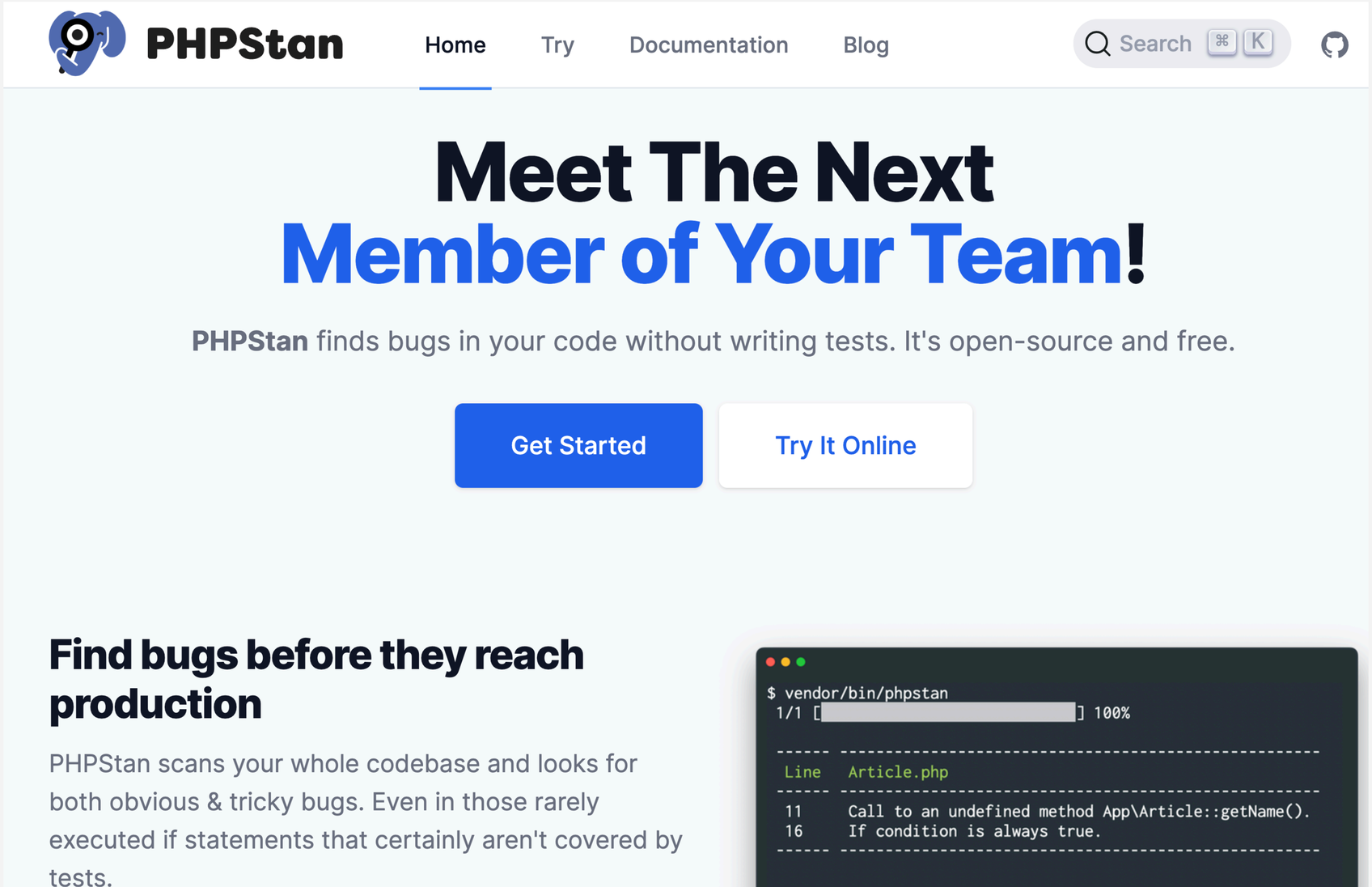
pixiv

ユニオン型宣言 (PHP8.0)

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

いたるところに型が書けるようになった

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クイックガイド 2023

 うさみけんた@tadsan

PHPStan（PHP Static Analysis Tool）はコードを実行せずに検査できるツールです。本稿では業務アプリケーションにPHPStanを導入するまでに押さえておきたい事柄を記述します。

本稿についての補遺は `https://scrapbox.io/php/Kaigi2023` にまとめているので、併せてお読みください。

導入

PHPStanは本稿記述時点の1.9.x系において、PHP 7.2以降で実行できます。PHPStanは

```
composer require --dev phpstan/phpstan \
    phpstan/extension-installer
```

のようなコマンドでのインストールが基本です。

プロジェクトルートの`phpstan.dist.neon`に、以下

コード上の多くの良くない傾向が示唆されているだろうと思いますが、まずは現状をポジティブに捉えて、PHPStanはプロジェクトを**さらに良くしていくための道具**なのだと考えましょう。

このコマンドを実行すると`phpstan-baseline.neon`というファイルが生成されます。先ほど**「ポジティブに捉えよう」**と言ったばかりなのですが、「`Function xxx not found.`」や「`Constant XXX not found.`」といったエラーは最初に解決しておくべきです。

PHPStanは特別な設定なしでコードを解析できるように設計されています。特に`composer.json`で`autoload`が設定されていれば自動的に解析されるようになっています。スクリプト実行時に関数や定数が動的に定義される場合やクラス名が動的にエイリアスされる場合は、初期化ファイルを`composer.json`の`autoload.files`か、`phpstan.dist.neon`の`bootstrapFiles`に追加してください。たとえばCodeIgniterプロジェクトであれば`"vendor/codeigniter4/framework/system/Test/bootstrap.php"`を追加するのがよいでしょう。

さて、ここで作ったベースラインファイルは`include`

201X年、PHPは型の炎に包まれた!!

いまやPHPは静的型付きと言っても過言ではない

（本当か…？）

だが型なしは
滅びていなかった

型なしは
どこからくるの？

今回のお題

PHPカンファレンス沖縄2023

レギュラートーク 30分

入力＋検査＝型安全

うさみけんた　tadsan

☆ 6

みなさんは、ブラウザからのアクセスなど外部からの入力をどのように扱っているでしょうか。

PHPには$_GET/$_POSTのようなスーパーグローバル変数やfilter_var(), filter_input()などの関数、各フレームワーク独自の方法など、さまざまな方法があります。
また、近年ではJSONエンコードされたリクエストなどの用途も増えています。
このトークではHTTPレベルでの入出力をPHPがどう扱っているのか、どのように処理するのが安全なのか、さまざまな実装パターンと型の関係について紹介します。

そうです

今回のお題

PHPカンファレンス沖縄2023

レギュラートーク 30分

入力＋検査＝型安全

うさみけんた　tadsan

☆ 6

みなさんは、ブラウザからのアクセスなど外部からの入力をどのように扱っているでしょうか。

PHPには$_GET/$_POSTのようなスーパーグローバル変数やfilter_var(), filter_input()などの関数、各フレームワーク独自の方法など、さまざまな方法があります。
また、近年ではJSONエンコードされたリクエストなどの用途も増えています。
このトークではHTTPレベルでの入出力をPHPがどう扱っているのか、どのように処理するのが安全なのか、さまざまな実装パターンと型の関係について紹介します。

pixiv

みなさんに
覚えておいて
ほしいこと

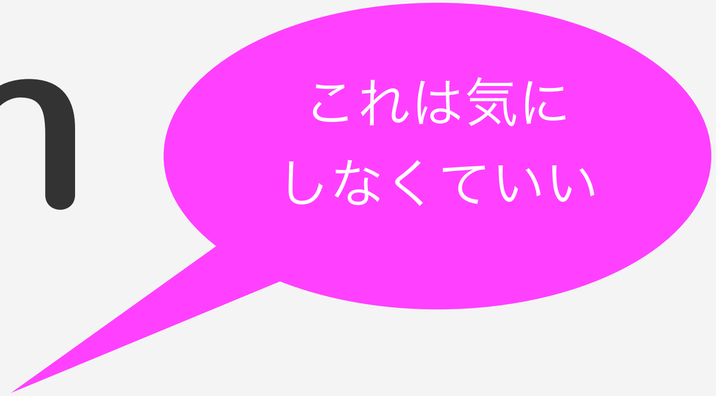
型なしとは何か

Validation

Validation
Sanitize

Validation
Sanitize
Escape

Validation
~~Sanitize~~
Escape



pixiv

自信をもって
説明できますか？

…ここから本題

クエリパラメータから ID値をとりたい

```
/** IDから記事を取得 */
function getById(int $id): Article
{
    // 中でデータベースに問い合わせ
}
```

実際あぶない

```
$id = $_GET['id'];
$article = getById($id);
```


pixiv

そもそも何者

```
$id = $_GET['id'];
$article = getById($id);
```


pixiv

ここでブラウザ開く

https://phpstan.org/try

型なしとはどういう状態か

- ある式の値の型が静的に定まっていない（mixed, TSのunknown/any）
- 型付けされた関数に渡したときに要求された文脈に従っているか
  （関数呼び出しをしたときにTypeErrorが出ないか安全だと言いきれない）
- PHPではいろんなところから型なしが吹き出す
  - スーパーグローバル、eval、json_decode()、unserialize ()、PDOStatement::fetch()

```
// ?id=1
$_GET['id'] === '1';
```

```
// ?id=a
$_GET['id'] === 'a';
```

```
// ?id[]=a
$_GET['id'] === ['a'];
```


pixiv

```
// ?id[foo]=bar
$_GET['id'] ===
        ['foo' => 'bar'];
```

キャストすれば
いいのか

危険ではないが
よくはない

```
$id = (int)$_GET['id'];
```


pixiv

```
// ?id=32x
$id = (int)$_GET['id'];
$id === 32;
```

```
if (is_numeric($_GET['id'])) {
    throw new BadRequestException();
}

$id = (int)$_GET['id'];
```


pixiv

is_numeric()

php Downloads Documentation Get Involved Help php8.2

is_numeric

(PHP 4, PHP 5, PHP 7, PHP 8)

is_numeric — 変数が数字または数値形式の文字列であるかを調べる

説明

```
is_numeric(mixed $value): bool
```

指定した変数が数値または 数値形式の文字列であるかどうかを調べます。

pixiv

数値形式の文字列

PHP の 文字列 は、 int や float と解釈できる場合は 数値と見なされます。

PHP 8.0.0 以降の正式な仕様は下記の通りです:

```
WHITESPACES     \s*
LNUM            [0-9]+
DNUM            ([0-9]*)[\.]{LNUM}) | ({LNUM}[\.][0-9]*)
EXPONENT_DNUM   (({LNUM} | {DNUM}) [eE][+-]? {LNUM})
INT_NUM_STRING   {WHITESPACES} [+-]? {LNUM} {WHITESPACES}
FLOAT_NUM_STRING {WHITESPACES} [+-]? {EXPONENT_DNUM} {WHITESPACES}
NUM_STRING      ({INT_NUM_STRING} | {FLOAT_NUM_STRING})
```

整数 123

小数 1.23

指数表記
1.84467440737l1E+1

前後にスペース

PHP は *先頭から始まる* 数値形式の文字列という概念も持っています。 これは、数値形式の文字列から始まり、その後に任意の文字が続く文字列です。

pixiv

だいたいのIDは正の整数だけ

```
// ?id=1.23
if (is_numeric($_GET['id'])) {
    throw new BadRequestException();
}

$id = (int)$_GET['id'];
```

```
$id = filter_var(
    $_GET['id'] ?? '',
    FILTER_VALIDATE_INT
);
```

安全に
int|false

pixiv

```
$id = filter_var(
    $_GET['id'] ?? '',
    FILTER_VALIDATE_INT,
    ['options' => [
        'min_range' => 1,
    ]]);
```

安全に
int|false

pixiv

ここでブラウザ開く

https://phpstan.org/try
https://php-play.dev/

filter_var()

Submit a Full Request Report a Bug

filter_var

(PHP 5 >= 5.2.0, PHP 7, PHP 8)
filter_var — 指定したフィルタでデータをフィルタリングする

説明

```
filter_var(mixed $value, int $filter = FILTER_DEFAULT, array|int $options = 0): mixed
```

パラメータ

value
フィルタリングする値。値をフィルタリングする前に、 内部的に 文字列への変換 が行われることに注意しましょう。

filter
適用するフィルタの ID。フィルタの型 に、利用できるフィルタの一覧があります。

pixiv

うまいこと値のバリデーションできる標準関数

……話は遡り
2007年

伝説のプレゼン

ITConversations
A Conversations Network Channel

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.

ホルモンのPHP (前半)

- Rasmus LerdorfがPHPを作った歴史的経緯を話している
- 昔のPHPコードとか開発体制とか
- 構文解析とか苦手なのでPHPの構文がめちゃくちゃだったという自虐
- 利己的な動機で始めたプロジェクトだが開発者を惹きつけるように工夫している様子
- 自宅サーバに入れたDrupalが遅いのでプロファイリングした

ホルモンのPHP (後半)

- 世の中にセキュリティの問題が多いウェブサイトが多すぎる
- IEとかFlashとかAdobe ReaderとかUTF-7とか罠がいっぱい
- 俺たちのウェブは完全に壊れている。特にXSSがヤバイ。JS実行されちゃう。
-

ホルモンのPHP (後半)

- 世の中にセキュリティの問題が多いウェブサイトが多すぎる
- IEとかFlashとかAdobe ReaderとかUTF-7とか罠がいっぱい
- 俺たちのウェブは完全に壊れている。特にXSSがヤバイ。JS実行されちゃう。
- だからfilter関数作ったよ！ サニタイズすれば絶対にXSS起こらない！

フィルタには
サニタイズ機能もある

~~フィルタには~~
~~サニタイズ機能もある~~
使わないで！！

サニタイズとは何か

わからん

？？？

今年書いた記事

225

211

PHP Advent Calendar 2022 24日目

@tadsan (園島 ぬぬ)

なぜ出力時のHTMLエスケープを省略してはならないのか

PHP xss

最終更新日 2023年04月13日 投稿日 2023年01月01日

メリークリスマス！ 週末もPHPを楽しんでますか？
ところでWebセキュリティはWebアプリケーションを公開する上で**基礎中の基礎**ですよね！

メジャーな脆弱性を作り込まないことはWeb開発においては専門技術ではなく、プロとしての基本です。
中でも**XSS** (クロスサイトスクリプティング Cross-Site Scripting)やインジェクションについての考慮は常に絶対に欠いてはならないものです。

現実にはプログラミングには自動車のような運転免許制度がないため、自動車学校に

フィルタ関数を
使えば解決するのか

今年書いた記事

225

211

PHP Advent Calendar 2022 24日目

@tadsan (園島 ぬぬ)

なぜ出力時のHTMLエスケープを省略してはならないのか

PHP xss

最終更新日 2023年04月13日 投稿日 2023年01月01日

メリークリスマス！ 週末もPHPを楽しんでますか？
ところでWebセキュリティはWebアプリケーションを公開する上で**基礎中の基礎**ですよね！

メジャーな脆弱性を作り込まないことはWeb開発においては専門技術ではなく、プロとしての基本です。
中でも**XSS** (クロスサイトスクリプティング Cross-Site Scripting)やインジェクションについての考慮は常に絶対に欠いてはならないものです。

現実にはプログラミングには自動車のような運転免許制度がないため、自動車学校に

なんだよ
外部公開してない
ライブラリの自慢かよ

キタカミの里から
帰ってきたら
公開したい！！

対応予定

スーパーグローバル($_XXX)
PSR-7 ServerRequest