純粋 vs 副作用
PHPはなぜ難しいのか？

Pure vs. Side Effects: Navigating PHP's Complexities

pixiv Inc.
USAMI Kenta

pixiv

2025-06-28 #phpcon
PHPConference Japan 2025

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）

PHPStanクイックガイド2023

2023/09/17に公開 2024/12/13

PHP PHPStan Tech

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。

> この記事の初出はPHPerKaigi 2023のパンフレット原稿です。

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` 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、以下のように記述してください。

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

PHPStan型付けマニュアル

2025/03/30に公開 2025/04/15

PHP PHPStan Tech

こんにちは！ 楽しくPHPStanを使っていますか？ それともPHPStanに使われていますか？

PHPStanは非常に賢く、容易にPHPコードの *「嫌な気配」* を検知してくれます。ただ、PHPStanが指摘することが常に正しいなどといったことはなく、いつでもプログラムが動作しているという事実が前提で、静的解析はその影を追っているに過ぎません。

PHPStanがどのようなメカニズムで型を付けているのか理解できていないと、自我を失って機械に言われるがままに意図しないコードを書かされる人形になってしまいます。本稿ではPHPStanを自律的に使うために前提となる知識を紹介します。

ここからプログラミング言語と型、そしてPHPについての議論を始めたいのですが、「PHP」という名前はプログラミング言語の名前であり、PHP言語で書かれたソースコードを実行するプログラム名でもあります。ややこしいのですが、単に**PHP**と書くと言語名、**php**と書くとPHPを実行するソフトウェアを指すことにしましょう。

にゃんだーすわん
にゃーん

目次

技術職向け夏インターンシップ『PIXIV SUMMER BOOT CAMP 2025』基盤/プラットフォームコース

インターン

ピクシブ株式会社 の求人一覧

ピクシブは、世界中のクリエイターが創作活動をもっと楽しめるような事業を展開しています。

募集終了

MISSION

Accelerate creativity.
創作活動を、もっと楽しくする。

勤務地の所在地

151-0051 東京都渋谷区千駄ヶ谷4-23-5-6F

今回のお題

PHP Conference Japan 2025

採択 2025/06/28 11:25〜 トラック1 - 1F 大展示　レギュラートーク(25分)

純粋 vs 副作用 〜 PHPはなぜ難しいのか？

うさみけんた　tadsan

☆ 3

純粋関数(pure function)という言葉を聞いたことはありますか？ 簡単にいうと、同じ引数を渡せば必ず同じ結果を返す関数のことで、「数学的な関数」と説明されることもあります。

同じ引数で同じ結果ということは『確実な再現性がある』ということで、「純粋」の概念を知って純粋と不純な処理を切り分けられれば、コードを見通しよく、テストしやすいコードにすることもできます。

いくつかの静的解析ツールとIDEは純粋性について分析を提供しており、意味のない純粋関数の呼び出しについて警告を与えてくれます。
これと関連して、PHP 8.5向けには `#[NoDiscard]` というattributeも追加され、戻り値を処理する必要がある実装をマークすることもできます。

では、既存実装をPureあるいはNoDiscardに明解に分類できるかというと… 現実には一筋縄にはいかない、さまざまな考慮事項があります。

本トークでは「純粋」および「副作用」という概念、そしてそれらをとりまく周辺事情についてお話しします。

pixiv

純粋関数 pure function

副作用 side effect

きいたことありますか？

pixiv

数学的な関数？

$$f(x) \quad \sin(x)$$

関数fのxに2を代入する

$$f(x) = 2x$$

$$f(2) = 4$$

~~関数fのxに2を代入する~~

2に関数fを適用する

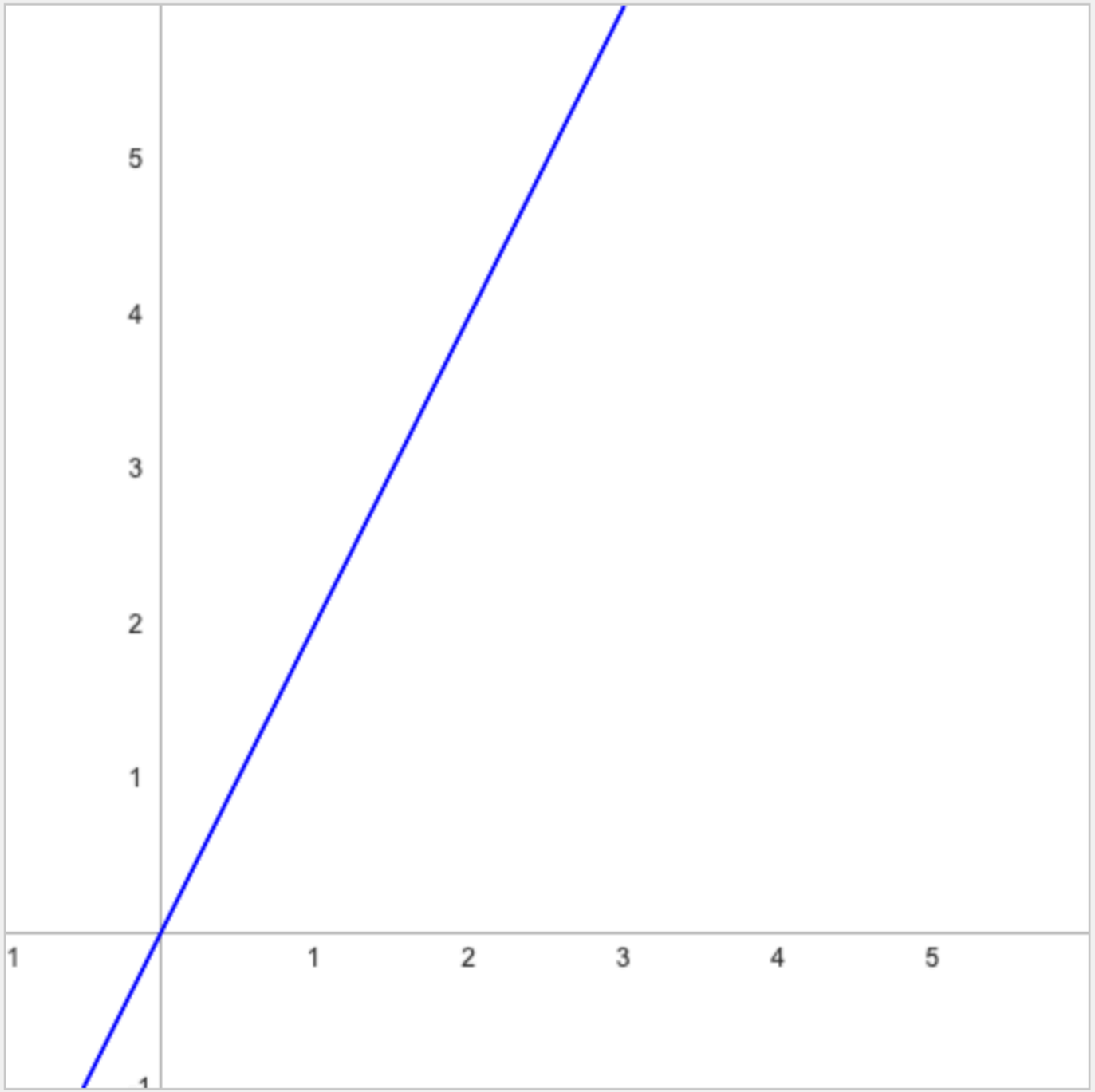
$$f(x) = 2x$$

$$f(2) = 4$$

$$f(x) = 2x$$

f(x)	f(0)	f(1)	f(2)	f(3)
2x	0	2	4	6

f(x)	f(0)	f(1)	f(2)	f(3)
2x	0	2	4	6



PHP Playground

PHP Playground let you to execute basic PHP code in real time using WebAssembly technology.

About PHP Playground?


< Request and Report

Donate

Version: 8.4

UI Theme:

HTML Preview

```php
<?php $f = fn($x) => 2 * $x ?>
<style>
td, th{padding: 1em; text-align: center; width: 3em}
th{background-color: #053B6D; color: #FFF}
th[scope="row"]{background-color: #0B60AC; color: #FFF}
</style>
<table border>
<tr><th><code>f(x)</code></th>
<?php foreach (range(0, 3) as $n): ?>
    <th><code>f(<?= htmlspecialchars($n) ?>)</code></th>
<?php endforeach ?>
</tr>
<tr><th scope="row">2x</td>
<?php foreach (array_map($f, range(0,3)) as $n): ?>
    <td><?= htmlspecialchars($n) ?></td>
<?php endforeach ?>
</tr>
</table>
```

$f(x)$	$f(0)$	$f(1)$	$f(2)$	$f(3)$
$2x$	0	2	4	6

いつどこで計算しても結果は同じ！

数学的な関数

$f(x)$ $\sin(x)$

プログラミングにおける関数

今日は特に区別しません！

@tadsan (園島 ぬぬ)

PHPの関数とメソッドは別物

PHP

投稿日 2023年03月05日 32528 views

✅ 関数はメソッドの気取った言い換えではありません

今年に入ってから6回くらい別の人に個別に説明したので記事に書きます。

一目でわかる関数とメソッド

```php
$array = [1, 2, 3];
// これは関数呼び出し
$length = count($array);

$object = new ArrayObject($array);
// これはメソッド呼び出し
$lenth = $object->count();
```


pixiv

$$f(x) = 2x$$

```
$f = function($x) {
        return 2 * $x;
};
```

$$f(x) = 2x$$

```
$f = fn($x) => 2 * $x;
```

この記事は最終更新日から5年以上が経過しています。

@tadsan (園島 ぬぬ)

PHPの無名関数式とは何か、そしてPHP7.4のアロー関数にそなえよう

PHP

最終更新日 2019年08月08日　投稿日 2019年08月04日　31760 views

PHP 7.4は現在beta1までリリースされたので、今頃はみなさまも12月のPHP 7.4.0正式リリースに向けて遊び倒してるところだと存じます。7.4でPHPに新たに導入される文法はいくつかありますが、その目玉のひとつがArrow Functions 2.0です。

ちなみにこの記事は「**アロー関数はすべての無名関数式の代替になるわけではない**」として書きはじめたので、今回本当に書きたかったのはそのあたりです。

最初にまとめると、「**アロー関数は簡単なことを短く書ける**」記法です。
複雑なことをするには**既存の`function`式の方が適してます。**

0〜3までの数に $f 関数を適用する

$$f(x) = 2x$$

f(x)	f(0)	f(1)	f(2)	f(3)
2x	0	2	4	6

0〜3までの数に $f 関数を適用する

```
$f = fn($x) => 2 * $x;
array_map($f, [0, 1, 2, 3]);
```

f(x)	f(0)	f(1)	f(2)	f(3)
2x	0	2	4	6

PHP Playground

PHP Playground let you to execute basic PHP code in real time using WebAssembly technology.

About PHP Playground?


< Request and Report

Donate

Version: 8.4

UI Theme:

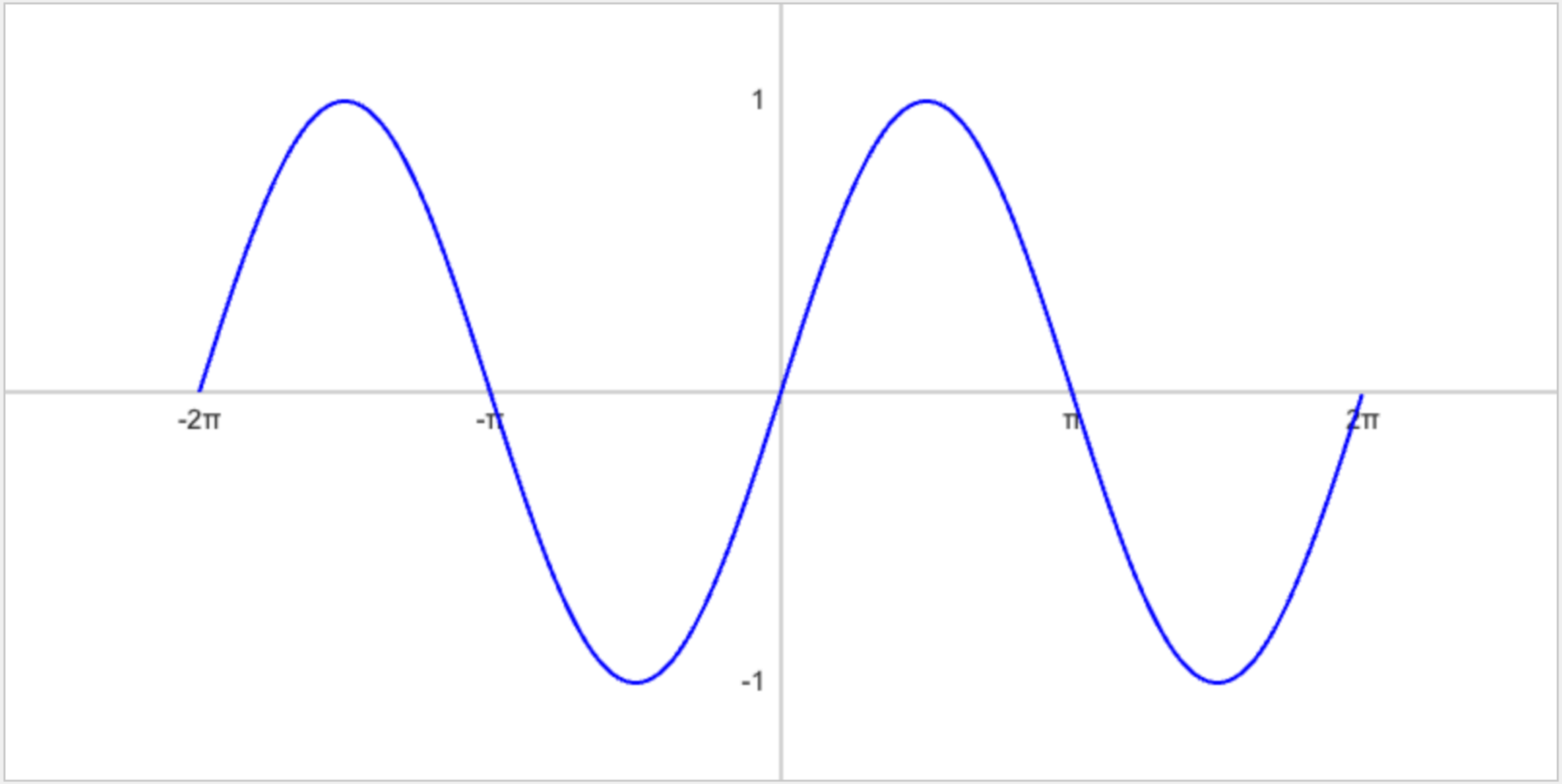
HTML Preview

```php
<?php $f = fn($x) => 2 * $x ?>
<style>
td, th{padding: 1em; text-align: center; width: 3em}
th{background-color: #053B6D; color: #FFF}
th[scope="row"]{background-color: #0B60AC; color: #FFF}
</style>
<table border>
<tr><th><code>f(x)</code></th>
<?php foreach (range(0, 3) as $n): ?>
    <th><code>f(<?= htmlspecialchars($n) ?>)</code></th>
<?php endforeach ?>
</tr>
<tr><th scope="row">2x</td>
<?php foreach (array_map($f, range(0,3)) as $n): ?>
    <td><?= htmlspecialchars($n) ?></td>
<?php endforeach ?>
</tr>
</table>
```

$f(x)$	$f(0)$	$f(1)$	$f(2)$	$f(3)$
$2x$	0	2	4	6

Math 関数

目次



PHP関数
＝
数学の関数

PHP関数
≠
数学の関数

関数の結果を画面に表示する

```php
$f = fn($x) => 2 * $x;

echo "f(2) = ", $f(2);
echo "sin(0) = ", sin(0);
```


pixiv

関数の結果を画面に表示する

```php
$f = fn($x) => 2 * $x;

var_dump(["f(2)" => $f(2)]);
var_dump(["sin(0)" => sin(0));
```

われわれは
var_dump() も
関数と呼んでいる

pixiv

関数の結果を画面に表示… しない

```php
$f = fn($x) => 2 * $x;

$f(2);
sin(0);
```

計算した結果を
どこにも使わない

計算資源の無駄！
SDGsに反する！

pixiv

なぜ同じような関数なのに違いが…

関数はひとつ！じゃない！！

```
printf("Hello, world!");

sprintf("Hello, world!");
```

画面に文字を表示する

文字列を生成して
捨てている！

pixiv

php Downloads Documentation Get Involved Help php8.4

PHP マニュアル › 関数リファレンス › テキスト処理 › 文字列 › String 関数

printf

(PHP 4, PHP 5, PHP 7, PHP 8)
printf — フォーマット済みの文字列を出力する

説明

```
printf(string $format, mixed ...$values): int
```

format にしたがって、出力を生成します。

画面に文字を表示する
（or HTTP出力）

作った文字列は
受け取らないと意味ない

php Downloads Documentation Get Involved Help php8.4

PHP マニュアル › 関数リファレンス › テキスト処理 › 文字列 › String 関数

Change language: Japanese

sprintf

(PHP 4, PHP 5, PHP 7, PHP 8)
sprintf — フォーマットされた文字列を返す

説明

```
sprintf(string $format, mixed ...$values): string
```

フォーマット文字列 **format** に基づき生成された文字列を返します。

pixiv

なぜ差がついたのか…

~~慢心・環境の違い~~

状況を整理しましょう

PHPの「関数」には種類がある

- **数学っぽい関数**
  - 同じ引数を渡したとき、関数の呼び出し結果（戻り値）は**必ず**同じになる
  - 戻り値は受け取らないと無駄になる

PHPの「関数」には種類がある

- **数学っぽい関数**
  - 同じ引数を渡したとき、関数の呼び出し結果（戻り値）は**必ず**同じになる
  - 戻り値は受け取らないと無駄になる
- **数学っぽくない関数**
  - 関数の呼び出し結果（戻り値）は別に受け取らなくてもいい
  - どこかに何かの影響を及ぼす

pixiv

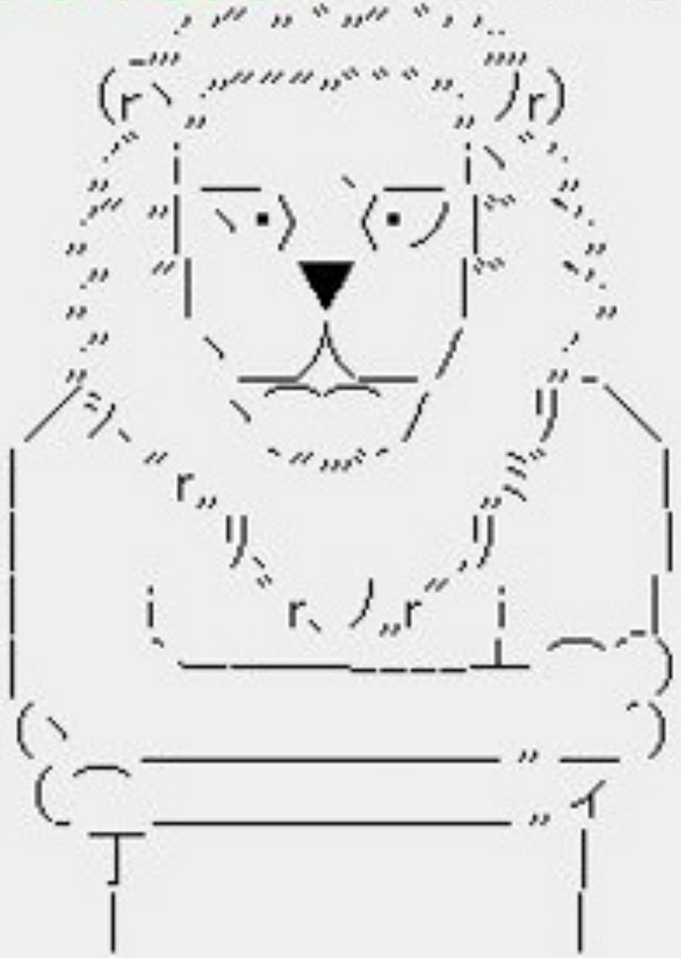
休憩

話は変わって

ユニットテストは好きですか？

好きですか？

名無し募集中。。。 : 2011/12/06(火) 18:32:24.37 0



テスト書いてないとかお前それ
@t_wadaの前でも
同じこと言えんの?

pixiv

好きかはともかく
お得なことはいっぱい

ユニットテストの基本スタイル

```
function twice(int $n): int {
    return $n * 2;
}
```

テスト対象の関数

pixiv

ユニットテストの基本スタイル

```
function twice(int $n): int {
    return $n * 2;
}

class TwiceFuncTest extends TestCase {
    function test(): void {
        $this->assertEquals(4, twice(2));
    }
}
```

期待値と結果を
比較してチェック

pixiv

ね、簡単でしょ？

簡単に済まないのが現実

純粋 vs 副作用
PHPはなぜ難しいのか？

Pure vs. Side Effects: Navigating PHP's Complexities

pixiv Inc.
USAMI Kenta

pixiv

2025-06-28 #phpcon
PHPConference Japan 2025

PHPの「関数」には種類がある

- **数学っぽい関数**
  - 同じ引数を渡したとき、関数の呼び出し結果（戻り値）は**必ず**同じになる
  - 戻り値は受け取らないと無駄になる
- **数学っぽくない関数**
  - 関数の呼び出し結果（戻り値）は別に受け取らなくてもいい
  - どこかに何かの影響を及ぼす

何かってなんだよ…

pixiv

いつの間にか
どこかの何かに
影響を及ぼす

pixiv

あるいは勝手に
どこかの何かの
影響を受ける

pixiv

何かってなんだよ…

PHPの「関数」には罠がいっぱい！

- **呼び出すごとに結果が変わる**（かもしれない）
  - 日時関数: time(), date(), strtotime(), …etc.
  - ランダム関数: rand(), random_bytes(), …etc
- **呼び出すごとにどこかに影響を及ぼす**（かもしれない）
  - 出力関数: var_dump(), header(), ob_start(), setcookie() …etc.
- **両方！** DB・ファイルシステム関数・外部通信

罠 = 副作用

pixiv

「テストしにくい」の正体

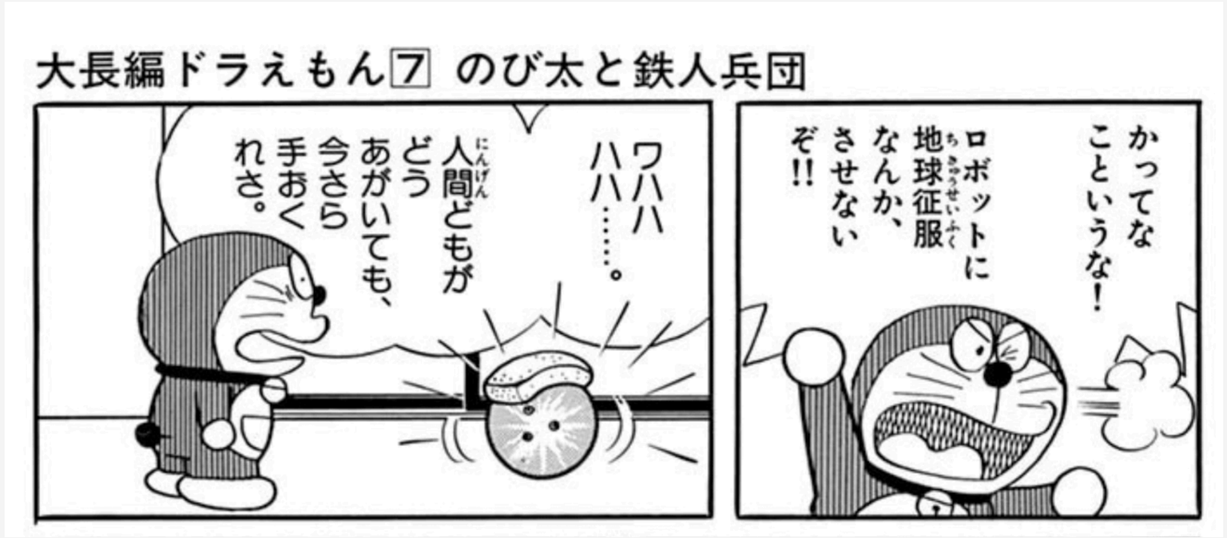
- **テスト対象を実行した結果が予想しにくい**
  - 外部の状態の影響を受ける・影響を受ける対象が多すぎて把握できない
  - 現在時刻・DB/ファイルの内容・外部APIの実行結果・ネットワーク… etc.
- **テスト対象を実行した結果に何が起こるか制御できない**
  - ファイルに書き込まれる、DBが更新される、不必要な外部リクエスト… etc.

不規則に副作用を起こすコードは容易に制御不能になる

秩序のないコードが
人間に牙を剥く

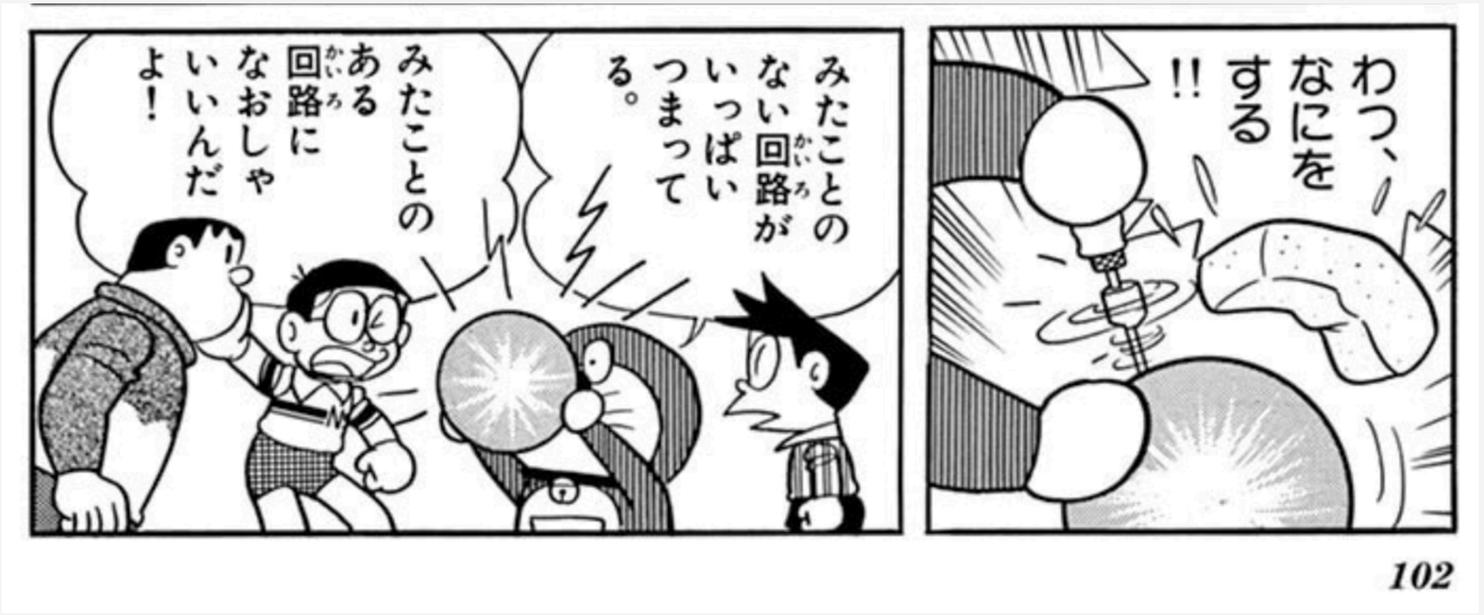
閑話休題

ドラえもん
ご存じですか？



侵略者が送り込んだ未知の技術の電子頭脳に対して…

工学的アプローチによって戦力化を提案する野比のび太



『大長編ドラえもん(7) のび太と鉄人兵団』
著：藤子・F・不二雄／藤子プロ
1987年2月25日初版第1刷、2011年第86刷
Kindle版 p97, p102より引用

みたことのある回路に
なおしゃいいんだよ！

のび太は
技術者の鑑

制御できないものは
制御できる部分に
ブレークダウンする

パターン: Ports and Adapters (構造に関するパターン)

別名:「Ports & Adapter」

別名:「Hexagonal Architecture」

ヘキサゴナルアーキテクチャ(Hexagonal architecture翻訳)
Alistair Cockburn著、tai2訳
2015年10月9日公開、2025年6月28日閲覧より引用

イベントが外側の世界からポートに届くと、特定テクノロジーのアダプターが、利用可能な手続き呼び出しか、メッセージにそれを変換して、アプリケーションに渡す。よろこばしいことに、アプリケーションは、入力デバイスの正体を知らない。アプリケーションがなにかを送る必要があるとき、それはポートを通じてアダプターに送られて、受信側のテクノロジーが必要とする信号を生む(人力であれ自動であれ)。アプリケーションは、実際に他方のアダプターの正体を知ることはなしに、全側面のアダプターと意味的に完全なやりとりをする。

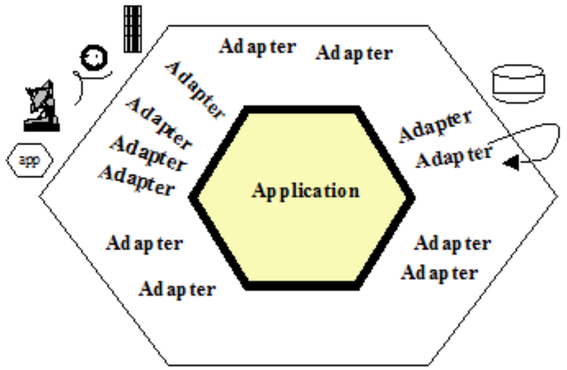


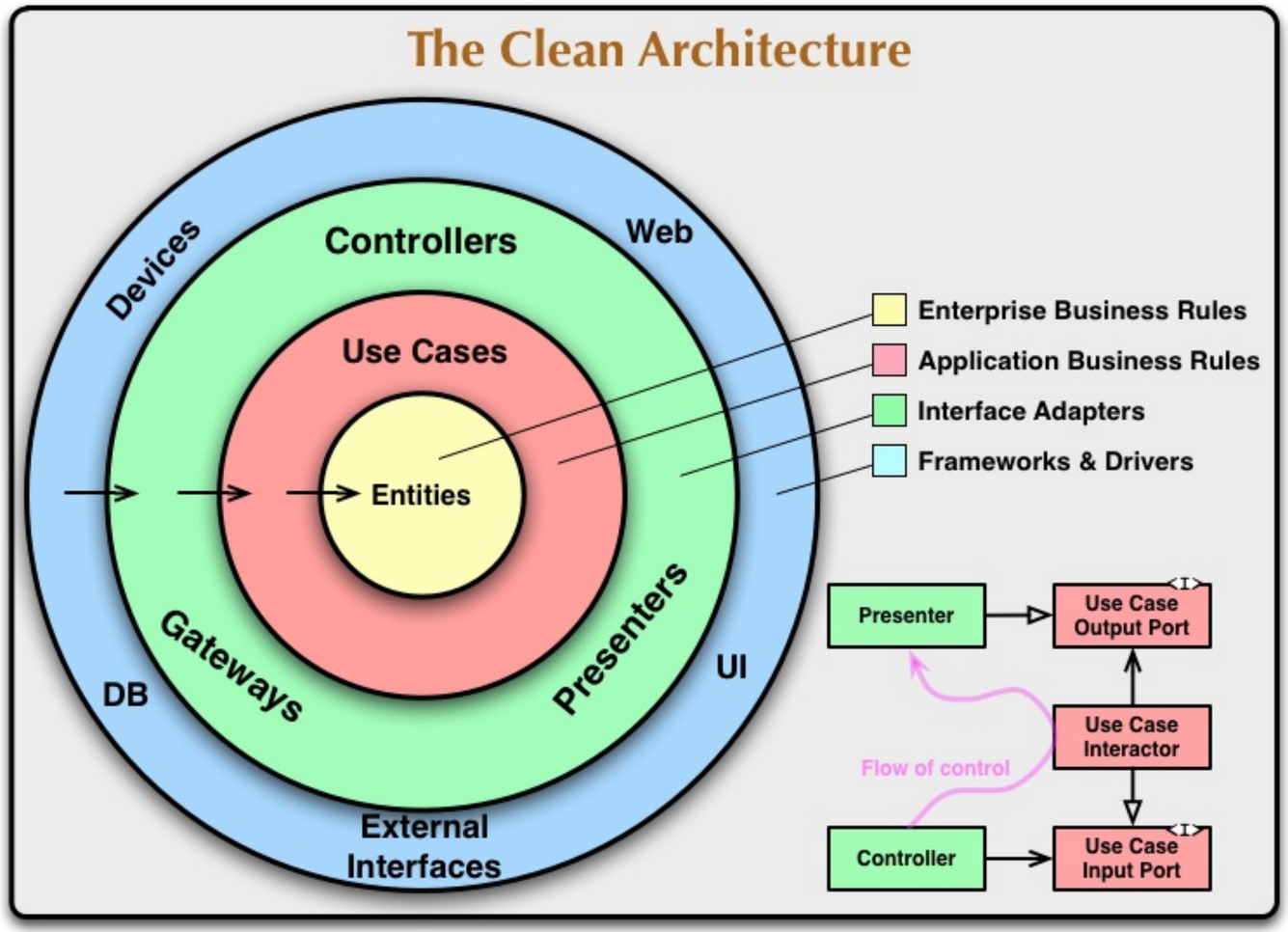
図1

クリーンアーキテクチャ(The Clean Architecture翻訳)

tai2 2015-10-05

Robert Martin (a.k.a. ボブおじさん) による、 The Clean Architecture の翻訳です。似たようなアーキテクチャである ヘキサゴナルアーキテクチャ も翻訳したので参考にしてください。

この記事を翻訳して公開したことは 8th Light, Inc. に報告済みです。いまのところ苦情は来ていません。



クリーンアーキテクチャ(The Clean Architecture翻訳)
Robert Martin著、tai2訳
2015年10月9日公開、2025年6月28日閲覧より引用

副作用がある部分を切り離す

- 任意の処理を純粋関数で書くということは「**できない**」
  - 現在時刻・DB/ファイルの内容・外部APIの実行結果・ネットワーク… etc.
- それらに依存する部分を抽象化して切り離すことで、
  テスト時にシンプルな偽物に差し替えてテストすることができる
  - DIは純粋関数型になれないオブジェクト指向言語に許された武器
  - 時刻への依存はPSR-20、乱数への依存は組込みのRandomizerで分離

PIXIV FANBOX

クリエイター

にゃんだーすわん

クリエイターページ

投稿する

ダッシュボード

投稿管理

プラン管理

クリエイター設定

ファン一覧

おたより

支援金管理/振込

FANBOXプリント

FANBOXプリントとは

コンテンツ提供者申請

クリエイター優待

にゃんだーすわん

ソフトウェア・ハードウェア

プロフィール　投稿　プラン　ショップ

pixiv

Zenn

投稿する

PHP の乱数実装がグダグダな話

2020/12/13に公開　2022/07/20

PHP　Tech

2022-07-19

これらの問題を解決する **Random Extension 5.x** 並びに **Random Extension Improvement** RFC が可決され、 `master` に merge されました。 PHP 8.2 より利用可能になります。

- https://wiki.php.net/rfc/rng_extension
- https://wiki.php.net/rfc/random_extension_improvement
- https://github.com/php/php-src/commit/4d8dd8d258ff365b146bcadcb277ede8992706d0

2022-06-18

これらの問題を解決するため、 PHP 8.2 に対して **Random Extension 5.x** の RFC が作成され、投票が始まっています

zeriyoshi

フォロー

Game Application Server-side Engineer

バッジを贈る

バッジを贈るとは →

目次

- 壊れた Mersenne Twister 実装の問題
- PHP 7.1 における srand() / rand() のエイリアス化の問題

pixiv

Zenn

投稿する

Open 2021/09/29にコメント追加 24

ランダムをテストすること

PHP PHPUnit

にゃんだーすわん 2021/09/29に更新

さいころを表現するクラスを作ろう。それはきっと以下のように使えるものだ。

```php
$dice = new Dice();

var_dump($dice->roll('3d6'));
// [5, 1, 2]
var_dump($dice->roll('5d6'));
// [4, 6, 1, 2, 4]
var_dump($dice->roll('3d10'));
// [3, 3, 2, 10, 4, 1, 7, 1, 9, 4]
```

引数の `"3d6"` のような文字列は「6面ダイスを3回振る」のような意味だ。

このクラスのテストはどうやれば実現できるだろうか。

1 2

スクラップをクローズ

他のユーザーの投稿を許可

10 ポスト

にゃんだーすわん

にゃーん

pixiv

制御できないものは
制御できる部分に
ブレークダウンする

「なんたらアーキテクチャ」は
構造をわかりやすく
図示してくれるが、
必須ではない

プログラム上で制御しやすい基本単位

#とは

そうです

pixiv

純粋関数

純粋関数とは…

純粋の反対語は不純(impure)

- 参照透過性（referential transparency）
  - 雑に言うと「同じ入力に常に同じ結果を返す」という性質
- 副作用を持たない（No side effects）
  - 与えられたパラメータだけに基いて戻り値を計算して返す以外に、外部から観測できる影響を与えない
- 「純粋関数型」と呼ばれる言語(Haskellなど)は、不純な性質が混ざらない

基本的に…

こんな関数は純粋じゃない！

- 実装の中で不純な関数を使ったり、グローバル変数を書き換える
  - time(), file(), rand(), $_GET, …etc.
- 明示的にecho, printなどで出力している
- 変数参照(&$var)を使っている
  - クロージャで & を使わない外部変数参照(use)はOK

関数の結果を画面に表示… されない

```php
$f = fn($x) => 2 * $x;

$f(2);
sin(0);
```

計算した結果を
どこにも使わない

計算資源の無駄！
SDGsに反する！

pixiv

PHPStan

Home Try

Try out PHPStan and all of its features here in the edit

```
1 <?php declare(strict_types = 1);
2
3 $f = fn(int $x) => 2 * $x;
4 $f(2);
5 sin(0);
6
```

Found 2 errors

Line	Error	
4	Expression "$f(2)" on a separate line does not do anything.	expr.resultUnu
5	Call to function sin() on a separate line has no effect.	function.resultUnu

やるじゃん

全関数を純粋と不純に分けたい…

pixiv

PHPStanの純粋判定を支える仕組み

- functionMetadata.hasSideEffects
  - PHPの標準関数が副作用を持つかどうかを管理するメタデータ(配列)
  - いまのところ、副作用がないものだけhasSideEffects=>falseと記録
  - JetBrains/phpstorm-stubsをインポート+個別補正している
- ImpurePoints
  - 型と同じように、コード内で副作用が発生する箇所を管理している

PHPDoc @pure タグ

- DocCommentに@pureと書かれた関数は純粋として扱われる
  - その関数を呼び出した結果が無視されていると、もちろん警告
- @pureがついた関数の実装に副作用があると、きちんと警告してくれる
  - ただし、それでも@pureの呼び出し側が無効化されるわけではない

resources/functionMetadata.php

```
16
17      return [
18          'BackedEnum::from' => ['hasSideEffects' => false],
19          'BackedEnum::tryFrom' => ['hasSideEffects' => false],
20          'CURLFile::getFilename' => ['hasSideEffects' => false],
21          'CURLFile::getMimeType' => ['hasSideEffects' => false],
22          'CURLFile::getPostFilename' => ['hasSideEffects' => false],
23          'Cassandra\\Exception\\AlreadyExistsException::__construct' => ['hasSideEffects' => false],
24          'Cassandra\\Exception\\AuthenticationException::__construct' => ['hasSideEffects' => false],
25          'Cassandra\\Exception\\ConfigurationException::__construct' => ['hasSideEffects' => false],
26          'Cassandra\\Exception\\DivideByZeroException::__construct' => ['hasSideEffects' => false],
27          'Cassandra\\Exception\\DomainException::__construct' => ['hasSideEffects' => false],
28          'Cassandra\\Exception\\ExecutionException::__construct' => ['hasSideEffects' => false],
29          'Cassandra\\Exception\\InvalidArgumentException::__construct' => ['hasSideEffects' => false],
30          'Cassandra\\Exception\\InvalidQueryException::__construct' => ['hasSideEffects' => false],
31          'Cassandra\\Exception\\InvalidSyntaxException::__construct' => ['hasSideEffects' => false],
32          'Cassandra\\Exception\\IsBootstrappingException::__construct' => ['hasSideEffects' => false],
33          'Cassandra\\Exception\\LogicException::__construct' => ['hasSideEffects' => false],
```

JetBrains/phpstorm-stubs

- PhpStormで使用するための関数メタデータを管理しているパッケージ
  - #[JetBrains\PhpStorm\Pure]というアトリビュートで純粋性を定義
- **問題点**
  - #[Pure]とマークされている関数の基準が曖昧
  - #[Pure(mayDependOnGlobalScope: true)] ？？？？？

「不純な純粋」というTig-HugなOxymoron

pixiv

Add `#[Pure(true)]` to some functions where the return value is important #1724

Merged **isfedorov** merged 2 commits into `JetBrains:master` from `zonuexe:add-pure-sideeffect-attr` on Mar 15

Conversation 0 | Commits 2 | Checks 12 | Files changed 4

zonuexe commented on Mar 14 — Contributor

Added the missing `#[Pure(true)]` attribute to some functions that have side effects but whose return value is important.

We could add it to more functions, but this PR only marks functions that are well-known and whose return value is obviously something you should use.

zonuexe added 2 commits 3 months ago


Add #[Pure(true)] to some functions where the return value is important ✕ ad23658


Add #[Pure(true)] to ob_get_*() functions ✕ 295c069


isfedorov merged commit **c99d99a** into `JetBrains:master` on Mar 15

11 of 12 checks passed

View details | Revert

Reviewers
No reviews

Assignees
No one assigned

Labels
None yet

Projects
None yet

Milestone
No milestone

Fix bad `Pure` attributes #1730

Open **cs278** wants to merge 6 commits into `JetBrains:master` from `cs278:fix-pure-attrs`

Conversation 19 | Commits 6 | Checks 12 | Files changed 5

cs278 commented on Apr 11 • edited ▾ Contributor •••

This PR is based on *my* interpretation of the documentation on the attribute that `Pure(true)` means a function:

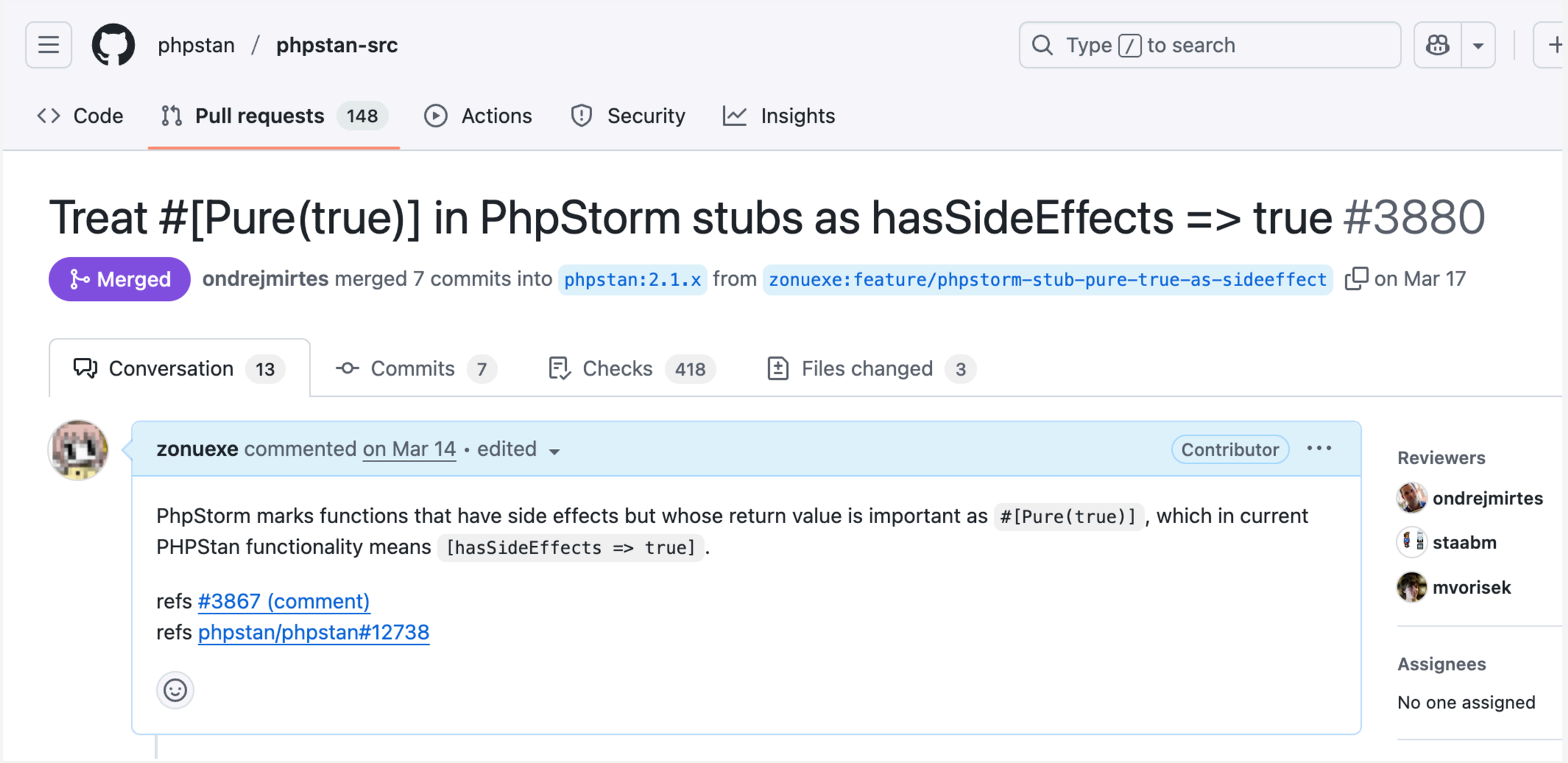
- Does **not** have any side effects on external state or the supplied parameters
- **and** the result depends on some external state outside of the supplied parameters.

This contradicts with the interpretation in #1724 by **@zonuexe** which was accepted:

> Added the missing `#[Pure(true)]` attribute to some functions **that have side effects** [emphasis mine] but whose return value is important.

The reasoning for each change is documented in the commit messages.

As such it reverts some of the changes made in that PR. If my interpretation is wrong then reject this, but please improve the documentation.


phpstan / phpstan-src
Type / to search
Code
Pull requests 148
Actions
Security
Insights
Treat #[Pure(true)] in PhpStorm stubs as hasSideEffects => true #3880
Merged
ondrejmirtes merged 7 commits into phpstan:2.1.x from zonuexe:feature/phpstorm-stub-pure-true-as-sideeffect on Mar 17
Conversation 13
Commits 7
Checks 418
Files changed 3
zonuexe commented on Mar 14 • edited
Contributor
PhpStorm marks functions that have side effects but whose return value is important as #[Pure(true)], which in current PHPStan functionality means [hasSideEffects => true].
refs #3867 (comment)
refs phpstan/phpstan#12738
Reviewers
ondrejmirtes
staabm
mvorisek
Assignees
No one assigned

pixiv

純粋じゃなくても
戻り値が大事な
関数はある

```
814  -        'curl_errno' => ['hasSideEffects' => false],
815  -        'curl_error' => ['hasSideEffects' => false],
     814 +    'curl_errno' => ['hasSideEffects' => true],
     815 +    'curl_error' => ['hasSideEffects' => true],
```

Comment on lines +814 to +815

staabm on Mar 16 — Contributor

This also looks wrong. I guess you should review all lines which changed from false to true

zonuexe on Mar 16 — Contributor, Author

Although they appear to be pure, they actually rely on some external state associated with the resource, so it makes sense that PhpStorm marks them as `#[Pure(true)]`.

While it may be convenient to make `curl_errno()` pure for convenience in common use cases, it is inconsistent with impure functions like `ob_get_level()` (#12577).

To solve this problem, `must_use` / `#[NoDiscard]` needs to be supported separately from `@pure` / `@impure`.

Reply...

Resolve conversation

CurlHandler/リソースは外部から観測できない内部状態に依存するので純粋と言いがたい

PHP 8.5に導入される新機能

#[NoDiscard]

- PHP RFC: Marking return values as important (#[\NoDiscard])
- このアトリビュートがついた関数の戻り値を無視すると、
  PHPランタイムがE_WARNINGを発生させてくれる
- #[NoDiscard]が導入される8.5ではビルトイン関数に対しては
  flock()とDateTimeImmutable::set*()のみ追加
  - 現状はちゃんと確認しないと事故るやつのみ、保守的にマークされてそう

pixiv

関数の結果を画面に表示… されない

```php
$f = fn($x) => 2 * $x;

$f(2);
sin(0);
```

計算した結果を
どこにも使わない

無駄なので除去する
最適化

pixiv

ではこれを実装すれば
うまくいくのか？

そう上手くいかない…

pixiv

PHPStan

Home

Try out PHPStan and all of its features here in

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

$f = fn(int $x) => 2 * $x;
array_map($f, [0, 1, 2, 3]);
array_map(sin(...), [0, 1, 2, 3]);

```

これは両方純粋なので
警告されるべき

PHP 8.1 – 8.4

No errors!

pixiv

PHPStan

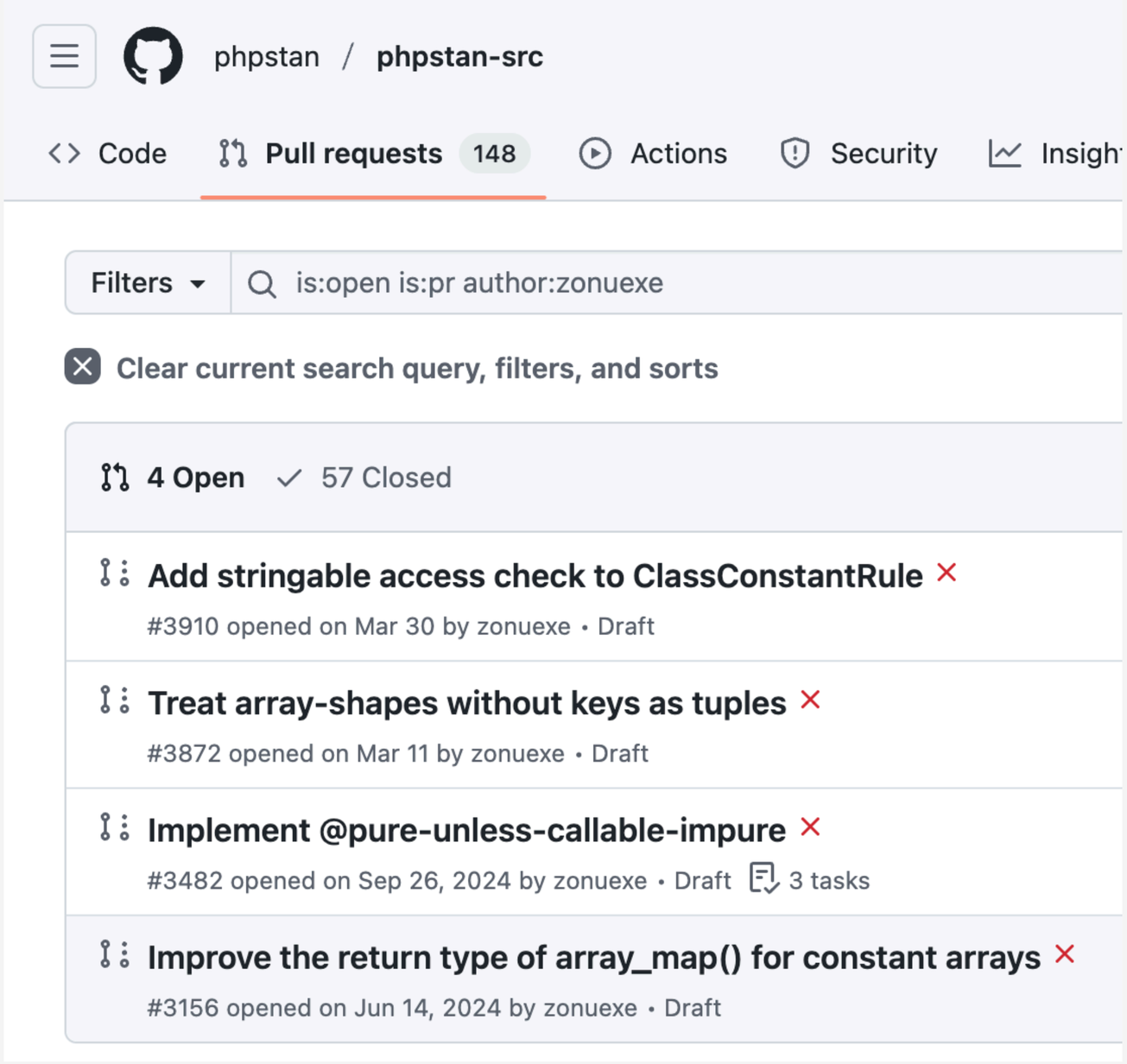
Try out PHPStan and all of its features here in

```php
1 <?php declare(strict_types = 1);
2
3 /** @pure */
4 function f(int $i): array {
5     return array_map(is_int(...), [$i]);
6 }
7
8 f(1);
9
```

純粋な呼び出しなので
警告されるべきではない

Error
Function f() return type has no value type specified in iterable type array. See: Solving PHPStan error "No value type specified in iterable type" missingType.iterableValue
Possibly impure call to function array_map() in pure function f(). possiblyImpure.functionCall
8 Call to function f() on a separate line has no effect. function.resultUnused

原因はわかっていて書きかけのPRがある


phpstan / phpstan-src
Code
Pull requests 148
Actions
Security
Insigh
Filters
is:open is:pr author:zonuexe
Clear current search query, filters, and sorts
4 Open
57 Closed
Add stringable access check to ClassConstantRule
#3910 opened on Mar 30 by zonuexe • Draft
Treat array-shapes without keys as tuples
#3872 opened on Mar 11 by zonuexe • Draft
Implement @pure-unless-callable-impure
#3482 opened on Sep 26, 2024 by zonuexe • Draft 3 tasks
Improve the return type of array_map() for constant arrays
#3156 opened on Jun 14, 2024 by zonuexe • Draft

pixiv

＿人人人人人人人人人＿
＞　僕の怠惰が原因　＜
￣Y^Y^Y^Y^Y^Y^Y^Y^Y￣

正直、体調不良と若干の燃えつきが重なっていた

PHPStanを
理解している人は
限られている

ガチでPHPStanを極めてみたい仲間を増やしたい！！！

pixiv

phpusers-ja Slack

pixiv

connpass
produced by Be PROUD

イベント検索

ダッシュボード　新着イベント

イベント管理　イベント作成

下書き中

イベントを編集する

7月
4

PHPStan mixed Live #1

PHPStanに関することなんでも！

募集内容	**参加枠1** 無料	参加者数 **0**人
出席登録	(イベント開始時間の2時間前から終了時間まで、参加者のみに公開されます)	

広告

イベントの説明

このイベントについて

- tadsanがPHPStanに関する何らかのコントリビュートをするための、もくもく会です
- PHPとPHPStanに関する質問があれば何でも答えます
- できれば毎週金曜に定期開催予定です

グループ　メンバーになる

Japan PHPStan User Group

イベント数 0回
メンバー数 1人

未公開

2025/07/04(金)
21:00 ～ 23:00

Googleカレンダー　icsファイル

このイベントに申し込む

受付票を見る

※受付や入場方法は主催者の案内に従ってください。

広告

pixiv