俺のPHPが合戦で負けるわけがない

There's No Way My PHP Could Lose in a Battle

pixiv Inc.
USAMI Kenta

pixiv

2024-11-30
紅白ぺぱ合戦

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
- 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）

connpass produced by Be PROUD

イベント検索

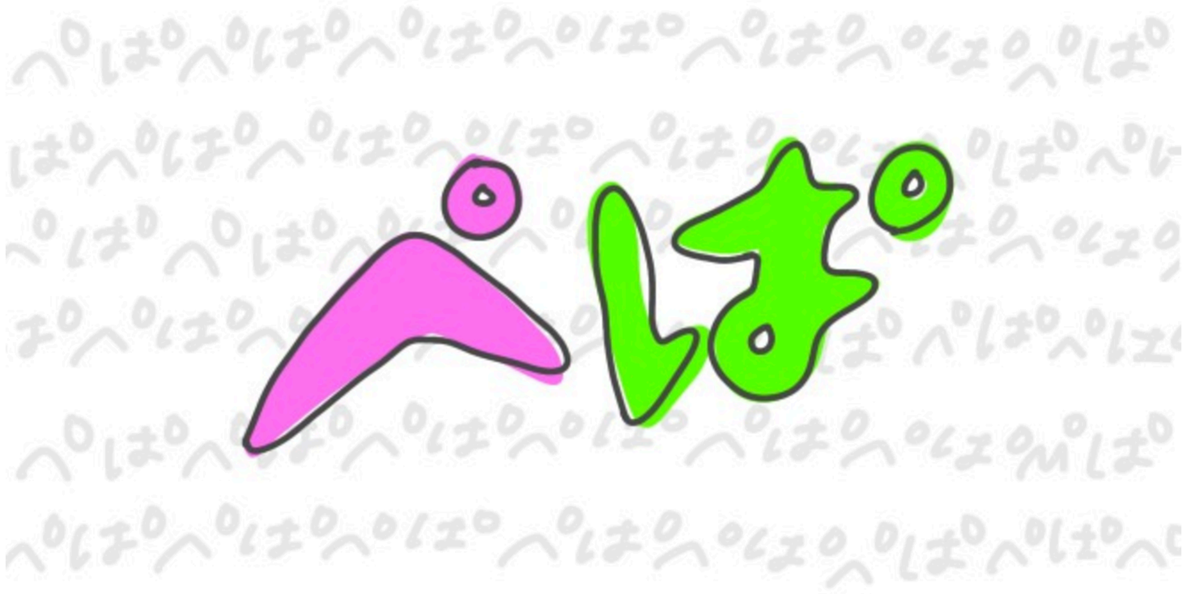
ダッシュボード　新着イベント

イベント管理　イベント作成

11月
30

紅白ぺぱ合戦

主催：asumikam



ハッシュタグ ： #cohackpp

フォロー参加者

開催前

2024/11/30(土)
17:00 ～ 21:00

Googleカレンダー　icsファイル

このイベントに参加できます

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。

申し込みキャンセル

募集期間
2024/08/27(火) 23:09 ～
2024/11/30(土) 21:00

イベントへのお問い合わせ

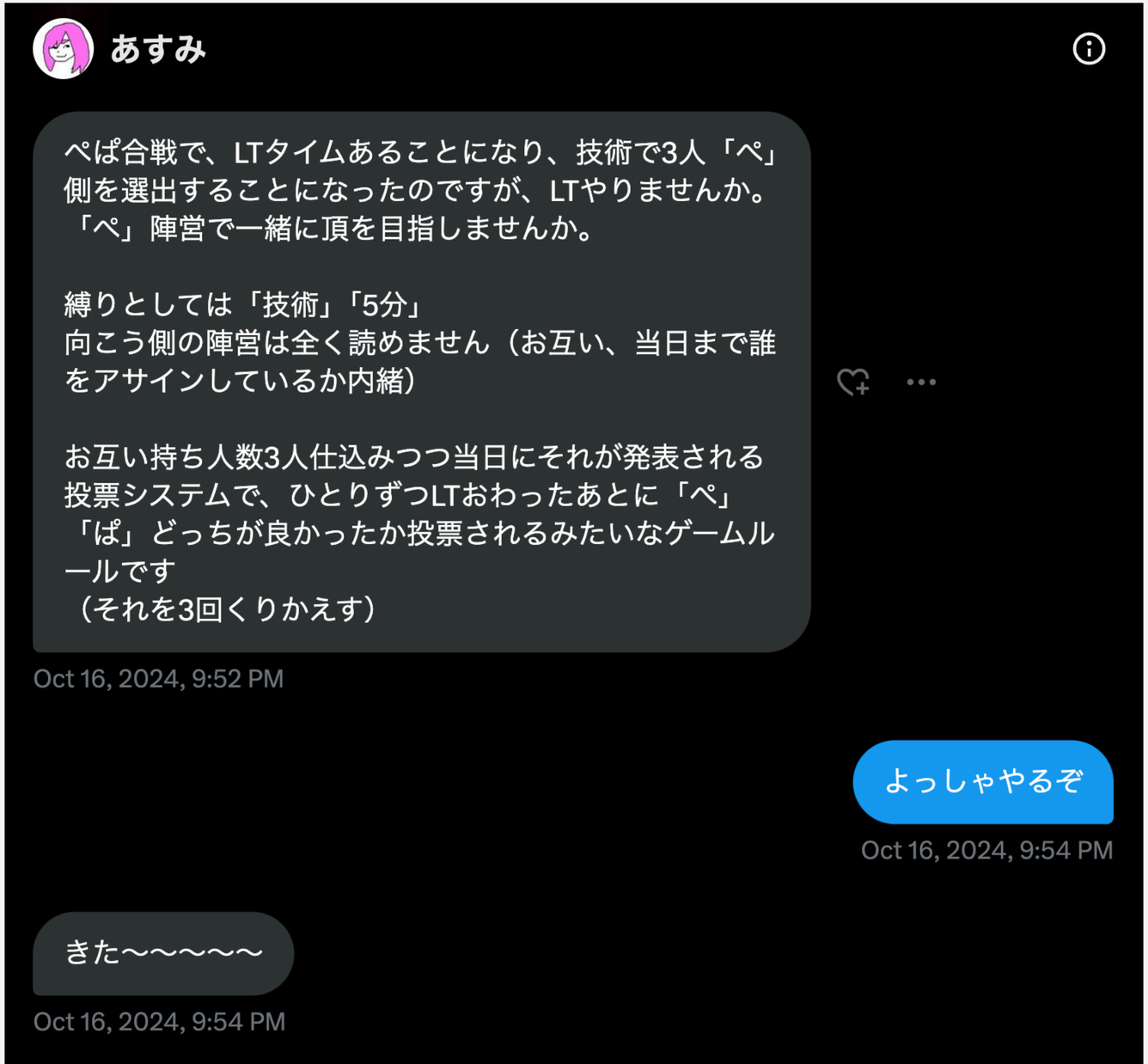
会場

おだわら市民交流センター UMECO 会議室5,6

〒250-0011 神奈川県小田原市栄町１丁目１－２７

pixiv

ぺとぱで戦えと言われた……



pixiv

Perlは魂の故郷

tadsanは
動的言語が大好き

なぜtadsanは
PHPを書いているのか

YAPCで就職したから(本当)

pixiv

こないだもこういう話をした

Perlの幻影

Chasing the Perl Ghost

pixiv Inc.
USAMI Kenta

pixiv

x

2024-10-06 #YAPC函館市電LT

pixiv

僕がPHP界隈で偉そうに話してるネタはYAPCとかビルコンの二番煎じだ！(本当)

大切なことはYAPCで教わった

Talk Information

Defence Against the Vim script Accepted #yapcasiaA

Kuniwak

github

Tweet

B!

Abstract

Vim scriptの静的解析ツールを書いて、メンテナンスしています。この経験から、Vim scriptの光と闇について話したいとおもいます。 なお、光と闇と書きましたが、光はありません。Perlと比べても、圧倒的なスコープの種類の多さ、謎の代入効果、厳しい文字列評価など、まさに劣らない機能をもっています。 これらの深淵な闇と、闇の魔術に対する防衛術を披露します。

Talk Details

pixiv

tadsanとPHPはビジネスの関係

プロフィールにも書いてる

pixiv

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

~~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~~

そんなわけで

お前をPHPerにしてやろうか

PHPとPerlの関係

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

Perlが捨てた CGIの末裔

PHPは過去と未来を繋ぐ言語

- PHPの仕様はCGIの時代から地続き
  - PerlやRubyはApache mod時代やPost WSGI時代に断絶がある
  - PHPはpreforkなサーバ上でもCGI互換のシェアードナッシング
    （すべての状態が隔離され、リクエストごとに初期化される）
- 汎用プログラミング言語の能力を持ちながら根幹がテンプレートエンジンにあるので、小規模からある程度大規模なアプリケーションにもフィットする

そういうことは
結構どうでもいい

PHPはどこでも65点がとれる言語

- 「現代におけるプロダクト開発とPHPを選定するワケ #phpkansai」
  - By @potato4d（PHPカンファレンス関西2017）
- 小難しいことを考えずに簡単にWebアプリを書ける（HTML+α）の言語
  - なんだかんだいってWEB+DB連携がさくっとできる言語ランタイムとしての存在感は健在
- **求められているのは素晴らしい言語ではなく簡単に使える言語**

Perl以上に雑に使える初心者向けWeb言語

pixiv

それでいいのか？

俺のPHPがこんなに静的なわけがない

There's No Way My PHP is so static

pixiv Inc.
USAMI Kenta

pixiv

2024-11-30
紅白ぺぱ合戦

PHPは静的型付き言語

Statically typed?

```
int n = 1;
int m = n + 2;
printf(”%d\n”);
```

```
int n = 1;
int m = n + 2;
printf("%d\n");
```

こんな当たり前の型をちんたら
書いてられないと思った人たちが
作った言語がPerl(という認識)

時代は型推論

TypeScript Download Docs Handbook Community

Playground TS Config Examples Help

v5.7.2 Run Export Share

```
1
2   const n = 1;
3   const m = n + 1;
4   //    ^? const m: number
5
6   function f () {
7       return 1 as const;
8   }
9
10  const o = f();
11  //    ^? const o: 1
12
13
```


pixiv

TypeScript Download Docs

Playground TS Config

v5.7.2 Run Export Share

```
1
2   const n = 1;
3   const m = n + 1;
4   //    ^? const m: number
5
6   function f () {
7       return 1 as const;
8   }
9
10  const o = f();
11  //    ^? const o: 1
12
13
```

型なんて人間が書かなくても
コードを解析すれば
一意に定まるじゃん、ワロス

pixiv

PHPでも型宣言？
とかできますけど…

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

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

DocCommentとかいう
机上の空論

pixiv

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

DocCommentとかいう
机上の空論

コーディング時の
ヒントとして役に立つ…
かもしれない

pixiv

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

intしか渡されないことは
実行時に保証される

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


pixiv

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

intしか渡されないことは
実行時に保証される

型宣言に反する値が
返されたら実行時エラー

pixiv

そうだけど
それだけじゃないんだ

PHPStan

Home Try Documentation Blog Search ⌘ K

Meet The Next Member of Your Team!

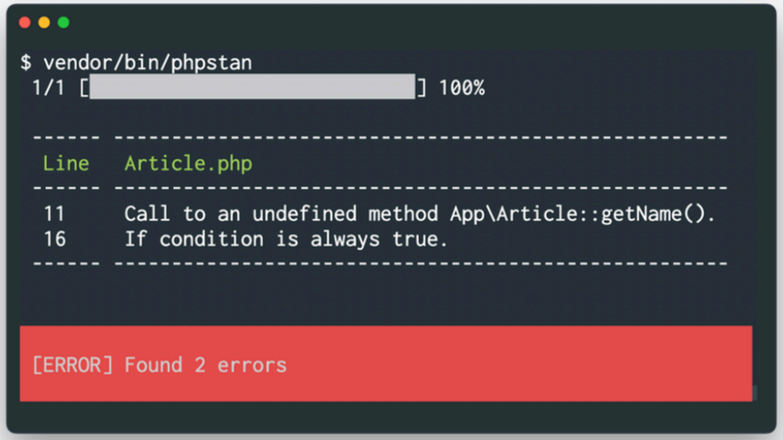
PHPStan finds bugs in your code without writing tests. It's open-source and free.

Get Started Try It Online

Find bugs before they reach production

PHPStan scans your whole codebase and looks for both obvious & tricky bugs. Even in those rarely executed if statements that certainly aren't covered by tests.

You can run it on your machine and in CI to prevent those bugs ever reaching your customers in production.



pixiv

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

$n = 1;
$m = $n + 1;

\PHPStan\dumpType(compact('n', 'm'));
```

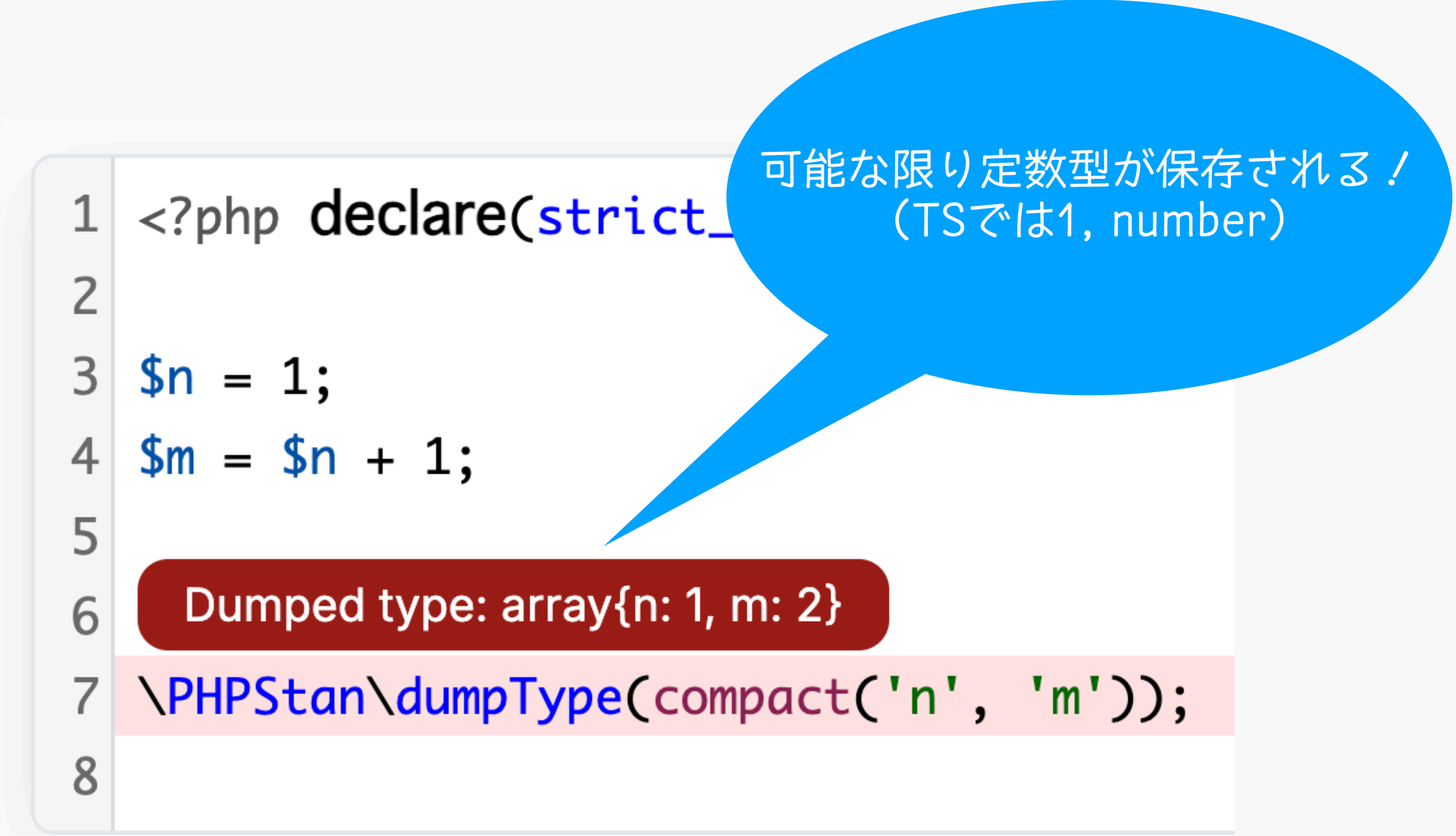
```php
<?php declare(strict_types = 1);

$n = 1;
$m = $n + 1;

Dumped type: array{n: 1, m: 2}
\PHPStan\dumpType(compact('n', 'm'));

```


pixiv


可能な限り定数型が保存される！
（TSでは1, number）
1 <?php declare(strict_
2
3 $n = 1;
4 $m = $n + 1;
5
6 Dumped type: array{n: 1, m: 2}
7 \PHPStan\dumpType(compact('n', 'm'));
8

pixiv

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

$foo = 'foo';
$bar = 'bar';

$foobar = $foo . $bar;

\PHPStan\dumpType($foobar);
```


pixiv

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

$foo = 'foo';
$bar = 'bar';

$foobar = $foo . $bar;

Dumped type: 'foobar'
\PHPStan\dumpType($foobar);

```


pixiv

```php
1  <?php declare(stric
2
3  $foo = 'foo';
4  $bar = 'bar';
5
6  $foobar = $foo . $bar;
7
8
9  \PHPStan\dumpType($foobar);
10
```

Dumped type: 'foobar'

可能な限り定数型が保存される！
（TSではstring）

pixiv

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

/** @return array{} */
function search(string $word, string $order): array
{
    if (!in_array($order, ['asc', 'desc'], true)) {
        throw new Exception('並び順は asc/desc で指定してね');
    }

    \PHPStan\dumpType($order);

    return [];
}

```


pixiv

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

/** @return array{} */
function search(string $word, string $order): array
{
    if (!in_array($order, ['asc', 'desc'], true)) {
        throw new Exception('並び順は asc/desc で指定してね');
    }
    Dumped type: 'asc'|'desc'
    \PHPStan\dumpType($order);

    return [];
}

```


pixiv

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

/** @return array{} */
function search(string $word, string $
{
    if (!in_array($order, ['asc', '
        throw new Exception('
    }
    Dumped type: 'asc'|'desc'
    \PHPStan\dumpType($order);

    return [];
}

```

分岐などのフローも解析して
型を絞り込んでくれる！！

pixiv

その辺の静的言語とは別の方向に尖りつつあるPHPにご期待ください