

配列の型と向き合う

2020-02-26 PHP勉強会@東京
#phpstudy

！ お前誰よ



- うさみけんた (@tadsan) / Zonu.EXE
- GitHub/Packagistでは id: zonuexe
- ピクシブ株式会社 pixiv運営本部
- Emacs Lisper, PHPer, Rubyist
- Emacs PHP Modeのメンテナ引き継ぎました
- 好きなリスプはEmacs Lispです
- Qiitaに記事を書いたり変なコメントしてるよ

pixiv



お絵描きがもっと楽しくなる場所を創る

pixiv 2018年度新卒
エントリー受付中



Friends of Emacs-PHP development

Join us!

<?php

- Repositories 25
- People 5
- Teams 1
- Projects 0
- Settings

Find a repository... Type: All Language: All

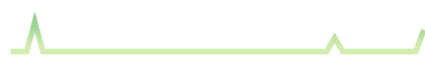
Customize pinned repositories New

php-runtime.el

PHP-Emacs bridge, call PHP function from Emacs

- php
- emacs
- melpa

Emacs Lisp 2 1 GPL-3.0 Updated 2 days ago



phpactor.el

Interface to Phpactor (an intelligent code-completion and refactoring tool for PHP)

Emacs Lisp 8 4 1 issue needs help Updated 3 days ago



php-mode

A PHP mode for GNU Emacs



Top languages

Emacs Lisp PHP

Most used topics Manage

emacs melpa php

major-mode

People 5 >



Invite someone



Search or jump to...



Pull requests

Issues

Marketplace

Explore



emacs-jp

Japan

<http://emacs-jp.github.com/>

Repositories 18

People 37

Teams 8

Projects 0

Settings

Find a repository...

Type: All

Language: All

Customize pinned repositories

New

reading-init.el

init.el読書会のページ

html

dotfiles

emacs

emacs-lisp

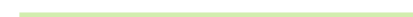
Ruby ★ 6 1 Updated 6 days ago



helm-c-yasnippet

Helm source for yasnippet

Emacs Lisp ★ 14 5 Updated on 21 Mar



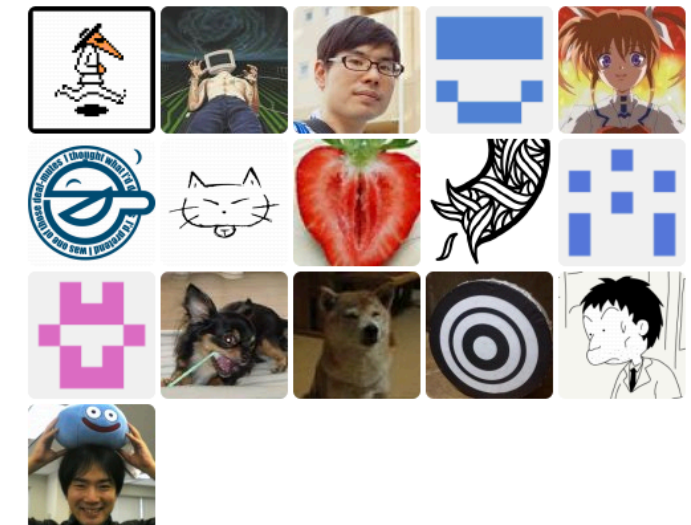
replace-colorthemes

Top languages

Emacs Lisp Ruby TeX

People

37



近況



文字を書いても
書いても終わらない

何が
あったのか？

本日のお題

Yesterday



tadsan  5:06 PM

まだLT枠空いてるのか... 何話そうかな



Shohei Anzai 11:09 PM

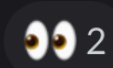
joined #area-tokyo along with harukei.

Today



tadsan  10:07 AM

[@koyhoge](#) LT空いてたら「配列の型と向き合う」話します



koyhoge 10:08 AM

りょうかいー。よろしくです。

型、ついて
ますか？



@tadsan 2017年10月23日に更新 13430 views

✎ 編集する ⚙️

PHPにコメントを書かせる、あるいは DocComment入門 (前編)

PHP

PHPDoc

⚠️ この記事は最終更新日から1年以上が経過しています。

コメントを書く必要がないほど読める^{リーダブルな}コード、つまり名が体を表し、コメントで回りくどく弁解説明の必要がないほどわかりやすいソースコードは極めて理想的なコードではあります。



@tadsan

2019年01月30日に更新 24287 views

✎ 編集する ⚙️

2018年のPHPDoc事情とPSR-5

PHP

PSR

PHPDoc

⚠️ この記事は最終更新日から1年以上が経過しています。

PHPDocとは、クラスや関数などのブロックに記述できるDocComment内に記述する書式の通称です。この書式の情報源として時折PSR-5が参照されることがあるので簡単に状況をまとめます。

TL;DR



@tadsan

2020年01月29日に更新 10290 views

編集する



このPHPがテンプレートエンジンのくせに慎重すぎる (前篇)

PHP

この記事ではPackagistで公開可能な形式のPHPのライブラリ(Composerパッケージ)を公開するための道具立てを紹介します。あと、現代のPHPerはツールを組み合わせでサクッと開発しているんだという自慢です。

タイトルは「このTypeScriptがHello, world!のくせに慎重すぎる」と「この勇者が俺TUEEEくせに慎重すぎる」のぱくr...パロディです。

テンプレートエンジンのくせに型安全なんてなまいきな。

今回はLaravelなので皆様既に
PhpStormのために型を
つけてる前提でいきます

前提

mixedは
災しいのもと

| mixedすると何が起こるか

- 間違っって引数に渡しても誰も怒ってくれない
- PhpStormで補完が効かなくて非常に気持ち悪い
- 静的解析性も開発体験も損ねる

PhpStormでの
開発体験を上げるには
mixedを追放して
型を付けていくしかない

肝に命じて
ほしい

さて

日曜の夜くらいから
記事を書いていた

PHPDoc 型付けマニュアル (2020年2月版)

PHP PHPDoc

本文

📖 良い記事を書くには



前提として、この記事は**2020年2月時点**での各種ツールの対応状況について記述しています。

PHPにおけるDocCommentの基礎については、この記事を読む前に「[PHPにコメントを書かせる、あるいはDocComment入門（前編） - Qiita] (<https://qiita.com/tadsan/items/ab2d96c906110badf45e>)」に目を通してください。

この記事は執筆時点の最新版(**Phan 2.5.0**, **PHPStan 0.12.11**, **Psalm 3.9.3**, **PhpStorm 2020.1 EAP**)での実装状況をもとに記述しています。これらのツールは非常に短いペースで改善されているため、常に最新版を利用することを強く推奨します。

概要だけ書いて
さくっと投稿する
つもりだった

書いても
書

書いても
書

終らん

なので「配列」の
話だけします

@param array

@return array

@var array

書いてます
か？



っどさん
@tadsan



arrayはね、mixed

午後11:53 · 2019年10月5日 · [Twitter Web App](#)

1 いいねの数



こいつは何を
言っているのか

| return array

```
<?php

function getIds(): array {
    return ...
}

foreach (getIds() as $id) {
    $id
}
```

| return array

```
<?php
```

```
function getIds(): array {  
    return ...  
}
```

```
foreach (getIds() as $id) {  
    $id // ← こいつの型がわからん  
}
```

arrayは
mixed

arrayを
許しては
いけない

Laravelのコードを
読んでもarrayまみれ

じゃあ
どうすればいいの

@return int[]

@return BookId[]

解決

本当に？



配列の用途
は複数ある

リストの
array


```
$ids = [1, 5, 9];
```

```
// @return int[]  
// @return array<int>  
$ids = [1, 5, 9];
```

@return int[]

phpDocumentor

形式

@return array<int>

コレクションジェネリクス記法

旧PSR-5形式

@return

ArrayObject<int>

コレクションジェネリクス記法

旧PSR-5形式

連想配列の array

```
$id_name_table = [  
    1 => "田中",  
    5 => "佐藤",  
    9 => "山田",  
];
```



```
// @return array<int,string>
$人_name_table = [
    1 => "田中",
    5 => "佐藤",
    9 => "山田",
];
```


構造体の
array

```
$record = [  
    "id" => 1,  
    "name" => "田中",  
];
```

これはPSSR-5形式
では表現できない

```
// @return  
array{id:int, name:string}  
$record = [  
    "id" => 1,  
    "name" => "田中",  
];
```

// @return

array{id:int, name:string}

array-shapes 記法
または object-like array

// @return

array{id?:int, name:string}

名前と:の間に? を
付けると省略可能キ—

Object-like arrays

Psalm supports a special format for arrays where the key offsets are known: object-like arrays.

Given an array

```
["hello", "world", "foo" => new stdClass, 28 => false];
```

Psalm will type it internally as:

```
array{0: string, 1: string, foo: stdClass, 28: false}
```

You can specify types in that format yourself, e.g.

```
/** @return array{foo: string, bar: int} */
```


現在はPSR-5(Draft)から
外されたが、既にほとんどの
静的解析ツールには実
装されているので使える

ほとんどの？

Psalm
Phan
PHPStan

ほとんど=

PhpStorm以外

型付けられない
いいお前だけ

いいますぐ使おう

Psalm
Phan
PHPStan

安心ください

Code tools

PHP



deep-assoc-completion



klesun

[Overview](#)[Versions](#)

```
/** @param $segment = AmadeusBaggageAdapter::joinLocations()[0] */  
private static function transformSegment(array $segment)
```

```
{  
    $airline = $segment['|'];  
    return [  
        'segmentDetail'  
        'airline'  
        'departure'  
        'destinati'  
        'bagWithou'  
        'bagWithou'  
        'isStopover' string  
        'destinationCity' unknown  
        'airline' unknown  
        'flightNumber' unknown  
        'bookingClass' unknown  
        'rebookRequired' string  
    ]['raw'],  
    $amount',
```

```
$fullPricing = $this->ge  
if ($result['error'] = $  
$pricingRecord = $fullPr  
$pricingRecord['parsed']
```

```
['currentPricing'] = $pr  
['pricingBlocks'] = $pr  
rd = $this->getFlightSe  
rd = $this->getFlightSe  
rd = $this->getFlightSe
```


たとえばこういう
コードがあるじゃろ

```
/**
 * @param array{id:int, name:string} $record
 * @return array{id:int, name:string} $record
 */
function getRecord(int $id, array $record): array
{
    return [
        'id' => 123,
        'name' => '田中'
    ];
}
```

“

```
$record = getRecord();
```

```
echo $record['|'];
```

```
/** @var Book
```

```
$foo = null
```

 **id** = 123

 **name** = '田中'

Press ^. to choose the sel

”

“

```
addRecord( [
```

```
  |
```

```
] )
```



id



name

```
/*
```

```
$f
```

Press ^. to choose the selected (or

”

付けよう
型

入れよう

deep-assoc

いますぐ

ダウンロー
ド