# Attributeを極める

増補版

Mastering Attributes

pixiv Inc.
USAMI Kenta

pixiv



2022-03-25

# 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-）
- プログラミング言語にちょっとこだわりのある素人

# ゴランノスポンサー


888
8888888888
88888888888888888888
提　い供ーーーーーい
GMO INTERNET GROUP
new relic.
photoruction
pixiv
DIGITAL CIRCUS
PHPerKaigi 2023

# 先におことわり：

20分ではピクシブ百科事典の
事例紹介までたどりつけませんでした
(話はするけどあっさりめ)

さて

# PHPのアップデートサイクル

- PHPは年に一回バージョンアップされている
  - https://www.php.net/supported-versions.php
  - 最新バージョンは2022年12月リリースの8.2.x系
- いわゆるセマンティックバージョニングではない
  - 8.2.0… MAJOR.MINOR.RELASEの3レベル構成
  - 影響の大きなBC BreakはMAJOR

# PHP8.0で
# 変わったこと

Change language: 日本語


# Released!

PHP 8.0 は、PHP 言語のメジャーアップデートです。
このアップデートには、たくさんの新機能や最適化が含まれています。 たとえば 名前付き引数、 union 型、アトリビュート、コンストラクタでのプロパティのプロモーション、match 式、nullsafe 演算子、JIT、型システムの改善、エラーハンドリング、一貫性の向上などです。

**PHP 8 にアップデートしよう!**

# すべての機能を
# 使えていますか？

# その中でも存在感がありながら「何に使うの」度が高い

Change language: 日本語


# Released!

PHP 8.0 は、PHP 言語のメジャーアップデートです。
このアップデートには、たくさんの新機能や最適化が含まれています。 たとえば 名前付き引数、 union 型、アトリビュート、コンストラクタでのプロパティのプロモーション、match 式、nullsafe 演算子、JIT、型システムの改善、エラーハンドリング、一貫性の向上などです。

**PHP 8 にアップデートしよう!**


php
Downloads
Documentation
Get Involved
Help
php 8.2
Search
Change language:
日本語
php 8
Released!
PHP 8.0 は、PHP 言語のメジャーアップデートです。
このアップデートには、たくさんの新機能や最適化が含まれています。 たとえば 名前付き引数、 union 型、アトリビュート、コンストラクタでのプロパティのプロモーション、match 式、nullsafe 演算子、JIT、型システムの改善、エラーハンドリング、一貫性の向上などです。
PHP 8 にアップデートしよう!

# アトリビュート
# Attributes

# 実はPHPerKaigiでは

fortee

# PHPのDI、attributesとこれから

by 山岡広幸 / @hiro_y

PHPerKaigi 2021

採択 2021/03/28 14:50〜 Track A レギュラートーク（20分）

## PHPのDI、attributesとこれから

PHPerKaigi 2021

☆ 18 ビデオ スライド

山岡広幸 hiro_y

PHP 8で導入されたattributes。
今まで（やむをえず）PHPDocを用いて実装されてきた様々な書き方が、だんだん置き換わっていこうとしています。
今回は、DIの代表的なライブラリであるPHP-DIが新しいバージョン（7、2021年1月19日時点では未リリース）でどう変わろうとしているか、また、Symfonyなどの変化もあわせて紹介しつつ、PHPにおけるDIの行方について話したいと思います。



pixiv

fortee

# Attributesを使った
# メタプログラミング入門！

by おかしょい / 岡田 正平 / @okashoi

#[?] PHPerKaigi 2022

採択 2022/04/11 17:45〜 Track A LT（5分）

**Attributesを使ったメタプログラミング入門！** PHPerKaigi 2022

☆ 10 スライド

 おかしょい / 岡田 正平 okashoi

PHP8.0から追加された機能、Attributes。
これを使ったメタプログラミングの例として、クラスのプロパティにAttributesを付与することで



pixiv

# 毎年なにかと
# 言及されている

# アトリビュートってなんだ？

- 日本語での定訳としては**「属性」**
- もっとも知られているであろう用語としてはHTMLタグ
- あるもの(要素)の性質を表す情報のこと
- MDNではHTMLの属性について次のように説明している
  - 「属性は要素を拡張し、動作を変更したりメタデータを提供したりします。」
    Attribute（属性）- MDN Web Docs 用語集 より

<a href="?foo">



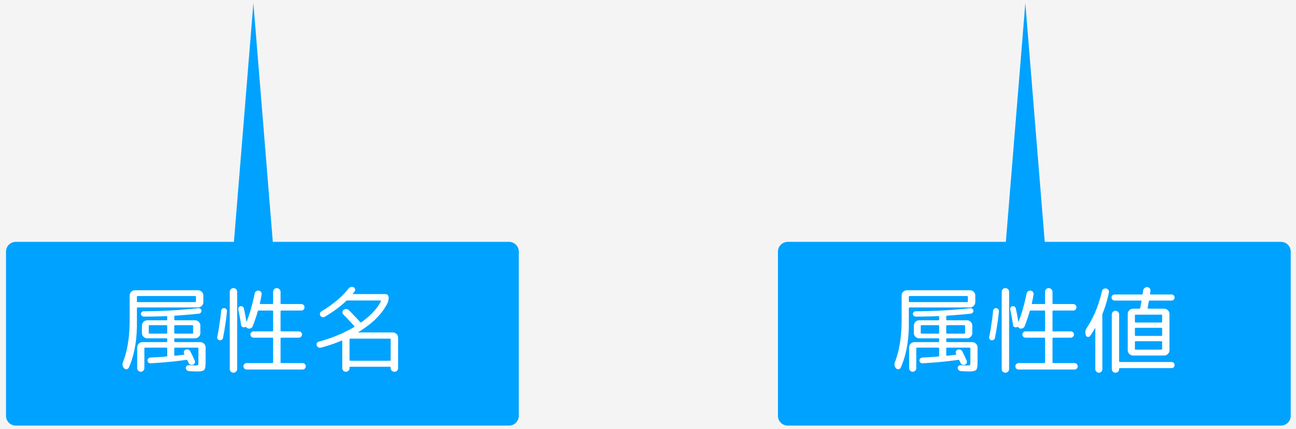
pixiv

<a **href**="?foo">




pixiv

<a href="?foo">






pixiv

<a href="?foo">




pixiv

<a href="?foo">

要素名



pixiv




pixiv

```
<a href="?foo">
   text
</a>
```

<a>要素



pixiv

# 今回はHTMLは本題ではないのでここまで

# ソースコードにも付加的な情報を付与したい！

# 付加的な情報…
# われわれは知っている

# Docコメント

```
/**
 * @param positive-int $id
 */
function find(int $id): Book
{
```

# コードに書きたい付加情報とは何か

- 人間に意図を伝えるためのドキュメント(コメント)
- 静的解析ツールが情報を読み取ってコードを検査するための情報
- フレームワークなどが情報を読み取って振る舞いを制御する
- プログラミング処理系(PHP)に特別な処理をさせるための指定

人間だけわかれば
ヨシ

人間もわかって
機械可読

実行時に可読で
効率高いとうれしい

構文の一部である
必要がある

# コードに書きたい付加情報とは何か

- 人間に意図を伝えるためのドキュメント(コメント)
- 静的解析ツールが情報を読み取ってエラーを検査するための情報
- フレームワークなどが情報を読み取って振る舞いを制御する
- プログラミング処理系(PHP)に特別な処理をさせるための指定

人間だけわかれば
ヨシ

人間もわかって
機械可読

実行時に可読で
効率高いとうれしい

構文の一部である
必要がある

**従来Docコメントが賄っていた領域**



pixiv

# DocCommentは自由帳

- 事実上、なんでも書ける
- なにを書いても怒られないが、@hogeのような形式が一般的
- phpDocumenterスタイルの「タグ」
  - @param int $foo（スペース区切り）
- Doctrineスタイルの「アノテーション」
  - @Annotation(”arg”, 1, 2, 3)



pixiv

# なぜDocコメント…

# 改めてDocコメント

# 実行時に読める不思議なコメント

- 関数やクラスなどの直前に /** ... */ のような形式でコメントを書くと、**Reflection**（リフレクション）という機能を使って読み取れる
  - Reflection::getComment() で文字列で取得できる
- 実行時とかにいい感じに情報を読み取って処理をしてやるとメタプログラミングが実現できる

# PHP Playground

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

< Request and Report

Version:

8.2

```php
<?php

class Foo
{
}

$ref = new ReflectionClass(Foo::class);
var_dump($ref->getDocComment());

```

```
bool(false)
```



pixiv

# メタ？

さっきも出てきた

# アトリビュートってなんだ？

- 日本語での定訳としては**「属性」**
- もっとも知られているであろう用語としてはHTMLタグ
- あるもの（要素）の性質を表す情報のこと
- MDNではHTMLの属性について次のように説明している

- 「属性は要素を拡張し、動作を変更したりメタデータを提供したりします。」
  Attribute（属性）- MDN Web Docs 用語集 より



pixiv

# Wikipedia日く

# メタ

23の言語版

ページ　ノート　　閲覧　ソースを編集　履歴表示　☆

出典: フリー百科事典『ウィキペディア（Wikipedia）』

ウィキペディアについての議論を行う「メタウィキ」についてはWikipedia:メタウィキメディアをご覧ください。

「**メタ**」のその他の用法については「メタ (曖昧さ回避)」をご覧ください。

**メタ**（*meta-*、 古希: μ ε τ ά-) とは、以下の意味を持つ接頭辞である：

**一般的な意味**

- 「あとに」という意味の古代ギリシャ語の接頭辞[1]。
- 転じて「超越した」、「高次の」という意味の接頭辞で[2]、ある学問や視点の外側にたって見る事を意味する[1]。
- 「変化」を意味する接頭辞。例えば metamorphose（変化）、metabolism（代謝）などで用いられる[3]。

**化学における意味**

- ベンゼン環の2置換体の構造異性体のうち、2つの置換基が炭素原子1つをはさんでいるものにつける接頭辞[2]。
- 同じ酸化物を水和して得られるオキソ酸の中で、水和度の低いものにつける接頭辞[2]。

https://ja.wikipedia.org/wiki/メタ
2022-12-14T19:41:26版より

## 「ある学問や視点の外側にたって見る」の用例 [ ソースを編集 ]

「ある学問や視点の外側にたって見る」という意味はアリストテレスの著書『メタピュシカ』（形而上学）に由来しており[1]、哲学では他にも「メタ倫理学」、「メタ哲学」等の用例がある。

数学基礎論や数理論理学では、数学や論理学を（人工）言語とみなした上で数学的手段を用いて研究としているので、研究対象たる「数学」や「論理学」のような「言語」と、それらの研究対象を研究するために用いる「数学」、「論理学」、「言語」を分けて考える必要があり、後者をそれぞれ「メタ数学」（英: *metamathematics*、超数学とも訳される）「メタ論理」、「メタ言語」と呼ぶ。またメタ数学やメタ言語等に関する概念には「メタ」という接頭辞を用い、例えばメタ数学の定理を研究対象の数学の定理と区別し、「メタ定理」（英: *metatheorem*）と呼ぶ。より一般に、これらの例のように何らかの理論（上では「数学」や「論理学」）を研究するための理論を一般に「メタ理論」と呼ぶ。

数理論理学をその始祖の一つとして持ち、人工言語たるプログラミング言語を研究対象の一つとして持つ計算機科学でも、「メタデータ」「メタプログラミング」「メタタグ」「メタヒューリスティクス」「メタ構文変数」「メタ検索エンジン」「メタファイル」「メタクラス」等の用例がある。

他にも「メタ認知」「メタ知識」「メタ記憶」「メタアナリシス」「メタデザイン」「メタモデル」「メタメッセージ」「メタ法価値論」「メタフィクション」等の用例がある。

https://ja.wikipedia.org/wiki/メタ
2022-12-14T19:41:26版より

# つまり？

# メタデータ
# データについてのデータ

# 例：
# 写真(画像データ)に対する「撮影場所」や「日時」などの情報

# 例：
# 楽曲(音声データ)に対する「演奏者」や「ジャンル」などの情報

# メタプログラミング

# プログラムについてのプログラミング

# メタプログラミングってつまり何

- プログラムそのものを対象にするプログラミング
  - クラスや関数などを動的に呼び出したり
  - ある処理をしたいとき、コードに直接的に書かなくても実現したりできる
  - 明示的に定義してなくてもなんか動いてたりするやつ
- フレームワークのようなコードは大なり小なりメタプロ的な要素がある
- 理窟がよくわからんけど動いてたりするのはこれ（魔法とか黒魔術（ブラックマジック）とか…）

# メタプロするなら
# コードから情報を
# 読み取れるとうれしい

(それが全てではない)

# ということで


php
Downloads
Documentation
Get Involved
Help
php 8.2
Search
Change language:
日本語
php 8
Released!
PHP 8.0 は、PHP 言語のメジャーアップデートです。
このアップデートには、たくさんの新機能や最適化が含まれています。 たとえば 名前付き引数、 union 型、アトリビュート、コンストラクタでのプロパティのプロモーション、match 式、nullsafe 演算子、JIT、型システムの改善、エラーハンドリング、一貫性の向上などです。
PHP 8 にアップデートしよう!

# Attribute クラス

(PHP 8)

## はじめに

アトリビュートを使うと、 コンピューターが解析できる構造化されたメタデータの情報を、 コードの宣言時に埋め込むことができます。 つまり、クラス、メソッド、関数、パラメータ、プロパティ、クラス定数にアトリビュートを指定することができます。 アトリビュートで定義されたメタデータは、 実行時にリフレクションAPIを使って調べることが出来ます。 よって、アトリビュートは、 コードに直接埋め込むことが出来る、 設定のための言語とみなすことができます。

# 基本的な定義方法

```
use Attribute;

#[Attribute(
    Attribute::TARGET_CLASS
  | Attribute::IS_REPEATABLE
)]
class MyAttr {
```

# 基本的な使いかた

```
use MyAttr;

#[MyAttr('foo', 'bar', 'buz')]
class Book
{
```

# 名前付きの呼び出しも可能

```
use MyAttr;

#[MyAttr(a: 'foo', b: 'bar', c: 'buz')]
class Book
{
```

# Attributeはどこに書けるの？

- 現在Reflectionから読み取れるのは6箇所
  - クラス
  - 関数
  - メソッド
  - プロパティ
  - パラメータ（仮引数）

宣伝

# これでみなさまはAttributeを定義できるようになりました

# ね、簡単でしょ？

# Docコメントと何が違うの？

# 比べてみよう

# 実行時に読める不思議なコメント

- 関数やクラスなどの直前に /** ... */ のような形式でコメントを書くと、**Reflection**(リフレクション)という機能を使って読み取れる
  - Reflection::getComment() で文字列で取得できる
- 実行時とかにいい感じに情報を読み取って処理をしてやるとメタプログラミングが実現できる

# 実行時に読める不思議なXXXX

- 関数やクラスなどの直前に #[Attr()] のような形式のコードを書くと、**Reflection**(リフレクション)という機能を使って読み取れる
  - Reflection::getAttributes() でReflectionAttributeを取得できる
- 実行時とかにいい感じに情報を読み取って処理をしてやるとメタプログラミングが実現できる



pixiv

# Q. 何が違うの？

# A. 違うよ、全然違うよ

# 実際に見てみましょう

# PHP Playground

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

< Request and Report

Version: 8.2

```
12  #[MyAttr("foo", "bar", "buz")]
13  class Foo
14  {}
15
16  $ref = new ReflectionClass(Foo::class);
17  foreach ($ref->getAttributes() as $attr) {
18  }
19
```

# ReflectionAttribute

- getAttributes() メソッドでとれるクラス
  - $ref = new ReflectionClass(Target::class);
  - $ref->getArguments() で値だけ取れる
  - $ref->newInstance() でアトリビュートをインスタンス化できる

# 何が変わったのか

- 明示的にReflectionしないと何も起こらないのはDocコメントと同じ
  - Attributeを書くだけなら、ただのコメントと同じ
- 引数の区切りがとても簡単
  - Docコメントでは自前で構文解析しないといけない
- 静的解析ツールが自然にサポートしてくれている
  - Docコメントでは変な記述をしても気付きにくい

# これでAttributeの機能は全部です！

# 完全マスター 

# 機能はめっちゃ
# シンプル

# あとはどういう「想い」を載せるか

# 創意工夫が大事

# とはいえ定石はある



pixiv

fortee

# PHPのDI、attributesとこれから

by 山岡広幸 / @hiro_y


採択 2021/03/28 14:50〜 Track A レギュラートーク（20分）

**PHPのDI、attributesとこれから** PHPerKaigi 2021

☆ 18 ビデオ スライド

山岡広幸 hiro_y

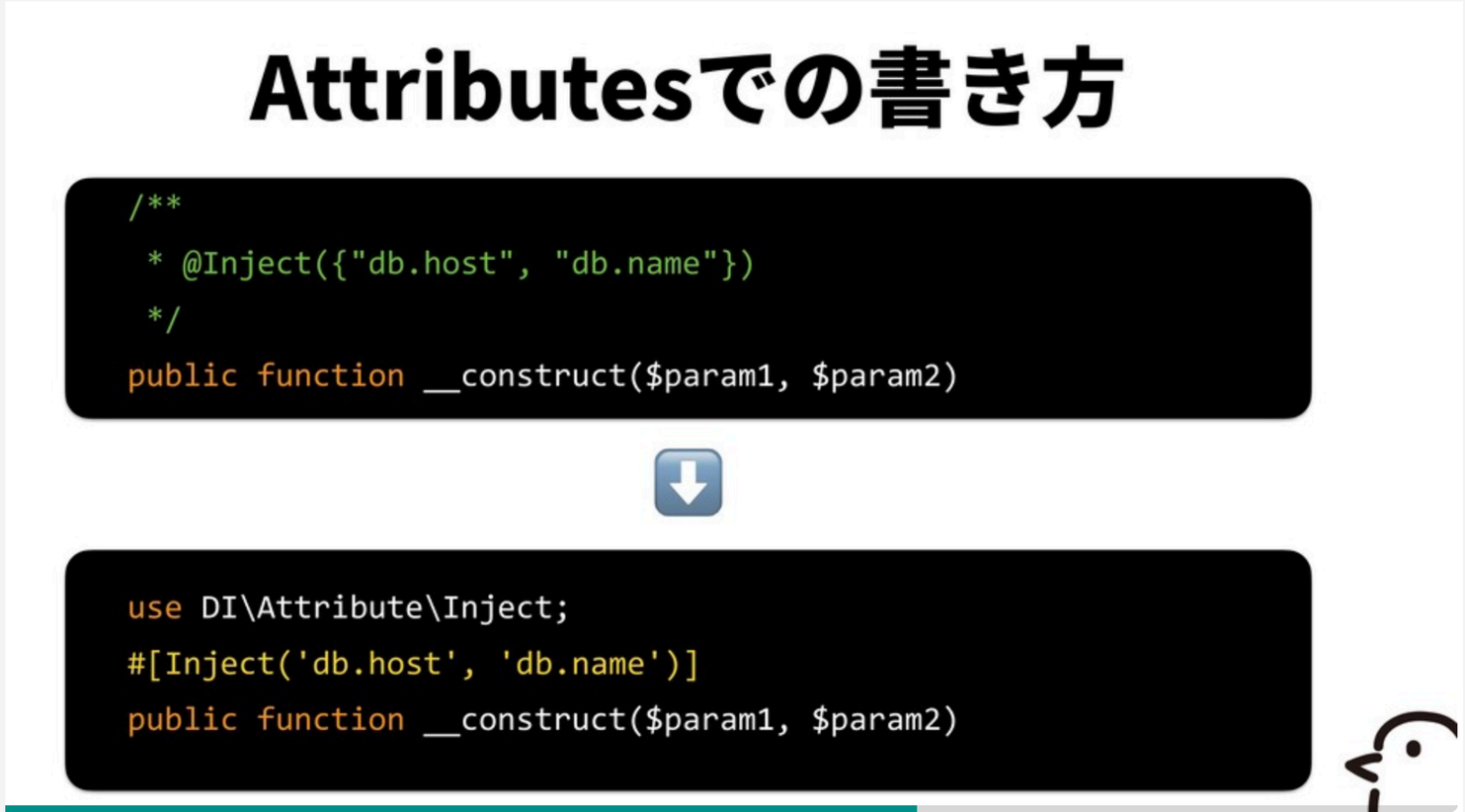
PHP 8で導入されたattributes。
今まで（やむをえず）PHPDocを用いて実装されてきた様々な書き方が、だんだん置き換わっていこうとしています。
今回は、DIの代表的なライブラリであるPHP-DIが新しいバージョン（7、2021年1月19日時点では未リリース）でどう変わろうとしているか、また、Symfonyなどの変化もあわせて紹介しつつ、PHPにおけるDIの行方について話したいと思います。



pixiv

# DI vs Attribute

- 山岡さんの2021年の発表で紹介された事例
- 依存性注入したいデータを明示するために使う



fortee

# Attributesを使った
# メタプログラミング入門！

by おかしょい / 岡田 正平 / @okashoi

#[?] PHPerKaigi 2022

採択 2022/04/11 17:45〜 Track A LT（5分）

**Attributesを使ったメタプログラミング入門！** PHPerKaigi 2022

☆ 10 スライド

おかしょい / 岡田 正平 okashoi

PHP8.0から追加された機能、Attributes。
これを使ったメタプログラミングの例として、クラスのプロパティにAttributesを付与することで



pixiv

# DB vs Attribute

- おかしょいさんの2022年の発表で紹介された事例
- データベースとPHPのクラスのプロパティを紐付けるために使う

できあがったものがこちら


```php
class User implements JsonSerializable
{
    use CanConvertFromDbRow, CanConvertToJson;

    #[DB('id'), JSON('id')]
    public int $id;

    #[DB('name'), JSON('name')]
    public string $name;

    #[DB('rate'), JSON('rate', JSON::OPTION_OMIT_EMPTY)]
    public float $rate;

    #[DB('active_flag'), JSON('isActive')]
    public bool $isActive;
}
```

# バリデーション vs Attribute

- 2022年11月に私が書いた記事
- アトリビュートは自然に発火しないので明示的に呼び出す必要がある

## Attributesを設計してみる

こんな感じで使えるようにしてみましょう。

```
class Address
{
    public function __construct(
        #[Length(min: 1, max: 100)]
        public string $city,
        #[RegExp(pattern: '/\A\d{3}-\d{4}\z/')]
        public string $postal,
    ) {
        validate_properties($this);
    }
}
```

PHP Advent Calendar 2022 1日目

@tadsan

投稿日 2022年11月30日　更新日 2022年12月01日

## Attributesで実現するPHP8時代のバリデータ

PHP

メリークリスマス！ みなさまに愛されたPHP7.x系は2022年11月28日をもってEOLを迎えました 🎉

さて、標記のAttribute(アトリビュート)とは、PHP 8.0で追加された機能です。

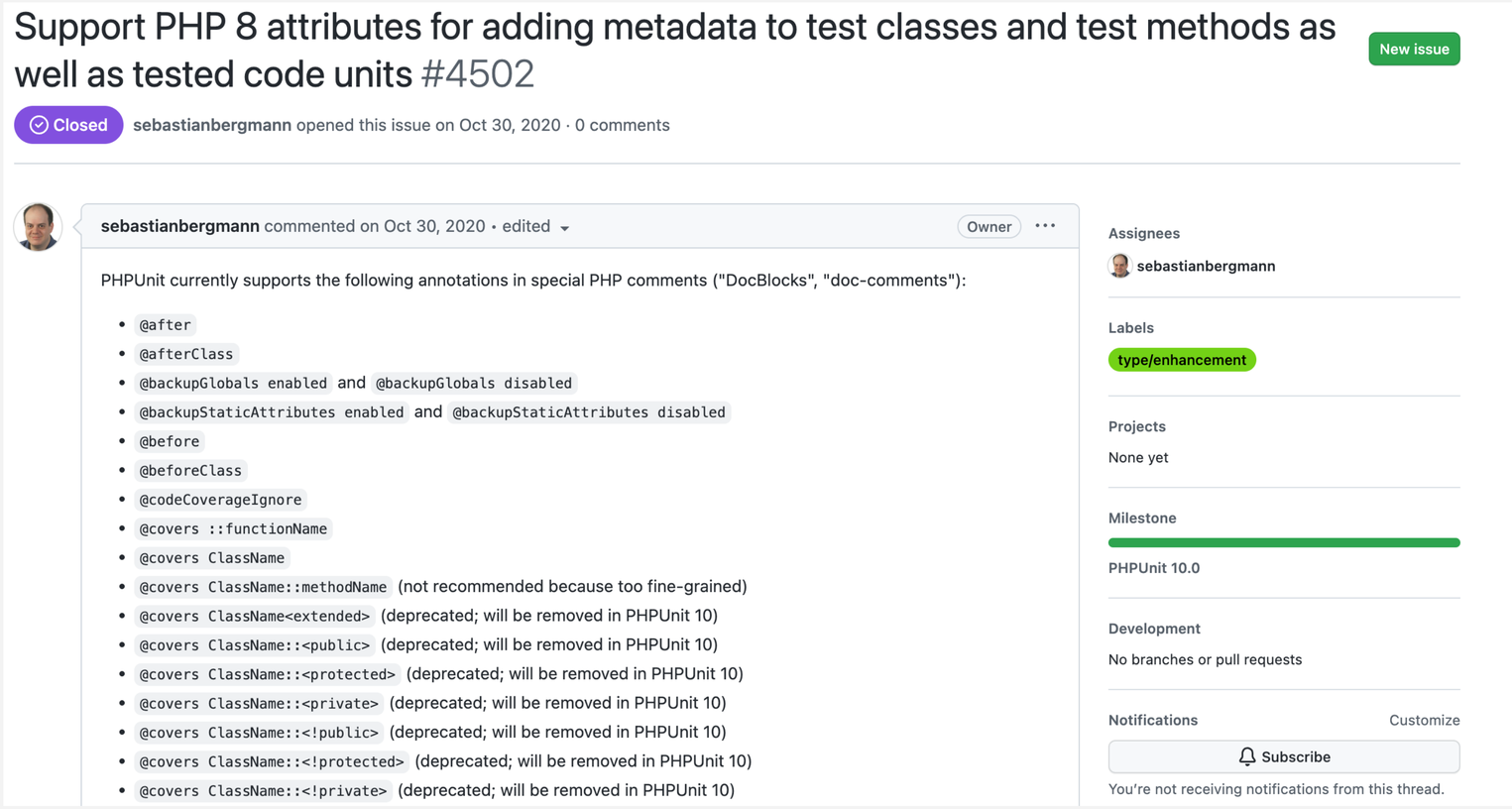
```
#[ほにゃらら(なんか: "書ける")]
class クラス{
    #[ほにゃらら(ここにも: "書ける")]
    public string $string;
}
```

# PHPUnit vs Docコメント

- PHPUnitはこれまでDocコメントによってテストの実行を制御できた
  - @dataProvider, @testWith … パラメタライズドテスト
  - @runInSeparateProcess, @runTestsInSeparateProcess … プロセスを隔離して実行
  - @covers … カバレッジターゲットの制御
  - @group … テストグループの宣言
- Doctrine式のAnnotationではなくスペース区切りが多かった

# PHPUnit 10 vs Attribute

- PHPUnit 10でAttributeに移行、11で非推奨化、12で削除予定
- 超大量の移行リスト… sebastianbergmann/phpunit#4502
- その他の変更は02氏の発表を参照のこと

# 明示 vs 黙示

# 「こっそりやる」vs「堂々とやる」

- PHPにはマクロやプリプロセッサがないので、明示的が基本ではある
- しかし、クラスローダーを悪用してソースコードを書き換えてしまう荒技も
  - Go! AOPなどのAOPフレームワークはクラスローダーでフックする

# PHPの後方互換性 vs Attribute

- PHPでは継承するときに型宣言を厳密に合わせないといけない
- PHP 8ではPHPの組み込みクラスにも型宣言が追加された
- そうするとPHP 7/8両対応のライブラリで非常に困る
- #[ReturnTypeWillChange] と書くと局所的に無視できるようになった
  - 問題の先送りができてうれしい！

# ひっそり佇むAttribute

- そこにあるだけでは何も起こさないというのは重要な性質
- # はPHPのコメント行なので、PHP 7以下のコードに書いても問題ない
- ということは… PHP 7以下でもパフォーマンスを気にしなければAttributeを使える
  - spiral/attributes というライブラリがある
  - 開発用途やデプロイスクリプトなどで使う分には問題にならない

# doctrine/annotations

- Docコメントのアノテーションを読み取るライブラリ
- Attributeとだいたい同じ使い方できるが仕様が微妙に異なる
  - デフォルトでは引数は __construct(array $values) として
    一個の配列にまとめて渡されてしまう
  - クラスに @Annotation @namedArgumentConstructor と書くと
    Attributeと同様に個別のパラメータで受け取れるようになる

# spiral/attributesの互換性

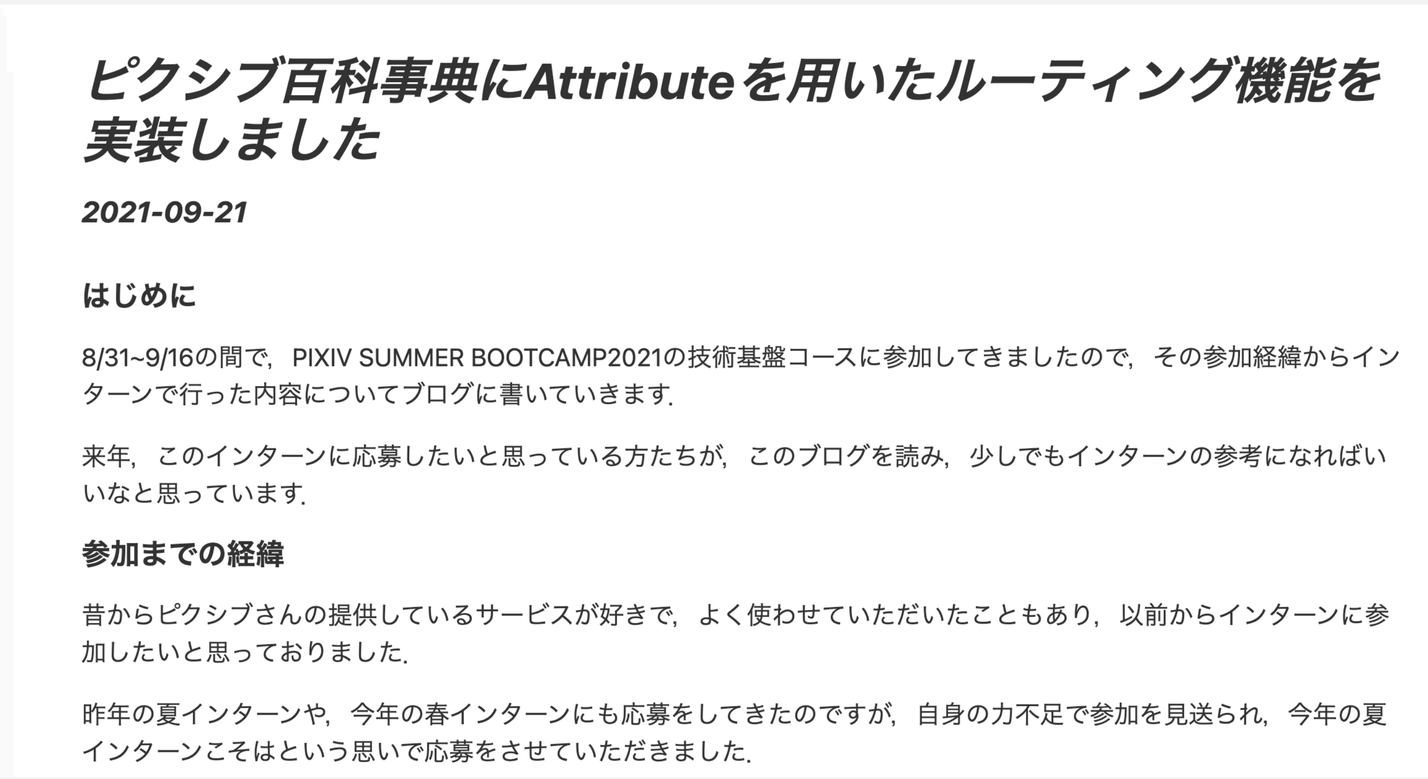
- PHP 7で利用できるAttributeを読み取るライブラリ
  - AnnotationとAttributeを相互運用できる
    - doctorine/annotationsのラッパーとしてPHPDocも読める
  - Annotationとの相互運用のためか、デフォルトでは引数がまとめて __construct(array $values) に渡されてしまう
    - クラスに #[Spiral\Attributes\NamedArgumentConstructor]と書くと通常のAttributeと同じ仕様で使える



pixiv

# ピクシブ百科事典 vs Attribute

- コントローラにルーティング情報が書ける
- これは実行時に読まず、スクリプトでファイルに書き出す方式

## ピクシブ百科事典にAttributeを用いたルーティング機能を実装しました

2021-09-21

### はじめに

8/31~9/16の間で，PIXIV SUMMER BOOTCAMP2021の技術基盤コースに参加してきましたので，その参加経緯からインターンで行った内容についてブログに書いていきます.

来年，このインターンに応募したいと思っている方たちが，このブログを読み，少しでもインターンの参考になればいいなと思っています.

### 参加までの経緯

昔からピクシブさんの提供しているサービスが好きで，よく使わせていただいたこともあり，以前からインターンに参加したいと思っておりました.

昨年の夏インターンや，今年の春インターンにも応募をしてきたのですが，自身の力不足で参加を見送られ，今年の夏インターンこそはという思いで応募をさせていただきました.

# キミだけの最高のユースケースを見付けだせ