はじめてのPHPDocと型

My first PHPDoc and Type

2016-09-09 ピクシブ社内勉強会

型とは何か

型… 型…

- 一言で誤解を恐れずに表現すると
  プログラミング言語が
  1 + [] (数字と配列のたしざん)
  みたいなのを叱れるようにする基盤

- コンパイルするときに型を付けるか
  動かすとき(runtime)に調べるかが、
  静的型付きと動的型検査の大きな差

動的型検査

- プログラムを動かしながら
  手さぐりで型を調べる
- 「$aの型はintだな、$bの型はstring…
  だけど数字だから足しざんできるね！」
  みたいな茶番を毎回やってる

静的検査

動かす前に把握する

- 実際に動かさなくても人間は整合性をとりながらPHPを書ける
- 人間がわかるってことは、機械にもやってれないことはない

型を推測してみよう（型推論）

```
// 絶対int返すマン
function x() { return 1; }
// 絶対array返すマン
function y() { return []; }
// たしざん
// $n1, $n2 は数字っぽい値が欲しい
function add($n1, $n2) { return $n1 + $2; }

$a = x(); // x()がintを返すからたぶんintのはず
$b = y(); // y()がarrayを返すからたぶんarrayのはず

$c = add($a, $b);
// 数字っぽい値以外を渡すとエラーが発生することは
// 理窟の上では動かさなくても明らか
```

型がない関数の問題

- 引数に何渡せばいいの… 何返ってくるの
  - 時間は有限なので、既存コードを読む時間は最小限にできるとウレシイ
- ほんとに「見ればわかる」シンプルさならいいけど、現実は甘くない
  - 型が特定できないことも多い

適当なコメント（Before）

```
// 絶対int返すマン
function x() { return 1; }
// 絶対array返すマン
function y() { return []; }
// たしざん
// $n1, $n2 は数字っぽい値が欲しい
function add($n1, $n2) { return $n1 + $2; }

$a = x(); // x()がintを返すからたぶんintのはず
$b = y(); // y()がarrayを返すからたぶんarrayのはず

$c = add($a, $b);
// 数字っぽい値以外を渡すとエラーが発生することは
// 理窟の上では動かさなくても明らか
```

PHPDocで型を付ける（After）

```php
/** @return int */
function x() { return 1; }
/** @return array */
function y() { return []; }

/**
 * @param int $n1
 * @param int $n2
 */
function add($n1, $n2) { return $n1 + $2; }
```

PHPDoc

DocComment

- PHPのリフレクションで実行時に
  取得できる特別な形式のコメント

- メソッドとかクラスの定義に
  /** ～ */ って書くと文字列になる

- 開始記号は必ず /** にすること
  (/* では無効なので注意！！！！)

DocCommentを取得する

```php
<?php

/** こんにちは〜〜〜〜 */
function hello() { throw new \Exception(); }

$ref = new \ReflectionFunction('hello');
echo $ref->getDocComment(), PHP_EOL;
exit;
```

APIドキュメントを生成する

- ソースコード中のコメントを基に、HTMLドキュメントを生成できる
- JavaDocにインスパイヤされたPHPDocって形式がある
- phpDocumentorプロジェクトがデファクト https://phpdoc.org/docs/latest/index.html

PHPDocの型

- 基本はPHPの型かクラス名を書く
- 複合型
  - 「intまたは文字列」 `int|string`
- 値が並んだ配列
  - 「intが並んだ配列」 `int[]`
- https://zonuexe.github.io/phpDocumentor2-ja/references/phpdoc/types.html

これだけ覚えて帰ってね

タグ名	意味	例
@param	引数を定義	@param int $n1
@return	返り値を定義	@return int[]
@var	変数/プロパティを定義	@var int
@property	マジックプロパティを定義	@property int $id

型: **array**について

- array と mixed[] は意味が同じなのでは
  - PHPのarrayは
    リストでもあり連想配列でもある
  - 標準化された用例ではないが、
    tadsanは連想配列をarrayと書く
  - 0はじまりの連番の配列は
    mixed[] とか User[] みたいに書く

標準化について

- 前述の通り、デファクトは phpDocumentor2の仕様 (phpdoc.org)
  - 日本語 https://zonuexe.github.io/phpDocumentor2-ja/
- 使ってるツールはいろいろある
  - PhpStorm, Phan, ApiGen, phpDox…
- PSR-5で提案されてるけど… どうなる？

PHPの型宣言

タイプヒント改め「型宣言」

- PHP5では「タイプヒント」と呼ばれ、引数に特定のキーワードかクラス名を書くことができた
  - `function getWorksByUser(UserModel $user)` みたいな
  - `array`, `callable`, `resource`
- PHP7では「型宣言」になって返り値の型が書ける

型宣言の難しさ

- PHP5はstringとかintとか書けなかった
  - そんなの簡単だろさっさと実装しろや…みたいな単純な問題ではない
- function hoge(int $n) を hoge("12") で呼び出したときにどうするの…?
  - パンドラの箱を開けてしまった感
- declare(strict_types=1) って書く

キャストの難しさ

```php
<?php declare(strict_types=0);

function myint(int $v) { return $v;}

myint(1.1);  // => 1
myint(true); // => 1
myint("1");  // => 1
myint(floatval(PHP_INT_MAX - 512)); // int(9223372036854774784)
myint(123456789012345678900.0);
// PHP Fatal error:  Uncaught TypeError: Argument 1
// passed to myint() must be of the type integer, float given
myint("123456789012345678900");
// PHP Fatal error:  Uncaught TypeError: Argument 1
// passed to myint() must be of the type integer, string given
```

Phan

etsy/phan（次回豫告）

- 静的解析ツール
  - PHPDocの定義を手がかりに、
    型が違ってるところを指摘してくれる
  - 11/3にPHPカンファレンスで
    詳しくしゃべってきます（30分枠）