

# PHP & syntax highlighting

PHP Editor and IDEs

# ！ お前誰よ



- うさみけんた (@tadsan) / Zonu.EXE
  - GitHub/Packagistでは id:zonuexe
- ピクシブ株式会社でひたすらPHPやってる p(ixi)v
- フレームワークなき道にフレームを作るおしごと
- 最近やってること
  - 会社にPhpStormを導入しました
  - Emacsのphp-modeとかの改善やってます

いままでの復習

# なんとかStormに負けたくない Emacs 初級篇

2016-03-01 PHP BLT #3

# 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

# Phanによる PHPコード静的解析

公開補訂版

ピクシブ株式会社  
うさみけんた @tadsan

pixiv

JAPAN phpcon 2016, 3rd November

要点

# | 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>



ところで、PHPの開発環境  
(エディタ/IDE)はいろいろある

# みなさんは何を使っていますか？

---

- vi? Emacs? Vim? Spacemacs? NeoVim?
- Sublime Text? Atom? VisualStudio Code?
- PhpStorm? IntelliJ IDEA?
- NetBeans? Eclipse?
- nano? micro? 秀丸? TeraPad?

Syntax highlighting!

# 各種エディタでPHPを強調表示するには

---

- Emacs → php-mode 入れる
- Vim → StanAngeloff/php.vim 入れる
- Eclipse, NetBeans  
→ PHPが入ってるエディションを入れる
- IntelliJ IDEA → PHPパッケージ入れる

各種最新版に  
アップデートすべし

(PHP BLTで毎回言ってる気がする…)

```
namespace Baguette\Response;

use GuzzleHttp\Psr7\Stream;
use Psr\Http\Message\StreamInterface;
use Teto\HTTP;

/**
 * Interface of Response classes
 *
 * @author USAMI Kenta <tadsan@zonu.me>
 * @copyright 2016 Baguette HQ
 * @license http://www.apache.org/licenses/LICENSE-2.0 Apache-2.0
 */
final class RawResponse implements ResponseInterface
{
    /** @var Stream */
    public $content;
    /** @var string */
    public $content_type;
    /** @var int */
    public $status_code;
}
```

```
namespace Baguette\Response;

use GuzzleHttp\Psr7\Stream;
use Psr\Http\Message\StreamInterface;
use Teto\HTTP;

/**
 * Interface of Response classes
 *
 * @author USAMI Kenta <tadsan@zonu.me>
 * @copyright 2016 Baguette HQ
 * @license http://www.apache.org/licenses/LICENSE-2.0 Apache-2.0
 */
final class RawResponse implements ResponseInterface
{
    /** @var Stream */
    public $content;
    /** @var string */
    public $content_type;
    /** @var int */
    public $status_code;
}
```

```
namespace Baguette\Response;

use GuzzleHttp\Psr7\Stream;
use Psr\Http\Message\StreamInterface;
use Teto\HTTP;

/**
 * Interface of Response classes
 *
 * @author    USAMI Kenta <tadsan@zonu.me>
 * @copyright 2016 Baguette HQ
 * @license   http://www.apache.org/licenses/LICENSE-2.0 Apache-2.0
 */
final class RawResponse implements ResponseInterface
{
    /** @var Stream */
    public $content;
    /** @var string */
    public $content_type;
    /** @var int */
    public $status_code;
}
```



```
use GuzzleHttp\Psr7\Stream;
use Psr\Http\Message\StreamInterface;
use Teto\HTTP;

/**
 * Interface of Response classes
 *
 * @author    USAMI Kenta <tadsan@zonu.me>
 * @copyright 2016 Baguette HQ
 * @license   http://www.apache.org/licenses/LICENSE-2.0 Apache-2.0
 */
final class RawResponse implements ResponseInterface
{
    /** @var Stream */
    public $content;
    /** @var string */
    public $content_type;
    /** @var int */
    public $status_code;
}
```

```
namespace Baguette\Response;

use GuzzleHttp\Psr7\Stream;
use Psr\Http\Message\StreamInterface;
use Teto\HTTP;

/**
 * Interface of Response classes
 *
 * @author USAMI Kenta <tadsan@zonu.me>
 * @copyright 2016 Baguette HQ
 * @license http://www.apache.org/licenses/LICENSE-2.0 Apache-2.0
 */
final class RawResponse implements ResponseInterface
{
    /** @var Stream */
    public $content;
    /** @var string */
    public $content_type;
    /** @var int */
    public $status_code;
}
```

```
namespace Baguette\Response;

use GuzzleHttp\Psr7\Stream;
use Psr\Http\Message\StreamInterface;
use Teto\HTTP;

/**
 * Interface of Response classes
 *
 * @author    USAMI Kenta <tadsan@zonu.me>
 * @copyright 2016 Baguette HQ
 * @license   http://www.apache.org/licenses/LICENSE-2.0 Apache-2.0
 */
final class RawResponse implements ResponseInterface
{
    /** @var Stream */
    public $content;
    /** @var string */
    public $content_type;
    /** @var int */
    public $status_code;
}
```

悲しみ

だいたいPHPDocを  
無視しやがる

Emacsのphp-modeなら  
手を出せるので直した

```
namespace Baguette\Response;

use GuzzleHttp\Psr7\Stream;
use Psr\Http\Message\StreamInterface;
use Teto\HTTP;

/**
 * Interface of Response classes
 *
 * @author USAMI Kenta <tadsan@zonu.me>
 * @copyright 2016 Baguette HQ
 * @license http://www.apache.org/licenses/LICENSE-2.0 Apache-2.0
 */
final class RawResponse implements ResponseInterface
{
    /** @var Stream */
    public $content;
    /** @var string */
    public $content_type;
    /** @var int */
    public $status_code;
}
```

# ありがちなミスを防ぐ工夫

```
/**
 * てきとーに書ける。文章中に int って書いても強調される
 *
 * @param string|boolean[] $a
 * @param ArrayObject|Hoge[] $b 変数も強調表示される
 * @param str|boolean $c 間違った型名には色はつかない
 * @params ArrayObject $d @params だと色は付かない (@paramが正しい)
 */
public function __construct($a, $b, $c, $d)
{
    /* ふつうのブロックコメントは別の色になる */
}
```



(スライドここまで)