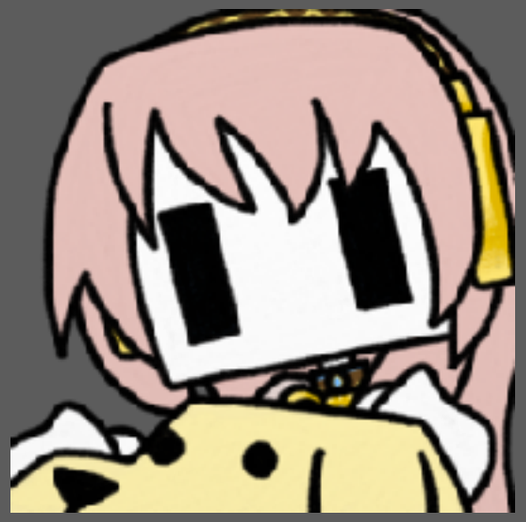
プログラミング環境としての php

Introducing PHP as a Programming Environment

pixiv

2017-08-19 NEEC Kamata #opendevcon
Open Developers Conference 2017 Tokyo

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
- Qiitaに記事を書いたり変なコメントしてるよ
- 最近やってること
  - 万難が排されつつあるのでPHP7 移行
  - えいやえいやとテスト書いたり直したりしてる

pixiv

最初に
覚えてほしいこと

現在の安定版は

7.1系です

質問です

あなたの想像する

PHPコードはどれ？

A

```
<!DOCTYPE html>
<html><head><title>Hello, PHP!</title></head>
<p>
<?php if (date('H') < 19): ?>
    こんにちは
<?php else: ?>
    こんばんは
<?php endif; ?>
</p>
<p>今は<?= htmlspecialchars(date('H')) ?>時です</p>
```

B

```
<?php require 'bootstrap.php';
$db = mysqli_connect();
$stmt = mysqli_prepare($db, 'SELECT * FROM books');
mysqli_stmt_execute($stmt);
print '<h1>本の一覧</h1>';
print '<ul>';
foreach ($stmt as $s) {
    print '<li>'; print $s->name; print '</li>';
}
print '</ul>';
```

C

```
<?php
namespace Hoge\Http\Controller;
class BooksController extends BaseController {
    public function index() {
        $books = Books::getAll();

        $this->render(compact("books"));
    }
}
```

いろいろある

よくあるPHPのコード分類

- A: HTMLの条件分岐と値の出力
- B: DBの内容をHTMLとして出力
- C: クラス定義だけ書かれたコード

よくあるPHPのコード分類

- A: HTMLの条件分岐と値の出力
  ↑ SSI的PHP
- B: DBの内容をHTMLとして出力
  ↑ CGI的PHP
- C: クラス定義だけ書かれたコード
  ↑ フレームワーク的PHP

割とどうとでも

書けるのがPHP

PHPの実行手順のイメージ

- コードを書く（or 既製コードを準備）
- レンタルサーバーを借ります
- (S)FTPで.phpファイルを転送します
- デプロイ完了 ＼(^o^)／

PHPで書くとURLに

.phpって付くのでは??

そんなことはない

（そんなこともある）

ディレクトリ配置のイメージ

これをな

```
php-project/
├── inc
│   └── bootstrap.php
└── public（ドキュメントルート）
    ├── index.php
    ├── login.php
    ├── logout.php
    └── terms.html
```

ディレクトリ配置のイメージ

こうじゃ

```
php-project/
├── inc
│   ├── bootstrap.php
│   └── routes.php
└── public（ドキュメントルート）
    ├── .htaccess（Apacheの場合）
    └── index.php
```

このPHPを動かすに
はどうすればいいの

PHPの動作方法にはいろいろある

- Server API (SAPI) と呼ばれる
  - コマンドライン (cli, cli-server)
  - Apache+mod_php (apache)
  - CGI (cgi-fcgi)
  - PHP-FPM (fpm-fcgi)

どこでPHPを動かせばいいか

- 何も考えずにレンタルサーバ借りて、ファイルを適当に転送すれば動く
- HerokuとかPaaSでも簡単に動く
- オンプレとかVPSとかでアプリケーションサーバーを実行する方法もある

テストするだけなら

PHPだけあればok

```
php -S
localhost:8989 index.php
```

ビルトインサーバー

（php -S）PHP5.4以降

開発用の簡易HTTP鯖

よくある.htaccess（Apache HTTPd）

```
# ファイルがあったら、それを返す(.phpなら実行)
# 存在しなかったら、index.phpを実行
RewriteCond %{REQUEST_FILENAME} !-f
RewriteRule ^ index.php [QSA,L]
```

愚直な routes.php

```
<?php
$path = substr(explode('?',  $_SERVER['REQUEST_URI'], 2)[0], 1);
$is_post = $_SERVER['REQUEST_METHOD'] === 'POST';

if ($is_get && $path === '/') {
    show_index(); return;
} elseif ($path === '/login') {
    $is_post ? post_login() : show_login(); return;
}

http_response_code(404);
echo "ないよー";
```

これを毎回自分で書くのは
ちょっぴり大変なので、
フレームワークを利用する

PHP

micro framework

Slim, Lumen など

シンプルなルーティングがしたかった

https://qiita.zonu.me/simple-routing

そろそろ言語の話

> PHPには「protected属性」も「仮想メソッド」もありますよ。情報学科の教官が「重要だ」っていうやつは何でもね。僕自身は、こんなものどうでもいいと思ってますけど。

–Rasmus Lerdorf

伝説のPHP作者「Rasmus Lerdorf」名言集を聞くと嫌PHP厨がファビョる

https://anond.hatelabo.jp/20100427231539 の訳を抜萃

実際ふつうの言語にある機能はだいたいある

Linux, *BSD,
macOS, Windows

RubyやPythonで書けること
は概ねできると思ってよい
（異常に苦手なことはある）

Cっぽさ、Perlっぽさ、
そこはかとないJavaっぽさ
（結果としてどれでもない雑然とした感じ）

Cっぽいコード

```php
<?php

$fp = fopen('input.txt', 'r');
$i = 0;
while (($line = fgets($fp, 4096)) !== false) {
    printf("%d: %s", ++$i, $line);
}
```

Perlっぽいコード

```php
<?php

$con = mysqli_connect() or die(1);
# ただし、現在は例外を利用すべき
# テストもしにくくなるので非推奨
```

Javaっぽいコード

```php
<?php
abstract class Foo {
     abstract function sample($n);
}
final class Bar extends Foo implements Hoge {
     public function sample($n) {
          if (!is_numeric($n)) {
               throw new InvalidArgumentException("errorMsg");
          }
     }
}

$bar = new Bar(); $bar->sampleCode(128);
```

PHPの基本構文

データ型・リテラル

```php
<?php
// 全ての変数は $ から始まる
$string = 'string' . "value";
$array  = [1, 2, 3];
$bool   = true;
$int    = 56789;
$float  = 1.234;
$obj1   = new stdClass;
$obj2   = (object)["name" => "Miku", "age" => 16];
```

関数定義

```php
<?php
namespace zonuexe;

function triangle_area(float $h, float $b) :float {
    return $h * $b / 2;
}

function triangle_area($h, $b) {
    return $h * $b / 2;
}
```

関数・クラス定義の注意

- RubyやPythonでは関数(メソッド)を再定義したり、定義済みクラスに定義を追加できる(オープンクラス、モンキーパッチング)
- その代り特定のクラスが複数ファイルに分割されることがないのでわかりやすい

クロージャ

```php
<?php

$h = function($s) {
    return htmlspecialchars($s, ENT_QUOTES, 'UTF-8');
};

$array = array_map($h, getNames());
```

PHPの良い言語機能

array

```php
<?php
$old_num = array(1, 2, 3);
$fruits = ["りんご", "ばなな", "みかん"];
```

```php
$tadsan = [
    "name" => "めぐりねるか",
    "age"  => 28,
    "from" => "北海道",
];
```

foreach

```php
<?php
$fruits = ["りんご", "ばなな", "みかん"];

for ($i = 0; $i < count($fruits); $i++) {
    printf("%s\n", $fruits[$i]);
}

foreach ($fruits as $fruit) {
    printf("%s\n", $fruit);
}
```

ジェネレータ

```php
<?php
function fizzbuzz ($to) {
    for ($i = 1; $i <= $to; $i++) {
        $s = "";
        if ($i % 3 == 0) { $s .= "Fizz"; }
        if ($i % 5 == 0) { $s .= "Buzz"; }
        yield ($s === "") ? $i : $s;
    }
}
```

ジェネレータ + foreach

```php
// フィボナッチ数列を順に出力する
foreach (fizzbuzz(100) as $f) {
    echo $f, PHP_EOL;
}
```

// これを応用すれば、函数型プログラミングの
// 遅延リスト(無限リスト)を簡単に実装できる！

callable（文字列）

```php
<?php

$func= "strlen";
echo $func("abcde"); //=> 5
```

```php
$ary = ["a", "bc", "desc", "hello"];
$bry = array_map("strlen", $ary);
//=> [1, 2, 3, 4, 5]
```

callable（そのほか）

```php
<?php

$dt = new DateTime;
$method = [$dt, "getTimeStamp"];

echo $method();
// 1503115455
echo $dt->getTimeStamp();
// 1503115455
```

Reflection

```php
<?php

function sampleFnc($a, $b, $c = null){}

$ref = new ReflectionFunction("sampleFnc");
echo $ref->getNumberOfParameters();
//=> 3
echo $ref->getNumberOfRequiredParameters();
//=> 2
```

Webプログラミングに適した機能

実はPHPのWebはCGIの世界観

HTMLを print したりすると
出力される

テンプレート

```
<?php
// HTTPヘッダはボディよりも先に出力する
header("Content-Type: text/plain;");
echo "apple\n";
?>
orange
banana
<?= getMelon() ?>
<?php function getMelon(){ return "melon"; }
```

実は ?>ほげ<?php は

echo "ほげ"; と同じ

テンプレート

```
<?php $n = trim(fgets(fopen('php://stdin', 'r')));

for ($i = 1; $i <= $n; $i++) {
    if ($i % 3 === 0) { ?>Fizz<?php }
    if ($i % 5 === 0) { ?>Buzz<?php }
    elseif ($i % 3 !== 0) echo $i;
?>

<?php }
```

出力バッファリング

```php
<?php

ob_start();
var_dump(new DateTime);
$str = ob_get_clean();

var_dump($str);
```

$_SERVER

```php
<?php
// http://php.net/$_SERVER を読む

// ユーザーのリクエスト情報
echo htmlspecialchars($_SERVER['REQUEST_METHOD']);
echo htmlspecialchars($_SERVER['REQUEST_URI']);
echo htmlspecialchars($_SERVER['HTTP_ACCEPT_LANGUAGE']);
```

HTTPのための機能

```
<?php
$img = base64_decode('R0lGODlhAQABAIABAP///
wAAACH5BAEKAAEALAAAAAABAAEAAAICTAEAOw==');
http_response_code(404);
header("Content-Type: image/gif");
header("Content-Length:".strlen($img));

echo $img;
```

セッション機能

```php
<?php // logout.php
session_start();

$ls_logged_in = isset($_SESSION['user_id']);

if ($is_logged_in) {
    $_SESSION = [];
    session_regenerate_id();
}
```

まとめ

まとめ

- PHPは7.1系(2017年現在安定版)にしよう
- PHPは多様なスタイルがあるよ
- レンタルサーバーとかに適当に置けば
  まあなんとなく動くよ
- PHPはふつうの文法とWeb向きの
  機能があるプログラミング言語だよ