

PHP縛りゲーの世界

第136回 PHP勉強会@東京
2019年3月27日 #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
 - 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

待ってるによ



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

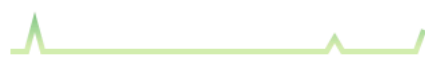
New

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



Top languages

Emacs Lisp PHP

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



Most used topics

Manage

emacs melpa php

major-mode

php-mode

A PHP mode for GNU Emacs



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

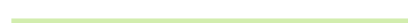
● Ruby ★ 6 🍴 1 Updated 6 days ago



helm-c-yasnippet

Helm source for yasnippet

● Emacs Lisp ★ 14 🍴 5 Updated on 21 Mar



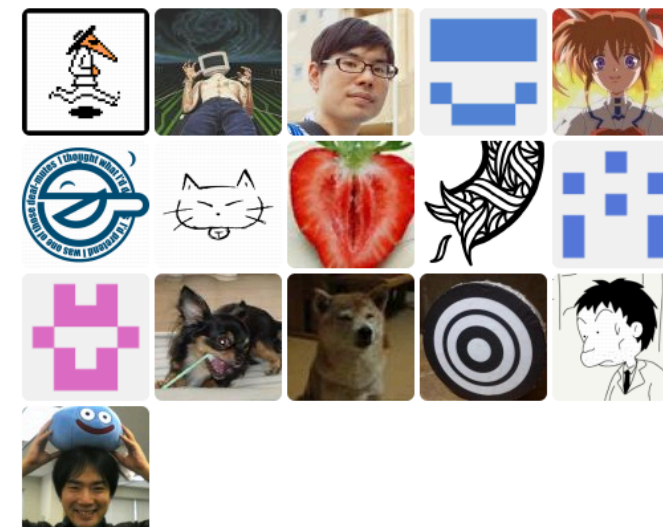
replace-colorthemes

Top languages

● Emacs Lisp ● Ruby ● TeX

People

37 >



宣伝

今週末は
PHPerKaigiです

CIとカ静的解析
の話をしてます

チケットを持って
ない人は今すぐ購
入

さて
みなさん

プログラミングは
好きですか？

プログラミングは
楽しいですか？

もっと楽しくし
たくないですか？

そうですね

制約を課すべし

“ゲームをプレイする際、本来ゲーム側からは設定されていない制限（縛り）を自ら科す事によって、より難易度の高いゲームをプレイする事。”

—ニコニコ大百科

<https://dic.nicovideo.jp/a/%E7%B8%9B%E3%82%8A%E3%83%97%E3%83%AC%E3%82%A4>

他言語における 実例

| セミコロレスJava

- 常識的なJavaは、式文の末尾にセミコロン(;)を要求
- ところが、工夫することで；の挿入を回避してコードを書くことが可能

| Pythonワンライナー

- Pythonはインデントベースの文法(オフサイドルール)を持つため、改行を要求する文法が多い
- ……が、さまざまなテクを駆使することでこの制約を回避

それならPHPは

| セミコロンスレスPHP

- PHPもJavaと同じく式文の末尾にセミコロン(;)を要求する
- だが実はPHPは ?> 直前の ; を省略できる
- ?> も禁止することで成立する

| ワンライナーPHP

- PHPは改行が文法的な意味を（ほぼ）持たず；で区切って複数の式文を一行に書ける
- Pythonと同等の難易度にはならない

どうしよう





\$

| PHPにおける \$ とは

- 言わずと知れた変数のシンボル(sigil)
- おそらくPerlに由来するが、Perlの意味とはまったく異なる
- 実際いろんなところで必須

| PHPにおける \$

```
<?php
```

```
function foo($a, $b) {  
    $name = "hoge"; // 変数代入  
    var_dump($_SERVER); // ← スーパーグローバル変数  
  
    foreach (piyo() as $p) { // ← 省略できない  
        echo $piyo->$name; // ← 可変プロパティ  
    }  
}
```

どうしよう

(2回め)

変数を利用できない＝
状態の更新ができない



大丈夫です

\$なしで意外と
書けます

FizzBuzzは
書けます

どうすんの

| PHPにおける \$

```
<?php
```

```
const FB = ['FizzBuzz', null, null, 'Fizz', null,  
'Buzz', 'Fizz', null, null, 'Fizz', 'Buzz', null,  
'Fizz', null, null];
```

```
call_user_func(new class { function __invoke() {  
    if (func_get_arg(0) - 1 )  
        call_user_func(__METHOD__, func_get_arg(0) - 1);  
    echo FB[func_get_arg(0) % 15] ?? func_get_arg(0), "\n";  
}}, 100);
```


| PHPにおける \$

```
<?php
```

```
const FB = ['FizzBuzz', null, null, 'Fizz', null,  
'Buzz', 'Fizz', null, null, 'Fizz', 'Buzz', null,  
'Fizz', null, null];
```

```
call_user_func(new class {  
    if (func_get_arg(0) - 1 )  
        call_user_func(__METHOD__, func_get_arg(0) - 1);  
    echo FB[func_get_arg(0) % 15] ?? func_get_arg(0), "\n";  
}}, 100);
```

定数定義 (\$不要)

PHPにおける \$

<?php

```
const FB = ['FizzBuzz', null, null, null, null,  
            'Buzz', 'Fizz', null, null, null, null,  
            'Fizz', null, null];
```

無名関数を起動

無名関数の起動

```
call_user_func(new class { function __invoke() {
    if (func_get_arg(0) > 0 )
        call_user_func(__METHOD__, func_get_arg(0) - 1);
    echo FB[func_get_arg(0) % 15] ?? func_get_arg(0), "\n";
}}, 100);
```

| PHPにおける \$

```
<?php
```

```
const FB = ['FizzBuzz', null, null, 'Fizz', null,  
'Buzz', 'Fizz', null, null, 'Fizz', 'Buzz', null,  
'Fizz', null, null];
```

FizzBuzzのロジック

```
call_user_func(new class { function __invoke() {  
    if (func_get_arg(0) - 1 )  
        call_user_func(__METHOD__, func_get_arg(0) - 1);  
    echo FB[func_get_arg(0) % 15] ?? func_get_arg(0), "\n";  
}}, 100);
```

| PHPにおける \$

```
<?php
const FB = ['FizzBuzz', null, null, 'Fizz', null,
'Buzz', 'Fizz', null, null, 'Buzz', null,
'Fizz', null, null];

call_user_func(new class { function __invoke() {
    if (func_get_arg(0) > 0 )
        call_user_func(__METHOD__, func_get_arg(0) - 1);
    echo FB[func_get_arg(0) % 15] ?? func_get_arg(0), "\n";
}}, 100);
```

100から0になるまで、
自分自身の関数を呼び出す

| PHPにおける \$

```
<?php
const FB = ['FizzBuzz', null, null, 'Fizz', null,
'Buzz', 'Fizz', null, null, 'Fizz', 'Buzz', null,
'Fizz', null, null];

call_user_func(new class {
    if (func_get_arg(0) - 1 )
        call_user_func(__METHOD__, func_get_arg(0) - 1);
    echo FB[func_get_arg(0) % 15] ?? func_get_arg(0), "\n";
}}, 100);
```

FizzBuzzをプリント

| PHPにおける \$

```
<?php
```

```
const FB = ['FizzBuzz', null, null, 'Fizz', null,  
'Buzz', 'Fizz', null, null, 'Fizz', 'Buzz', null,  
'Fizz', null];
```

引数をとる

```
call_user_func(new class { function __invoke() {  
    if (func_get_arg(0) - 1 )  
        call_user_func(__METHOD__, func_get_arg(0) - 1);  
    echo FB[func_get_arg(0) % 15] ?? func_get_arg(0), "\n";  
}}, 100);
```

変数禁止って言ってるのに

`func_get_arg(0)`

使ってるのずるくね？

ずるい

とってもずるい

しかしfunc_get_arg()を
禁止すると関数が完全に利用不
能になるので難易度急上昇↑↑

今回は妥協

今回は再帰呼び出しする
ことで状態更新を回避

さらなる難易度を
求めるあなたへ

| 難易度を下げる関数を禁止

- 状態を持てる関数
 - APCu, getenv(), ini_get()
- 変数テーブルの操作
 - compact(), extract()

| 難易度を下げる関数を禁止

- 定数の動的定義
 - `define()`, `constant()`
 - 更新不能のグローバル変数として状態を持てるため
- 禁止する文字を増やす
 - `{}` とか `:` とか(両方は重い)