

なぜ2018年にIDEと



Emacsを選ぶのか



Why was **IDE** and **Emacs** chosen, now in 2018?

お前誰よ



- うさみけんた (@tadsan) / Zonu.EXE
 - GitHub/Packagistでは id: zonuexe
- ピクシブ株式会社技術基盤チーム (pixiv.net)
- Emacs Lisper, PHPer
 - 入社前は自宅警備をしながらRuby書いてた
 - Emacs PHP Modeのメンテナ引き継ぎました
- Qiitaに記事を書いたり変なコメントしてるよ



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



Fri, 9th – Sat, 10th March
Coconeri hall



PHPer Kaigi 2018



A PHP mode for GNU Emacs

857 commits

12 branches

27 releases

39 contributors

GPL-3.0

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

zonuexe Merge pull request #422 from zonuexe/fix/setq-local

Latest commit 85f52b2 5 days ago

README.md

PHP Mode for GNU Emacs

build passing melpe 20180214.643 melpe stable 1.18.4

This project updates PHP Mode for GNU Emacs with features to make it more friendly to use with PHP 5.4 and later. This fork builds on the work of:

1. Turadg Aleahmad (Original Author)
2. Aaron S. Hawley
3. Lennart Borgman

All contributors listed below improved PHP Mode as well.

The current maintainers are:

1. Syohei Yoshida
2. Eric James Michael Ritz
3. USAMI Kenta (@zonuexe)

Please submit any bug reports or feature requests by creating issues on [the GitHub page for PHP Mode](#).

アジエインダ

1. PHPはどんな言語か
2. Emacsでなにができるか
3. IDEに何を期待するか

はじめに
言っておく

実用的なPHPの
Emacs環境構築

なんとかStormに負けたくない
Emacs 初級篇

2016-03-01 PHP BLT #3

エディタくらい
好きなの選びたい

勝つとか負けると
かの話ではない

エディタ選択は
専門職たるエンジニア
ニアの裁量の範疇

本当にそれでエラー
を起こさない
自信があるならな

僕は自信がない

pixiv <3 PhpStorm

希望者はIDEのライセンス購入
(IntelliJ IDEA, RubyMine)

PhpStorm先生は
お前の先輩よりも
PHPに詳しい

PhpStorm先生は

俺よりもずっと

PHPに詳しい

いますぐ生産性を
上げたいなら素直
にIDEを買ってあげ

IDE(IntelliJ IDEA/PhpStorm)

のライセンス料は

人件費よりずっと安い

回し者ではないので

宣伝はこの辺に

なぜ2018年にIDEと



Emacsを選ぶのか



Why was **IDE** and **Emacs** chosen, now in 2018?

なぜ

それが僕には
楽しかったから

(Just for Fun)

ほとんど^だ全^ての機
能をハックできる
エディタだから

ほとんどが全て

誇張ではない

エディタ機能の
多くの部分はLisp
で実装される

C言語で実装された関
数もラップできるし、
動的に上書きもできる

Lispの言語の
強力さもある

普通のやつらの上を行け ---Beating the Averages---

著者：Paul Graham

Copyright 2001 by Paul Graham

これは、Paul Graham: [Beating the Averages](#) を、原著者の許可を得て翻訳・公開するものです。

[プロジェクト杉田玄白](#)正式参加テキスト。

<著作権表示>

本和訳テキストの複製、変更、再配布は、この著作権表示を残す限り、自由に行って結構です。

(「この著作権表示」には上の文も含まれます。すなわち、再配布を禁止してはいけません)。

Copyright 2001 by Paul Graham

原文: <http://www.paulgraham.com/avg.html>

日本語訳: [Shiro Kawai](#) (shiro @ acm.org)

<著作権表示終了>

文中、Eric Raymondの "How to become a hacker" からの引用部分の訳は [山形浩生氏](#) の訳を 引用させていただきました。

<http://practical-scheme.net/trans/beating-the-averages-j.html>

ハッカーになろう ([How To Become A Hacker](#))

著者 Eric S. Raymond esr@thyrsus.com, [Thyrus Enterprises](#)

翻訳 [山形浩生](#) <hiyori13@alum.mit.edu> + 村川泰 + Takachin

目次

[なぜこんな文書を書いたか](#)

[ハッカーって何？](#)

[ハッカー的心構え](#)

1. [この世界は解決を待っている魅力的な問題でいっぱいだ](#)
2. [同じ問題を二度解くような無駄はいやだ](#)
3. [退屈と単純作業は悪](#)
4. [自由は善](#)
5. [心構えは技能の代用にはならない](#)

[基本的なハッキング技術](#)

1. [プログラミングを身につけること。](#)
2. [オープンソース UNIX 類のひとつを入手し、使いかたと動かしかたをおぼえること。](#)
3. [World Wide Web の使い方を学び、HTML を書くこと。](#)
4. [まともに英語ができないならば、身につけること。](#)

[ハッカー文化での地位](#)

<https://cruel.org/freeware/hacker.html>

職業エンジニアとしての
の見識を広めるために、
見知らぬ言語の見識を
深めるのは悪くない

異質なものに触れて
常用する言語を
相対化することで
得られるものはある

1. PHPは

どんな言語か

PHPを書いたこと
はありますか？

いろいろな
コードが書ける

| テンプレートエンジン

```
<!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>
```

| CGIっぽいスクリプト

```
<?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>';
```

| クラスベースOOP

```
<?php
namespace MyApp\Controller;
class BooksController extends BaseController{
    public function index() {
        $books = Books::getAll();
        $this->render(compact("books"));
    }
}
```

(当然ながら)

Webアプリに多用

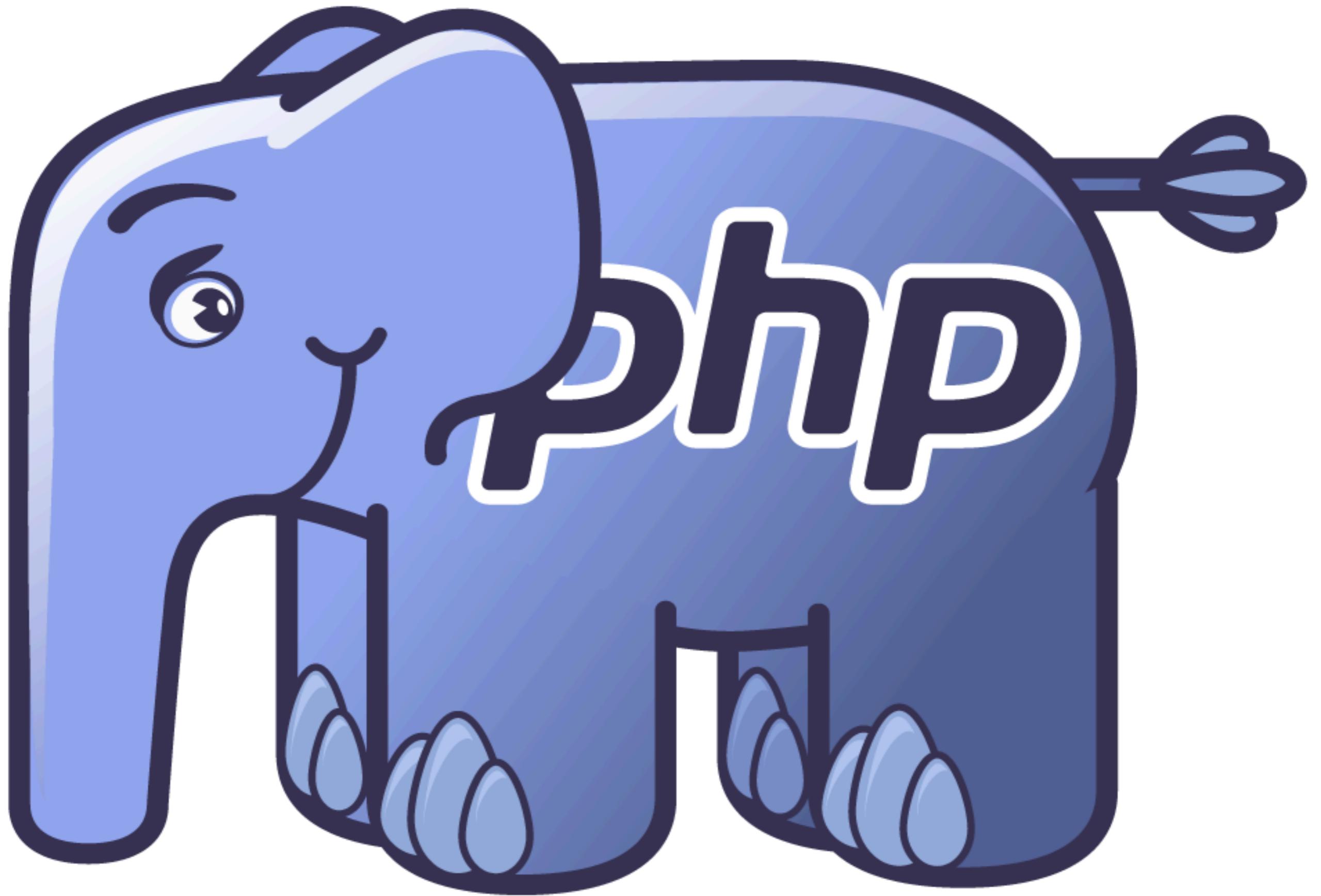
される言語

なんとなくPerlっぽさ(変数に\$)があり、PHP5ではちよつとJavaっぽい見たためのOOPが入ってる

むかししながらの
おてがるCGIっ
ぽいスクリプト

フレームワークを
利用した階層構造
アプリケーション

ど「ち」ちらもで「き」る
の「か」がPHP



ちやんとした
パッケージジマナー
ジヤカがある



Dependency Manager for PHP

Composer

npm, Bundlerのよ
うなもの

動的型検査

変数の型は
実行時まで
わからなない

型宣言がある

| 関数の型注釈（従来から可能）

```
<?php
```

```
/**
```

```
 * @param float $height
```

```
 * @param float $top
```

```
 * @param float $bottom
```

```
 * @return float
```

```
*/
```

```
function trapezoid($height, $top, $bottom) {  
    return (($top + $bottom) / 2) * $height;  
}
```

| 関数の型宣言 (PHP7から可能)

```
<?php
```

```
function trapezoid(float $height,  
    float $top, float $bottom): float  
{  
    return (($top + $bottom) / 2) * $height;  
}
```

JavaScript, Python,
Rubyなどと並ぶ動的
プログラミング言語

2018年現在おそらく唯一まともに
PHPの型について書いてある紙(雑誌)



PHPはオーブンプラス(モン
キーパッチ)はないので、クラ
スが実行時に拡張されるこ
とはあまり考えなくてよい

PSR

PHP標準勸告

本来のPHPは
カオスなCGI

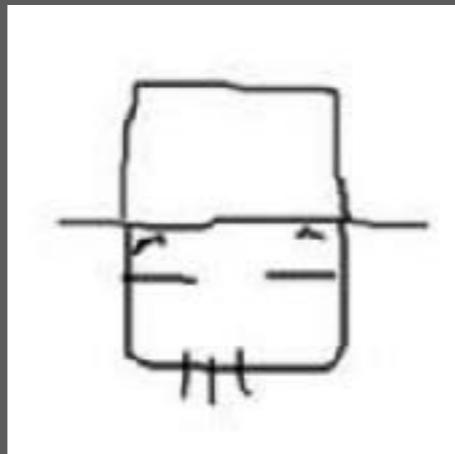
フレームワークでオブジェクトを相互運用できるようにするための足枷

实行环境

たいてい本番運用環境
(GNU/Linux, まれにFreeBSD)
と開発環境(Mac, Windows,
デスクトップLinux)は異なる

2. IDEに何を
期待するか

PhpStormはPHPのコンパイラ、
黄色いやつはゆるさない



@tanakahisateru said...



Takuto Wada

@t_wada

フォローする



JetBrains製の開発環境の「ほぼ Java」感とは、第三者のライブラリも含めてクラス名やメソッド名の検索や補完が快適にできて、かつ、リファクタリングの基本三要素(Rename, Extract, Inline)が完全自動化されている、ということです。後者は特に重要。

19:02 - 2017年1月11日

74件のリツイート 83件のいいね



2

74

83

https://twitter.com/t_wada/status/819379053530529792

精度の高い
静的解析と型推論

それに裏打ちされた、
入力補完(メソッド/プロパティ)、定
義ジャンプ、未定義検出、型エ
ラー検出、リファクタリング、
etc, etc...

大事なことは

自分の誤ちに
すぐに気付ける
こと

黄色いやつ

```
$this->hoge;
```

Field 'hoge' not found in PixivLibTestCase [more...](#) (⌘F1)

黄色いやつ

黄色いやつ

```
vsprintf("[%b] %s", $n, $msg);
```

Method call uses 3 parameters, but method signature uses 2 parameters

```
each ( format : string, args : array  
vsprintf("[%b] %s", $n, $msg);
```

次にすすめることを
示してくれること

Guzzle(HTTPクライアント)でリクエストする

```
<?php
require __DIR__ . '/vendor/autoload.php';
$result = (new GuzzleHttp\Client)->get(uri: 'https://httpbin.org/get');
```

レスポンスを得たい
次にどうする？

ソースコードを読む
ドキュメントを探す？

```
$result = (new GuzzleHttp\Client) -> get('url: https://httpbin.org/get');  
$result->
```

| | | |
|---|--|---|
|   | <code>getBody()</code> | <code>Psr\Http\Message\StreamInterface</code> |
|   | <code>getHeader(name : string)</code> | <code>string[]</code> |
|   | <code>getHeaderLine(name : string)</code> | <code>string</code> |
|   | <code>getHeaders()</code> | <code>string[][]</code> |
|   | <code>getProtocolVersion()</code> | <code>string</code> |
|   | <code>getReasonPhrase()</code> | <code>string</code> |
|   | <code>getStatusCode()</code> | <code>int</code> |
|   | <code>hasHeader(name : string)</code> | <code>bool</code> |
|   | <code>withAddedHeader(name : string, value : string string[])</code> | <code>static</code> |
|   | <code>withBody(body : \Psr\Http\Message\StreamInterface)</code> | <code>static</code> |
|   | <code>withHeader(name : string, value : string string[])</code> | <code>static</code> |

`^↓` and `^↑` will move caret down and up in the editor [>>](#)

[π](#)

動的言語であつても、
強力な静的解析機能で
サポートしてくれるの
が理想的なIDE

カオスなPHPを
御するために
IDEは強大な助け

3. Emacsで
何が出来るか



GNU Emacs

An extensible, customizable,
free/libre text editor — and more.

At its core is an interpreter for Emacs Lisp, a dialect of the Lisp programming language with extensions to support text editing.

↓ [GNU/Linux](#)

↓ [Windows](#)

↓ [macOS](#)



Welcome to [GNU Emacs](#), one component of the [GNU](#) operating system.

[Emacs Tutorial](#)

Learn basic keystroke commands (Emacs 入門ガイド)

[Emacs Guided Tour](#)

Overview of Emacs features at gnu.org

[View Emacs Manual](#)

View the Emacs manual using Info

[Absence of Warranty](#)

GNU Emacs comes with *ABSOLUTELY NO WARRANTY*

[Copying Conditions](#)

Conditions for redistributing and changing Emacs

[Ordering Manuals](#)

Purchasing printed copies of manuals

To start... [Open a File](#) [Open Home Directory](#) [Customize Startup](#)

To quit a partially entered command, type Control-g.

This is GNU Emacs 25.3.1 (x86_64-apple-darwin16.7.0, NS appkit-1504.83 Version 10.12.6 (Build 16G29)) of 2017-09-12

Copyright (C) 2017 Free Software Foundation, Inc.

Auto-save file lists were found. If an Emacs session crashed recently, type [M-x recover-session RET](#) to recover the files you were editing.

てす

いまどき
Emacs?

2017年の **php** 開発に



Emacsで勝つ 

PHP development in Emacs, 2017

pixiv

2017-08-19 NEEC Kamata [#opendevcon](#)
Open Developers Conference 2017 Tokyo

要約すると

Emacsの強さは拡張性
であり、基本機能は
残念ながらそれほど強
くはない

いまどきテキストエディ
タでIDEに対抗しよう
とすることは、竹槍で
戦闘機に対抗しようと
することに等しい

端末でSSHでログインし
た先の鯖でも動作する、
といったメリットはある

(これは開発よりはインフラのサーバオペレーション寄り)の需要ではある)

逆に「Emacsはターミナルで動かすもの」と誤解があるかもしれない

逆に「Emacsはターミナルで動かすもの」と誤解があるかもしれない



Spacemacs - VimとEmacsの良さを足した意欲的なエディタ

2018/02/12    0

 ソフトウェア, オープンソース

 [github](#), [Lisp](#), [Emacs](#), [GPL v3](#), [GUI](#), [Vim](#)

個人的にはAtomに昨年頭に乗り換えたのですが、2018年になってまたEmacsに戻ってきました。キーボードは何とか乗り越えた（機能拡張で）のですが、使い勝手の面でどうしてもEmacs以上にはならなかったようです。

そんなEmacs、Vimなどレガシーなエディタの中に住んでいる方に試して欲しいのがSpacemacsです。VimとEmacsユーザを両方満足させようという意欲的なエディタです。

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

ほかのツールを
組み合わせることで
匹敵する機能を持たせ
ることはできる

しかし

現時点で残念ながら
選ぶ・組み合わせるの
概念から抜け出せない

PhpStormには
埋められない
不満もある

PhpStormユーザは
「設定」しかできないが、
Emacsはいつでも
自分で「実装」できる

まとめ

すぐに強力な開発環境が欲しい
なら商用IDE

環境をハックし
て自分を前進さ
せながら Emacs

両方使ってもいい

プロとしてでの
道具を選ぼう

残念ながらEmacs

は鼻屑目に見て

非力

それはそれとして
Emacsをハック
するのは楽しい

自分とじっくりしよに
強くなれる

そんなわけでもない
いろいろ作ってきた



This repository

Search

Pull requests

Issues

Marketplace

Explore



emacs-php / psysh.el

Unwatch 1

1

Star 9

9

Fork 2

2

Code

Issues 1

Pull requests 0

Projects 0

Wiki

Insights

Settings

PsySH on Emacs, PHP interactive shell (REPL) <https://www.emacswiki.org/emacs/ZonuExe>

Edit

emacs

melpa

php

repl

[Manage topics](#)

24 commits

1 branch

4 releases

2 contributors

README.org

PsySH.el

melpa 20171022.2229

PsySH, PHP interactive shell (REPL)

Compatibility

- [php-mode](#)
- [auto-complete](#)
- [php-eldoc](#)



This repository

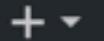
Search

Pull requests

Issues

Marketplace

Explore



emacs-php / phan.el

Unwatch

1

★ Star

1

Fork

2

Code

Issues 0

Pull requests 0

Projects 1

Wiki

Insights

Settings

Emacs Utilities for Phan (PHP static analyzer)

Edit

emacs

melpa

php

phan

major-mode

Manage topics

31 commits

1 branch

3 releases

2 contributors

README.org

phan.el

Emacs Utilities for [Phan](#) (PHP static analyzer)



This repository

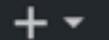
Search

Pull requests

Issues

Marketplace

Explore



emacs-php / composer.el

Unwatch 1

Star 6

Fork 3

Code

Issues 0

Pull requests 0

Projects 0

Wiki

Insights

Settings

PHP Composer interface for Emacs <https://www.emacswiki.org/emacs/ZonuExe>

Edit

emacs

melpa

php

composer

Manage topics

59 commits

1 branch

8 releases

3 contributors

README.org

composer.el

melpa 20180111.942 melpa stable 0.0.8

Composer interface for Emacs.

Specification

See [Composer Documentation](#). `composer.el` supports [Environment variables](#) (currently we support only `COMPOSER_BIN_DIR` and `COMPOSER_HOME`).



This repository

Search

Pull requests

Issues

Marketplace

Explore



nlamirault / **phpunit.el**

Unwatch 4

★ Unstar 23

Fork 13

Code

Issues 3

Pull requests 0

Projects 0

Wiki

Insights

Emacs mode to phpunit and test command line tool

emacs

phpunit

205 commits

2 branches

16 releases

10 contributors

GPL-2.0

nlamirault Merge pull request #53 from nlamirault/develop

Latest commit a137067 on Nov 27, 2017

README.md

phpunit.el

license GPL 2

- Master melpa stable 0.16.0 ✓ PASSED coverage 24%
- Develop : melpa 20171127.301 ✓ PASSED coverage 23%

Manage the [PHPUnit](#) tests from Emacs (>= 24.3)

Installation

The recommended way to install `phpunit.el` is via [MELPA](#):



This repository

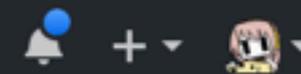
Search

Pull requests

Issues

Marketplace

Explore



zonuexe / dotfiles

Unwatch 1

Star 15

Fork 2

Code

Issues 0

Pull requests 0

Projects 0

Wiki

Insights

Settings

Branch: master

dotfiles / .emacs.d / site-lisp / pixiv-dev.el

Find file

Copy path

zonuexe Update

a90112a 11 days ago

1 contributor

176 lines (146 sloc) | 6.02 KB

Raw

Blame

History



```
1  ;;; pixiv-dev.el --- p(ixi)v -*- coding: utf-8 ; lexical-binding: t -*-
2
3  ;; Copyright (C) 2016 USAMI Kenta
4
5  ;; Author: USAMI Kenta <tadsan@zonu.me>
6  ;; Created: 2016-04-01
7  ;; Modified: 2017-11-28
8  ;; Version: 1.1.0
9  ;; Keywords: processes tools php
10 ;; Package-Requires: ((emacs "24.3") (flycheck "1") (psysh "0.0.4"))
11 ;; URL: https://github.com/zonuexe/dotfiles/tree/master/.emacs.d/site-lisp
12
13 ;; This file is NOT part of GNU Emacs.
14
```

~~tadsan先生の~~
~~次回作にご期待~~
~~ください~~

お前が

Emacs Lisperに

なるんだよ