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

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

開発してるもの

- PsySH（PHP） - PR/機能追加
- whoops!（PHP） - PR/機能改善

- php-mode.el（Emacs Lisp） - PR/機能追加/改善
- composer.el（Emacs Lisp） - 開発者(オーナー)
- psysh.el（Emacs Lisp） - 開発者(オーナー)
- phpunit.el（Emacs Lisp） - 共同メンテナ
- こまかいのいろいろ

pixiv

はじめに

言ってお く

~~Photoshop~~

PhpStorm

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

（僕が使ってるのはPHPだからPHPの話をするけど、あなたの使ってる[ピー]言語も例外ではない）

僕の仕事

＝

同僚をPhpStormに改宗させること

しょーもないヒューマンエラーを静的解析の力で捻じ伏せること

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...

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

では、なぜEmacsを利用するのか

Emacs=環境

（誇張ではない）

エディタの機能
＝ただの関数

標準機能すら
上書きできる

標準機能を置換する
べんりパッケージ
いろいろある

[ピー]よりも

ずっとはやい！

(Vimよりは遅いかも…)

ともあれ高機能なエディタは
好きな機能を
組み合せて構築できる
フレームワークのようなもの

なんとかStormに負けたくない

Emacs 初級篇

2016-03-01 PHP BLT #3

当時紹介したもの（1）

- php-mode.el
  - 今も開発してるから入れてくれ頼む
- phpunit.el
  - PHPUnitを簡単に実行できる
- psysh.el
  - PHPのREPLを実行できる
  - 選択範囲のコードを実行したりできる

当時紹介したもの（2）

- TRAMP
  - sshのファイルを直接編集できる
- Magit
  - Gitインターフェイス
  - エディタ内でgit blameできる
- magit-find-file
  - リポジトリ内の絞り込み検索

当時紹介したもの（3）

- php-eldoc
  - PHPの標準関数の引数が表示できる
  - 標準関数ごときでぐぐってはいけない
- smartchr.el
  - キーを連打すると入力文字列が入れ変る

そのほかべんりなもの

- flycheck
  - 文法チェックやLintの非同期実行
  - チェッカーを独自実装できる
- Projectile
  - プロジェクト管理ツール

プロジェクト固有のマイナーモード

- pixiv-dev.el
  - GitHubに置いてあるよ
  - べんりコマンドとか
- 社内用のLintツールとか

弱点

いちいち通信が発生する
ので、めちゃくちゃ遅い

PhpStorm先生に学ぶ

automatic deployment

保存したファイルを(S)FTPで
勝手に送ってくれるやつ

Emacs auto-deployment

`copy-file-on-save` is a minor mode to copy the file to another path on after-save-hook. This not only saves the backup in the project specific path, it also you can realize the deployment to the remote server over TRAMP.

Why `copy-file-on-save` ?

The original name of this feature was **auto-deployment**. It was named after JetBrains' automatic deployment function. But in Emacs this function can be realized by TRAMP's excellent file system abstraction layer. That is, deploying to remote server is just done with only `copy-file` function.

vs direct editing on TRAMP

TRAMP can log in to remote server from Emacs and edit the file directly. This means that you do not need to keep a full copy of the project on your client PC. But at the price you will feel latency to all file system operations.

The disadvantage of TRAMP is Emacs Lisp compatibility. Especially in the case of several packages, processing depending on the file system is lacking consideration or it may be slow even if it works. For example, Magit also works via TRAMP, but it's very slow.

https://github.com/zonuexe/emacs-auto-deployment

まとめ

素のエディタは

竹槍

勝つには

A. IDEが必要ないほど
PHPの仕様に詳しくなる

俺が
人間静的解析器だ

B. いろんなツールやパッケージを組み合せて必要な機能を実現していく

しかし道は
険しい

最終的には快適な
開発環境で、
バグ起こしにくい
開発を実現させる
ことが目的

やってしく
気持ち

最後にお願い

EmacsでのPHP開発で困ってることを共有してほしい

続きはWebで

@tadsan

https://github.com/zonuexe/emacs-php-development