

ポエム

[@ppworks](#)

ポエムとは

中の人もちよつとこまっている

---



# ポエムとは

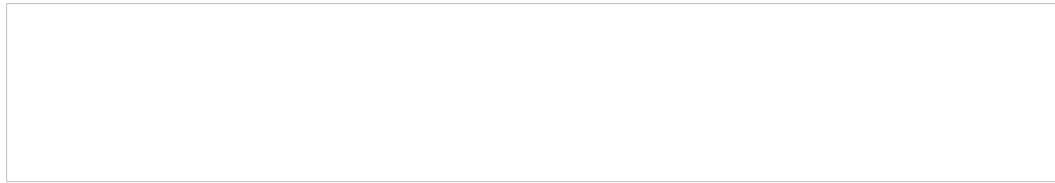
---



まずは

、 [pplog.net](http://pplog.net) へ

アクセス(´ω´)



- 
- つながれない
  - のこせない
  - しゃべれない

つなげられない

# つながれない

---

- twitterのフォロワーを自動でフォローとかない(〜ω〜)
- わざわざtwitter IDで検索しないといけない(〜ω〜)
- もしくは、大きな肉球をくりっくしてみましよう(〜ω〜)



のこせなしい

# のこせない

---

- ポエム投稿ごとのURLが用意されていない(〜w〜)
- あんまり考えないで書いて良い。考えた時点でポエムじゃない。ポエムは内面から溢れだすものである(〜w〜)

しやべれなない

# しゃべれない

---

- いわゆるコメントなんてないの(〜ω〜)
- 勝手につく足あとはないよ(〜ω〜)
- どうしても喋りたい場合はリアルで会った時にでも話しな(〜ω〜)

大事な話

# githubにポエム書いた

---

コンセプトデザインは、[@ken\\_c\\_lo](#) 氏だ



# 速攻作った

---

- コンセプトは結構ゆっくりもんでた。
- よしやるか！と言って思い立ってすぐだ。
- p4dハッカソンぽい雰囲気ですすめた

# ワークフロー

---

- デザイナー & エンジニアで開発するベストな雰囲気あった
- 生煮えプルリク。。。
- FIXMEでざっくりデザインからのFIXYOU。。。
- (〜ω〜)。。。
- github flow(ぽいナニカ)

[Gitを使ったデザイナーとプログラマの協業について話してきた #P4D](#)  
[#phpcon2013](#)



# 数字の話

---

- 9/26~1/15(111日間)
  - v257 => 2.3 deploy/ day
  - 866 commits

```
irb(main):001:0> Post.count  
=> 2746  
irb(main):002:0> Star.count  
=> 22493  
irb(main):003:0> User.count  
=> 616
```

# human.txt 置いた

---

<https://www.pplog.net/humans.txt>



# 用語の話(1)

---

ユーザーが色々勝手に用語を作ってくれてる

- ポエ消し
  - 前の（主に恥ずかしい）ポエムを消して次のポエムを書くこと
  - タイトルを「ポエ消し」とすることが多い
- ポエみ
  - 私の中からポエムが溢れだしてきた状態
  - ツイッター上でポエみ成分が検出された後ポエムが投稿されている事が多い

# 用語の話(2)

---

- 追ポエ
  - ポエム投稿後に、編集してポエムを追加すること
  - ポエム投稿時しか購読者への通知はないので、こっそり足すこともあれば、足したことをツイッターなどでほのめかすこともある
  - まったくほのめかさずにしれっと追ポエしているパターンもあるので、読者は気が抜けないのだ
- ポエム
  - 投稿内容のこと
  - pplogユーザーが発する言葉すべて
  - pplog.netのこと
  - pplogの正式な読み方は運営から発表はないし、誰も知らない。



# 皆様の声

---

- 2013.12.19 [Node.jsでスクレイピング \(pplog駆動\)](#)
- 2013.12.27 [ポエムを刻め pplogがとてもおもしろい](#)
- 2013.12.30 [pplog良さそう。](#)
- 2014.01.03 [チラシの裏の世界へようこそ](#)
- 2014.01.04 [pplogが素敵](#)
- 2014.01.04 [最新の1件しか公開できないブログのような「pplog」](#)
- 2014.01.05 [ポエムを共有するpplogに登録してみた](#)
- 2014.01.12 [実家の宗教と許嫁の話](#)

ここからは、ど  
うでもいしい話

# add-ons

---





# herokuの話

---

- herokuが東京リージョンじゃなくても工夫で良くなる。でも待ってます(´ω´)
- s3はusにして配信時にCDN使おう(CloudFront) pplogはこっち
- メインターゲットがJAPANで揺ぎないなら、s3 tokyoにして直接配信でもいいよ（でもs3がそんなに速くない？）
- fragment cacheの活用 => MemCachier
- メール投稿 => mandrillの Inbound 機能でweb hookを選択して設定
- DeployHook => HipChatに通知
- パフォーマンスチェック、1dynoで落とさない => New Relic
- SSL => 男のロマン

# railsの話

---

- fragment cacheは、ユーザーによって変わる箇所には使わない工夫が必要。あとlocaleごと取るようにする。`cache [I18n.locale, poem]`
- `papertrail`の`ignore`(`Rendered|Read fragment|Writer fragment`)を指定
- あたりまえだけどSQLも読もう
- twitterアイコンはtwitter側で変更すると参照先URLが消滅するのでs3にコピーしておく良さそう

# 便利な何か

---

- HipChat API
  - github からの通知
  - heroku への deploy 通知
  - bugsnap からの通知
  - [twitter のエゴサーチ \(@fakestarbaby氏作\)](#)
  - サイト内の動向アレコレ
  - 開発陣はみんなHipChatと眺めている(´ω`)
- CIは[wecker](#)使ってる
  - passしたら、herokuのステージングにデプロイとか出来るけど、使うのやめた

# herokuでなにか作ったので

---

[we love heroku](#) へ登録しました。

新年会のあとは

、 [pplog.net](http://pplog.net) へ

アクセス(´ω`)