

技術発信の信頼性を高める 10のポイント

10 things to improve the credibility of your article



pixiv Inc.
USAMI Kenta

pixiv

お前誰よ



- うさみけんた (@tadsan) / Zonu.EXE / にゃんだーすわん
- ピクシブ株式会社 pixiv事業本部 エンジニア
 - 最近はピクシブ百科事典(dic.pixiv.net)を開発しています
- Emacs Lisper, PHPer
 - Emacs PHP Modeを開発しています(2017年-)
- PHPerKaigi コアスタッフ、(昨年)PHPカンファレンス実行委員

今回のお題

プロポージャー

技術発信の信頼性を高める10のポイント

ピクシブ株式会社 うさみけんた

私が細々と技術記事を書き始めて10年以上が経ちますが、未熟な頃にどのようなことに気をつけて発信していたか、そして現在ほかの人の記事を読んで「惜しい」と感じるかを10のポイントにまとめて紹介します。

10という数は適当です

(任意の進数での $10_{(n)}$ と
いうことにしてください)

個人的な遍歴



- 2003年頃～ フリーソフトウェア／OSSの概念を知る(プログラミングできない)
- 2005年頃～ Vectorに簡単な設定ファイルなど公開(プログラミングできない)
- 2008年頃～ 簡単なスクリプトをWeb日記とかに載せはじめる
- 2010年頃～ Lispを知ってプログラミング言語にのめりこむ、Rubyはじめる
- 2012年頃～ 定職に就く、Emacs Lispはじめる、Qiita/GitHubはじめる
- 2014年頃～ QiitaにPHPの記事を書きはじめる
- 2016年頃～ WEB†DB PRESS(雑誌)に初めて寄稿、PHPカンファレンス

初めて技術記事(っぽいもの)
書いてから紙の雑誌に
寄稿するまで8年くらい

2000年代中旬から
いろいろなブログをつまみ
食いして技術を学んだ

わからないなりに濫読した
ブログの(—見無駄な)知識が
数年後に結びつく経験

初期の発信の原動力

- 環境構築やカスタマイズなど、自分で工夫したことについてのメモ
- リアルで教えてもらった技術知識についてのノート
- Webに断片的な情報しかない技術を体系的にまとめたかった
- 誤情報や中途半端な情報への憤り → 自分でやるしかない

誤情報への憤り

問題を解決したいときに
間違った情報が出てくる
とムカつく

自分の記事を
自分で読んで
ムカつかないように

誤情報への憤り

- 何か問題を解決したいときに間違った情報が出てくるとムカつく
 - こちらは読者として一方的に発信の成果を得ているだけなのだが、不条理にも憤りの感情を発信者(ブログの著者)に向けてしまう
 - さまざまな事情により適切ではない内容についても「嘘」とすら感じてしまう
- 誤った情報の伝播は自分でより適切な発信をすることでしか止められない

誤情報とは何か、読者にできること

- 単純な誤記・タイプミス・リンクミスなど
 - 誰でも気付けるもの、著者と同等以上の知見がないと指摘できないのものもある
 - Qiitaなら編集リクエストを送るとよい
- 記述対象への著者の事実誤認に基いた記述
 - コメント欄やSNSなどで指摘するといいかもしれない
- 時間的な変化によって事情が変わってしまったもの

間違った情報にしないための工夫

- 単純な誤記・タイプミス・リンクミスなど
 - 記事の公開前に自分で気をつける
 - 公開前に誰かにチェックしてもらう
- [宣伝] 昔からの友達がShodoというAI校正ツールを開発しています
<https://shodo.in/>

間違った情報にしないための工夫

- 記述対象への著者の事実誤認に基いた記述
 - 「対象の知識についてマスターする」は完全な対策にはならない
 - 全知全能の神ではないので無謬の技術記事にはならない
- 普遍的な正しさを語るのではなく、局所的にする
 - 「このコードは私の環境のPHP 5.4ではこう動く」

信頼できる情報源

- 公式マニュアル・標準仕様など
 - MDN Web Docs: HTML/JavaScript/CSS
 - 検索キーワードに「mdn」を付ける癖をつけると簡単にアクセスできます
 - PHPマニュアル、Pythonドキュメント、Rubyリファレンスマニュアル
 - そのほか日本のコミュニティ有志が独自に公開しているものもある
- その実装がGitHubなどで読めればソースコードを直接参照する

間違った情報にしないための工夫

- 記述対象への著者の事実誤認に基いた記述
- 極力、信頼できる情報源を直接参照するようにする
 - 曖昧な理解のブログをソースにしても誤解に誤解が増幅される
- 基本的と思えること(標準関数など)に関しても信頼できる情報源をあたり、読者も参照できるようにリンクする

固有名詞や言葉の使いかたに気を配る

- PHP, Ruby, JavaScriptなどの大文字小文字の使いかた
- 専門用語の使いかた
- 言葉使い
 - かしこまった文体（です・ます）
 - 論文のような文体（だ・である）
 - カジュアルなくだけた文体
 - 関西型言語

コードが動くことをどう保証するか

- 手元にいろんなOSや複数の言語処理系のインストールをするのは難しい
 - オンラインのサンドボックスサイトを使う
 - <https://wandbox.org/> 多くの言語・バージョンをサポート
 - <https://3v4l.org/> PHP特化・非常に多くのバージョンをサポート
 - URL共有できるので、第三者が見られ動作確認のソースにする
- 最近はDockerでの環境構築できるようになった

初心者のおすすめの発信

- そもそも正しい／間違いという枠に収まらない発信から始める
 - インストールや作業などのログ
 - 勉強会イベント・カンファレンスに参加した感想
 - 自分なりのカスタマイズやこだわりについてまとめる

書きたいネタがない…

- 書きたいと思ったことは既に誰かが書いてる
 - 個人的意見：大事なことは誰が何回書いてもいい
 - 既存の記事に関して、自分なりの意見や経験を付け加えるだけで価値のあるアウトプットになる