

Typst で登壇できるようになったので練習させてくれ

第 177 回 PHP 勉強会@東京

@takeokunn

Jun 25, 2025

Agenda

1. 自己紹介
2. 今日の話について
3. きっかけと違和感
4. Typst との出会いと感動
5. 実用に耐えるまでの道のり
6. Typst 登壇はこう変えた

自己紹介

自己紹介

- たけてい @takeokunn
- 仕事では PHP コミュニティ所属
- 個人ではテキストエディタコミュニティ所属



今日の話について

今日の話について

- Typst でスライドを生成できるようにした
- その内容をブログから抽出して AI にスライドを生成させた
- 実際に登壇がうまくいくのか検証する回
 - ▶ AI パワポカラオケ
- プロンプトをチューニングしきれてないのは御愛嬌
 - ▶ Gemini 2.5 Pro のデフォルト

AI 生成方法



FLEETING

AIを駆使してTypstスライドを生成する

takeokunn • 2025.6.10

fleeting typst

元ブログ(1)



FLEETING

NixでTypstをBuildしGitHub Pagesでホスティングする

takeokunn • 2025.1.26

fleeting

nix

typst

org-mode

元ブログ(2)



FLEETING

Typstテーマを自作した

takeokunn • 2025.5.6

fleeting

typst

org-mode

元ブログ(3)



FLEETING

Typst自作テーマで登壇スライドを作成した

takeokunn • 2025.6.1

fleeting typst

元ブログ(4)



FLEETING

Typstとpdfpcでプレゼン機能を実現する

takeokunn • 2025.6.8

fleeting typst

きっかけと違和感

かつての相棒：Google Slides

- かつては Google Slides は最高の相棒だった
- 画像位置を調整するのが簡単
- 過去に作ったスライド資産がある
- 外部への共有も簡単

当時のスライド作成フロー

- Markdown でアウトラインを書く
- 過去スライドをコピペしてくる
- アウトラインに沿ってスライドを作る
- スピーカーノートを用意する

心に刺さった “小さなトゲ”

- プレーンテキストで管理できない
- AI による自動生成できない
- Emacs から編集できない
- etc...

違和感の正体

- エンジニアとしての“業”（カルマ）
- Git 管理が恋しい
- 再利用性が低い（No DRY!）
- 自動化できない

エンジニアの性（さが）

- 僕らは、スライドという成果物も“コード”として管理したい生き物なんだ！

Typst との出会いと感動

運命の出会い: Typst

- そんな僕が出会ったのが Typst
- まさに「プレゼン資料も構造化・宣言的に書ける」言語
- 探し求めていた理想がそこにあった

最初の感動ポイント

- コードでスライドが書ける
 - ▶ Emacs から離れなくていい
- Git で管理できる
 - ▶ 変更履歴はすべてお見通し！
- テーマが自作できる
 - ▶ DRY 原則、スライドにも！
- コードハイライトが美しい

Typst スライドを支える技術（しくみ）

- Polylux
 - ▶ Typst のためのスライドエンジン
- code1st
 - ▶ コードを美しく見せるスタイリスト
- Nix
 - ▶ すべてを固める魔法の環境

実用に耐えるまでの道
のり



現実には甘くなかった

- 理想の環境はすぐそこに見えたが、やってみると色々な壁にぶつかる
- 「言うは易く行うは難し」を実感する日々

やってみて困ったこと（理想と現実のギャップ）

- テーマの調整が超むずい
- polylux や code1st のドキュメントとひたすらにらめっこ
- フォント設定
 - ▶ この日本語フォントを使いたいだけに…
- 画像の配置
 - ▶ ドラッグ&ドロップの偉大さを思い知る

それでも続けた理由（Typst の魅力に立ち返る）

- 記法が明快で、根本的な書き心地は最高だった
- スライド構成が自分の思考にフィットした
- アウトラインから考え、そのままプレーンテキストのままスライドを作る

補助ツールとの統合（そして“最強”の環境へ）

- Typst 単体ではなく、エコシステム全体で課題を解決していく
- pdfpc: PDF プレビュー
 - ▶ スピーカーノートを表示
 - ▶ ついに「登壇」できるようになった瞬間
- GitHub Actions
 - ▶ CI による検証
- GitHub Pages
 - ▶ Web 上に PDF を公開する

Typst 登壇はこう変えた

- ▶ 発表資料を“資産”にする

Typst 登壇はこう変えた

Before/After

- Typst を手に入れた結果、登壇準備のすべてが変わった
- Before/After で振り返る、スライド制作プロセスの変化
- 「きっかけと違和感」で感じていた違和感がすべて解消されていく

変化 1: 迷わず構成から入れる

- Before
 - ▶ まず Google Slides を開き、過去のスライドのどこからコピペするか悩む…
- After
 - ▶ まず Emacs でアウトライン（構造）を書くことに集中できる

変化 2: 見た目を気にしすぎなくていい

- Before
 - ▶ テキストボックスの位置、フォントサイズ、etc
 - ▶ ピクセル単位の調整地獄
- After
 - ▶ 自作テーマを適用すれば OK
 - ▶ 「見た目は後からついてくる」 安心感

変化 3: 再利用性・再現性が圧倒的

- Before
 - ▶ 過去スライドからコピペ
 - ▶ コピペミスやうっかり編集しちゃったなどヒューマンエラーが頻発
- After
 - ▶ 過去の資産は Git リポジトリに
 - ▶ Nix のおかげで再現性のある PDF を生成

Typst で登壇する未来

- 自分だけのテーマを育てる文化
- 発表のたびにテーマを少しずつ改善していく
 - ▶ 自分だけの“刀”を研ぐ感覚
- コード × ブログ × 登壇の三位一体
- Slide as Code の世界へようこそ

まとめとメッセージ

- Google Slides への違和感から始まり、困難を乗り越え、最強の環境を手に入れました
 - ▶ 一度書いてみれば、この快感が分かる
- その場限りの資料ではなく、再利用可能な“資産”を作りたい人は、ぜひ Typst へ！

ご清聴ありがとうございました