

現代におけるプロダクト開発と PHPを選定するワケ

2017.07.15 HANATANI Takuma #phpkansai

と、

大仰なタイトルですが、

ゆるくいきたいと思います

今日話すこと

- 自己紹介
- このセッションで伝えたいこと
- ユースケースから見るPHPである意義が強いレイヤとそうでないレイヤ
- 意義が強くないレイヤにおけるPHPの選定理由
- PHPをプロダクト開発においてきちんと運用するには
- おわりに

今日話すこと

- 自己紹介

- このセッションで伝えたいこと
- ユースケースから見るPHPである意義が強いレイヤとそうでないレイヤ
- 意義が強くないレイヤにおけるPHPの選定理由
- PHPをプロダクト開発においてきちんと運用するには
- おわりに

自己紹介

花谷拓磨

- ~ 2016.02 **PHP(Plain)**
- ~ 2016.10 JS / **PHP (Cake, Laravel)**
- 2017.01 ~ GNEX Ltd. JS
- 2017.04 ~ pixiv inc. JS / **PHP**
- FRONTEND CONFERENCE 2017実行委員長
- バックエンドはPlain PHP派
- Better Shell Script枠もPHPでちょい書き



今日話すこと

- 自己紹介
- このセッションで伝えたいこと
- ユースケースから見るPHPである意義が強いレイヤとそうでないレイヤ
- 意義が強くはないレイヤにおけるPHPの選定理由
- PHPをプロダクト開発においてきちんと運用するには
- おわりに

このセッションで伝えたいこと

「なんとなく」の言語化

どのようなユースケースにおいて
PHPを利用していますか？

その採用理由はなんですか？

「現場がPHPだから」

「PHPが使われているから」

「なんとなく」

なぜ 「現場がPHP」 なのか

なぜ 「PHPが使われている」 のか

慣習・風習を言語化する

目的

目的

①

Webでの開発における多くの選択肢の中、
今あえてPHPを採用する意義について再確認する

目的

②

その上で、他の選択肢を加味して、

PHPが適切である場合とそうでない場合を明確化する

目的

③

その適切化どうかの要素を常に意識することによって

意思決定をより円滑にする

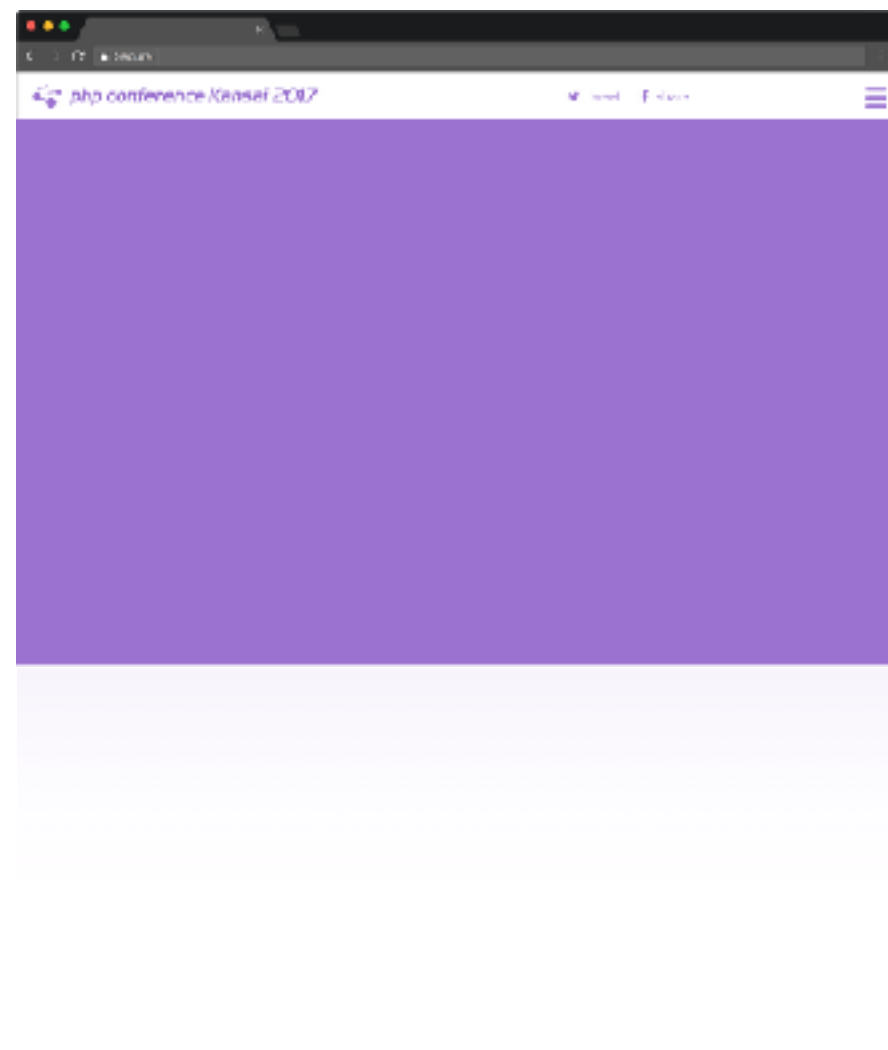
本題

今日話すこと

- 自己紹介
- このセッションで伝えたいこと
- **ユースケースから見るPHPである意義が強いレイヤとそうでないレイヤ**
- 意義が強くはないレイヤにおけるPHPの選定理由
- PHPをプロダクト開発においてきちんと運用するには
- おわりに

どのようなユースケースにおいて
PHPを利用していますか？

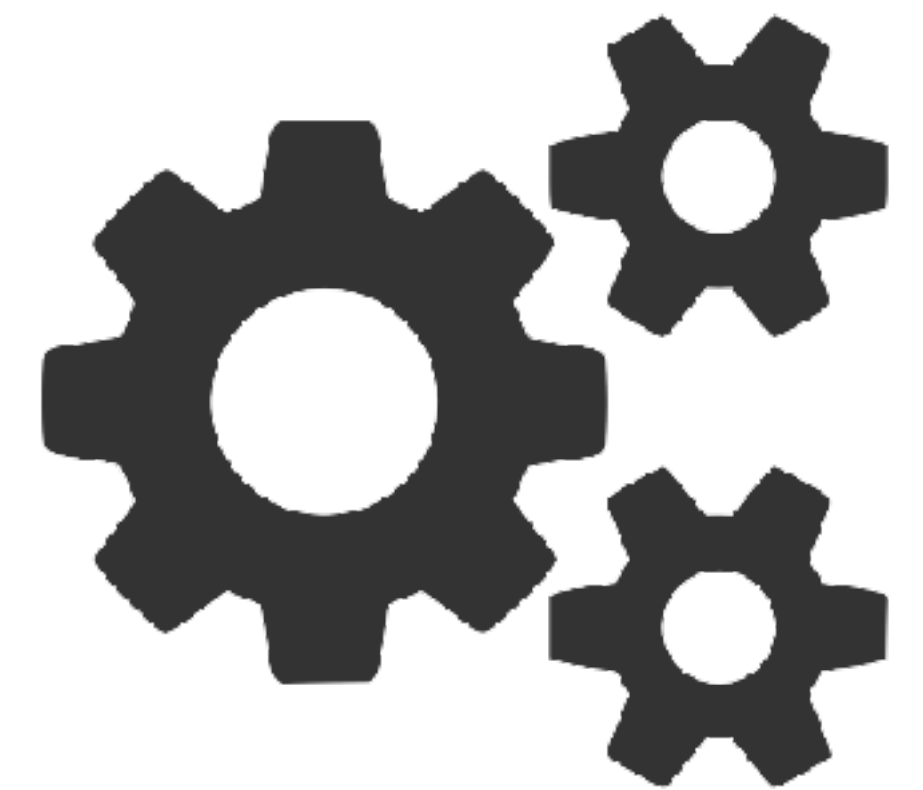
ざっくりユースケース



HP制作+α



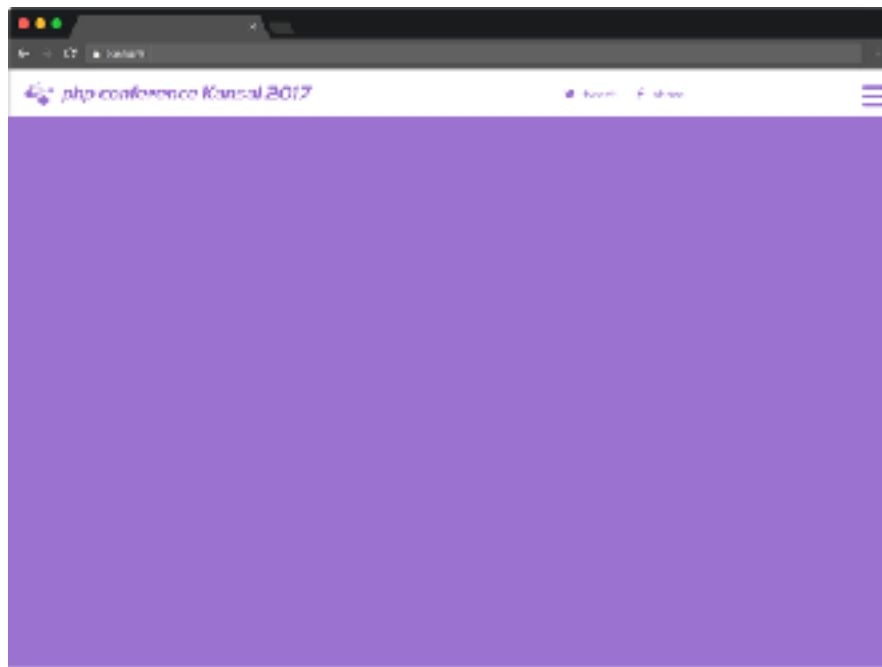
CMSのカスタマイズなど



Webアプリ・システム開発

ざっくりユースケース

ホームページ制作



- ・ ここでは、いわゆる「ホームページ」の制作系かつ通常のコーディング案件のことを指す
- ・ 基本的には静的サイトで問題ないが、一部に関してどうしても $+\alpha$ としてロジックが必要な場合に、その解決法として局所的にPHPが利用されているプロジェクト
- ・ 例) お知らせ欄の動的生成 問い合わせフォームの作成など
- ・ 基本的に更新頻度が高くなく、ホームページなので機能的な拡張はほぼない

HP制作 $+\alpha$

ざっくりユースケース

CMS



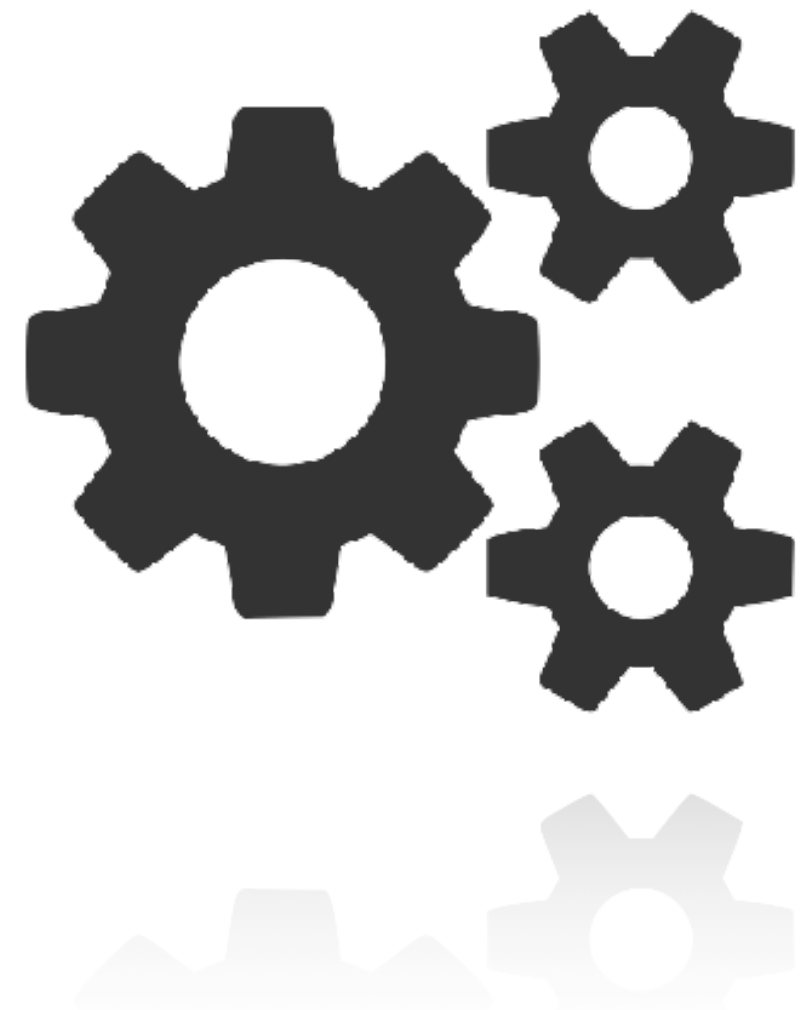
CMSのカスタマイズなど

- ・ CMSに対して、ベースを維持しながらも要件に合わせて拡張を行うもの
- ・ WordPressだけでも世界のWebサイトの1/4を占めている*というデータがあるほど多い案件であり、CMSの利用自体が動機になることも多く、それ自体が要件に含まれることも多い
- ・ 例) WordPress/MT EC-CUBEなど
- ・ 動的に更新を行いたい箇所はCMSに乗せることによって解決するため、比較的更新が少ない

* <https://venturebeat.com/2015/11/08/wordpress-now-powers-25-of-the-web/>

ざっくりユースケース

Webアプリケーション・システム開発



Webアプリ・システム開発

- ・ いわゆる普通のWebサービスに相当するものの開発。PHPの場合はWAF(Web Application Framework)を使わない場合もあるが、一般的にはそれらを利用して行うアプリケーションの開発
- ・ いわゆる「プロダクト開発」もこれらを言い換えたようなもの
- ・ 例) 業務系のシステム pixiv
- ・ サービス系の開発であれば勿論非常に機能追加の要望やロジックが大きくなるからゆえのメンテナンスコストがかかる

PHPのプロジェクトの特徴

圧倒的な「Webサイト」の多さ
メンテナンス頻度の少ないモノの多さ

PHPだからこそそのメリット

- HTMLと密結合で開発できるからこそ「Webサイトに対しての**小さなロジックの差し込み**」の容易さ
- 既に普及しているCMSが要件に入るからこそ排他的な技術選定と、それによって特定製品に集中する**ナレッジの豊富さ**
- それらの需要によって生まれた**公開環境の設営の容易さ**

「難しくないことをはやくかつうまくやる」

こと

今日話すこと

- 自己紹介
- このセッションで伝えたいこと
- ユースケースから見るPHPである意義が強いレイヤとそうでないレイヤ
- **意義が強くはないレイヤにおけるPHPの選定理由**
- PHPをプロダクト開発においてきちんと運用するには
- おわりに

「難しくないこと」 ではないものについて

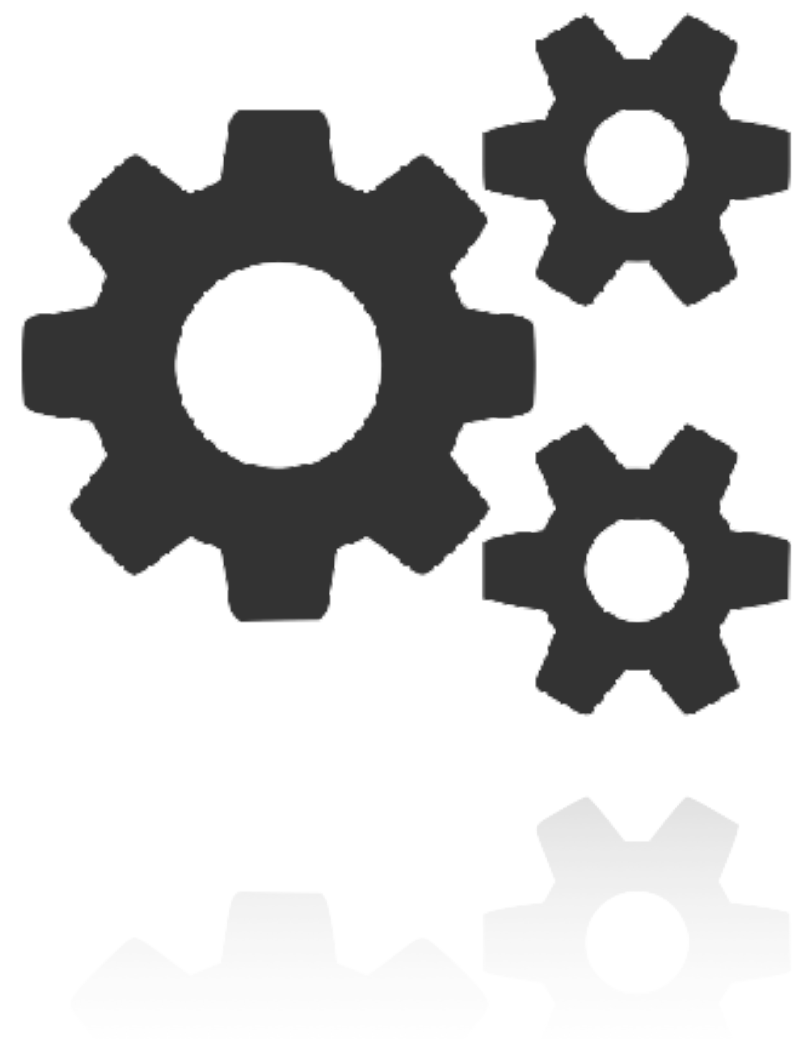
たとえば

「Webアプリケーション・システム開発」

Webアプリケーション開発

例) より良い選択肢があるパターン

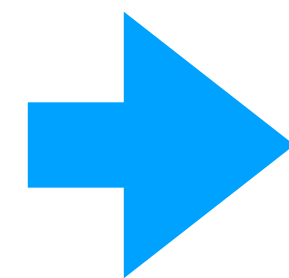
- ・ SNS的なサービスを開発したい
- ・ ユーザーや投稿基本的なCRUDが存在し、ユーザーに紐付けられた投稿に対してリアクションデータなどが複雑に紐づく
- ・ 画像リソースや動画リソースなど、様々なバイナリファイルも受け付け、変換処理を受け持つ……



Webアプリケーション開発

例) より良い選択肢があるパターン

- ・ SNS的なサービスを開発したい
- ・ ユーザーや投稿基本的なCRUDが存在し、ユーザーに紐付けられた投稿に対してリアクションデータなどが複雑に紐づく
- ・ 画像リソースや動画リソースなど、様々なバイナリファイルも受け付け、変換処理を受け持つ……

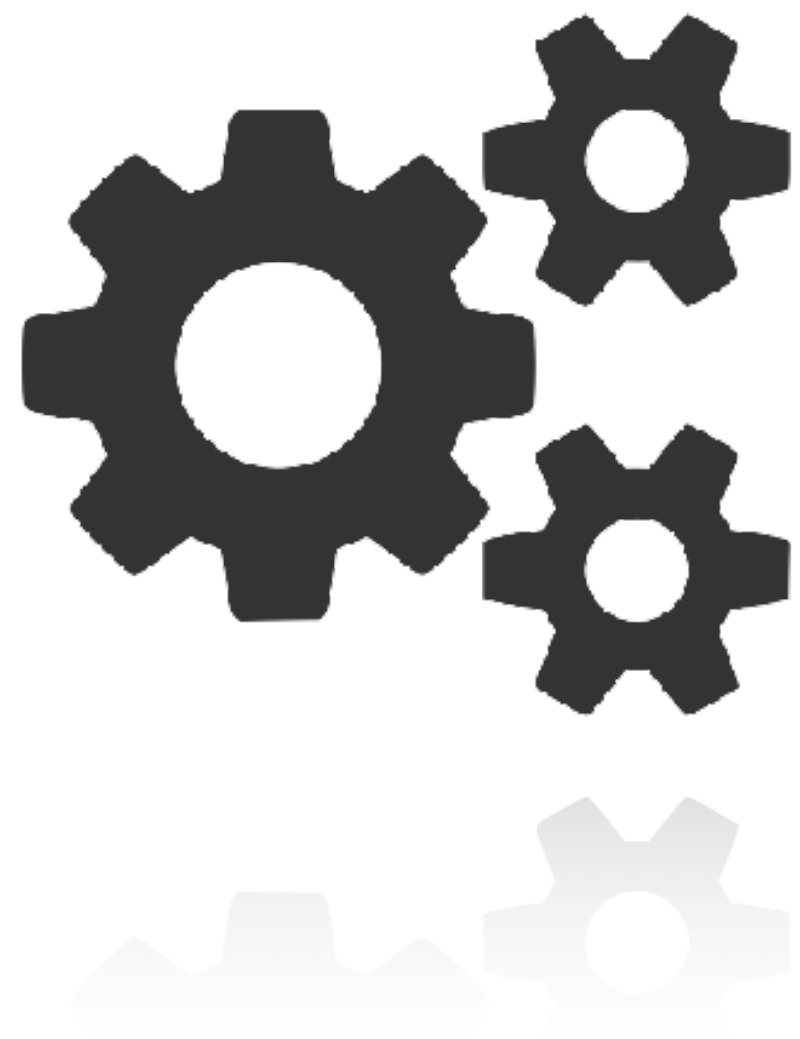


複雑なリレーションを操るためのActive Recordが完備、
画像周りもNative gemで対処できるRailsで作るのが最適解では？

Webアプリケーション開発

例) より良い選択肢があるパターン2

- ・ 業務アプリケーションを開発したい
- ・ Webインターフェースとして提供されるが、そこ自体が重要ではなく、後ろに存在するシステムが重要
- ・ 業務上の数字に関わるため、出来る限り堅牢なシステムを組みたい

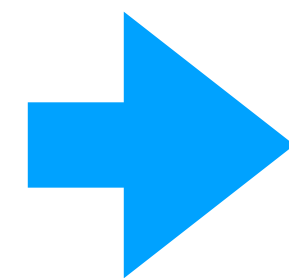


Webアプリケーション開発

例) より良い選択肢があるパターン2



- ・ 業務アプリケーションを開発したい
- ・ Webインターフェースとして提供されるが、そこ自体が重要ではなく、後ろに存在するシステムが重要
- ・ 業務上の数字に関わるため、出来る限り堅牢なシステムを組みたい



PHPにはない強力な型検査が存在し、かつ数値計算周りを含めて豊富な資産と実績のあるJavaで開発することが最適解では？

複雑な開発においては
最適解ではないかもしれない

とはいえリソース・**人員の確保**の容易さ
どんな案件でも**取り敢えずは使える**立ち位置

※ 例えば、とある求人サイトにて、PHPの求人は2600ほどあるが、Rails案件は1200ほどと約半数しかない

「リソース・要件の都合で一旦PHP」
という選択

それらの状況下でどう秩序を保つか 》

今日話すこと

- 自己紹介
- このセッションで伝えたいこと
- ユースケースから見るPHPである意義が強いレイヤとそうでないレイヤ
- 意義が強くはないレイヤにおけるPHPの選定理由
- **PHPをプロダクト開発においてきちんと運用するには**
- おわりに

秩序ある開発を行うために
必要なこと

秩序ある開発を行うために必要なこと

- 現代的な技術基盤、フレームワークの利用
- middlewareによる拡張などの、非破壊的なエコシステム
- パッケージマネージャによる外部の資産の依存管理
- （人によっては）ある程度実用的な型システム
- Web APIとしての運用するためのアーキテクチャの構築
- etc...

秩序ある開発を行うために必要なこと

現代的な技術基盤、フレームワークの利用

の、非破壊的なエコシステム

それ、**Laravel + 最新のPHP環境でできるよ**

- パッケージマネージャ
- (人によっては) ある程度実用的な型システム
- Web APIとしての運用するためのアーキテクチャの構築
- etc...

秩序ある開発を行うために必要なこと

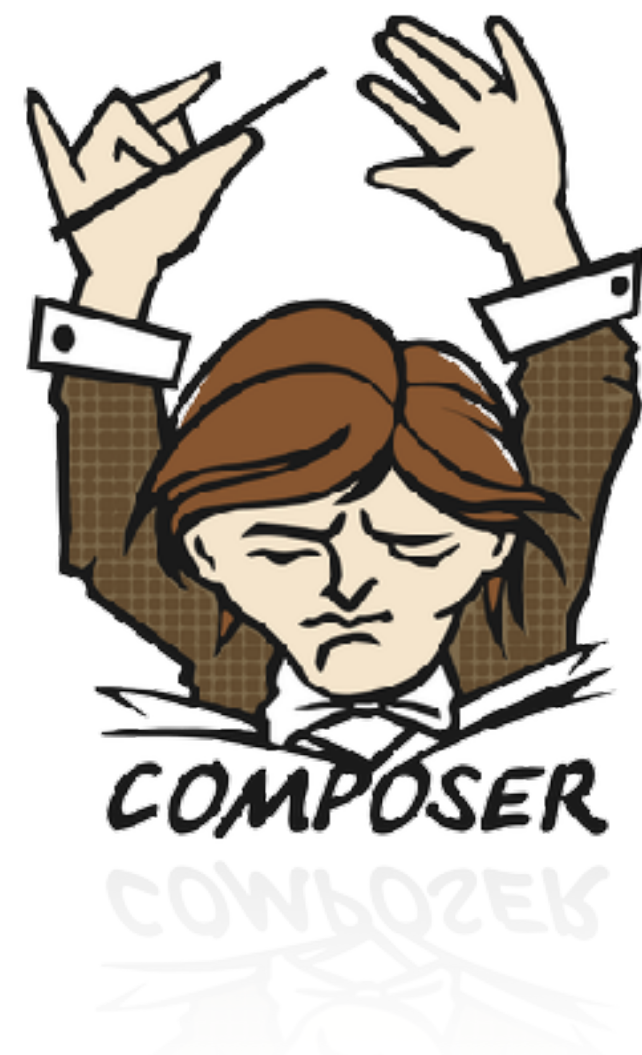
ある程度実用的な型システム



- PHP7以降から、厳密ではないものの、タイプヒントに**スカラ型**が追加実装され、型宣言となった Hack
- 通常の間数制限にオプショナルで型を定義することによって、明示的に型を設けた箇所に関しては、その型であることが保証される
- 正しくない場合は「**実行時に**」例外を投げる
- 静的な型検査が行われるわけではないので**過度な期待は禁物**

秩序ある開発を行うために必要なこと

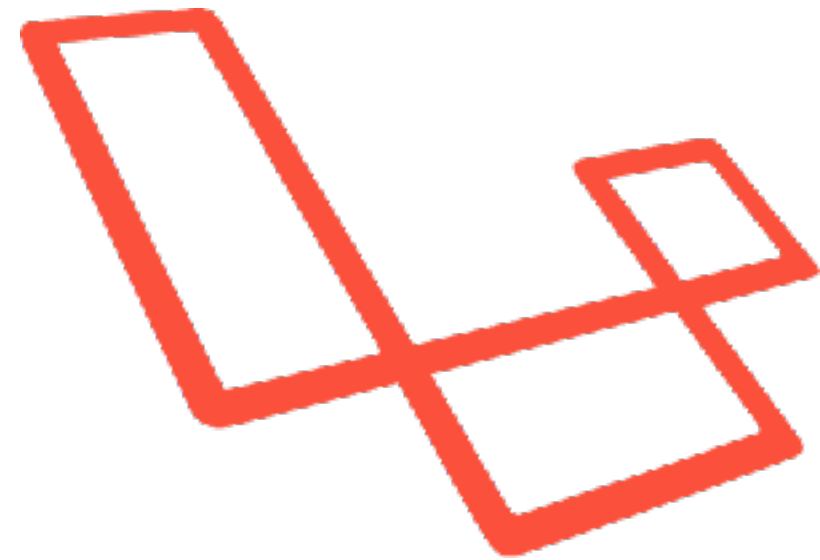
パッケージマネージャによる外部の資産の依存管理



- 気がついたらPEARが死んでいた
- 現代的なプロジェクトかつライブラリを必要する場合は既に必ず利用されているといっても過言ではないプロダクト
- 国内においては光の遅さを指摘されがち
- 依存関係をJSONファイルで記述することで簡潔に管理できるようにし、また他の言語のものと違ってNative Extensionの利用が不可であるためハマりづらい（デメリットでもある）

秩序ある開発を行うために必要なこと

現代的な技術基盤、フレームワークの利用



- 言わずと知れたモダンPHPフレームワーク
- middlewareの利用で効率的かつ管理しやすい形での拡張が可能
- トレンドのWeb API形式でのバックエンド実装も容易
- Webサービスを運営するエンジニアからも「フレームワーク自体の機能も多く、パッケージの追加によって多くの問題が解決可能であるため楽」という意見もあがりやすく、Rails的に解決していきたいモチベーションにも対応可能

「PHPだからできない」

「PHPだからできない」 ことはない

「PHPだからできない」 ことはない

※ ただし筋が良いとは限らない

今日話すこと

- 自己紹介
- このセッションで伝えたいこと
- ユースケースから見るPHPである意義が強いレイヤとそうでないレイヤ
- 意義が強くないレイヤにおけるPHPの選定理由
- PHPをプロダクト開発においてきちんと運用するには
- **おわりに**

まとめ

まとめ ①

歴史的経緯とメンテナビリティ

- PHPはその歴史上、他の言語などで作られるものと比較して、比較的メンテナンスの軽いものについての割合が高い傾向にある
- その一方で、PHP自体の手軽さをそのままに、継続的なメンテナンスや機能拡張への要望へ答えるようにアプリケーション開発も進化してきた
- とはいえ現在においてはWebアプリケーションの開発を行う選択肢としては、Ruby on Railsなどの他の言語や他の言語のフレームワークによる開発が一般的になってきた

まとめ ②

PHPを選択する意義の薄いこと・そうでないこと

- ・ 特にWebアプリケーション開発を専門に行う場合は、先述のようにそれ専用の技術が生まれているため、なんでもそつなくこなすPHPをわざわざ選ぶ意義は薄い（逆に言うと選んではいけないわけではない）
- ・ 逆に、特定のCMS指定の場合はそれに依存する技術選定しかできない、HP作成の場合はHTMLとある程度密結合に書くことが多い利点が生きてくることにより、他の選択肢を選ぶ強いモチベーションはない

まとめ ③

他の選択肢と比較してのPHPの採用メリット

- ・ 依然として古くからあることによるナレッジの多さ、業界におけるPHPを記述することが可能な人材の豊富さは揺るがない
- ・ また、アプリケーション開発以外を中心に行っていたとしても、ある一定まではアプリケーション開発においても同じ知見が適用できるという点で、HP制作などがメインの状況下においてアプリケーション開発を行うという選択肢に踏み切りやすい

PHPは「どこでも65点を出しやすい」言語

積極的に何か使いたいものがあれば、
その技術を使うと良い

「仮置き」しやすい特徴を有効活用することで
意思決定のオーバーヘッドを減らしつつ、
よりよいソリューションを模索しましょう