

# Laravel

---

# MVCフレームワーク

# Contents

---

- フレームワークとは
- MVCとは

# フレームワークとは

# フレームワークとは

---






## フレームワークとは

- ある言語で作るシステムの枠組み. 考え方.
- よく使う関数やクラスが用意されている.
- 規則が決まっているので, 誰が書いても似たコードになる.

# フレームワークとは

---

## メリットとデメリット

-  組み込まれている機能があるので実装の負荷が少ない.
-  コードの品質担保による保守管理の負荷低減.
-  セキュリティ面での担保.
-  学習コスト.
-  環境構築.

# フレームワークの例

---

## 各言語のフレームワークの例

- PHP:           Laravel, CakePHP, Symphony
- Ruby:           Ruby on Rails
- Python:        django, flask
- JavaScript:    Vue.js, React, Angular

※他にもたくさんあります... !

# Laravelの特徴

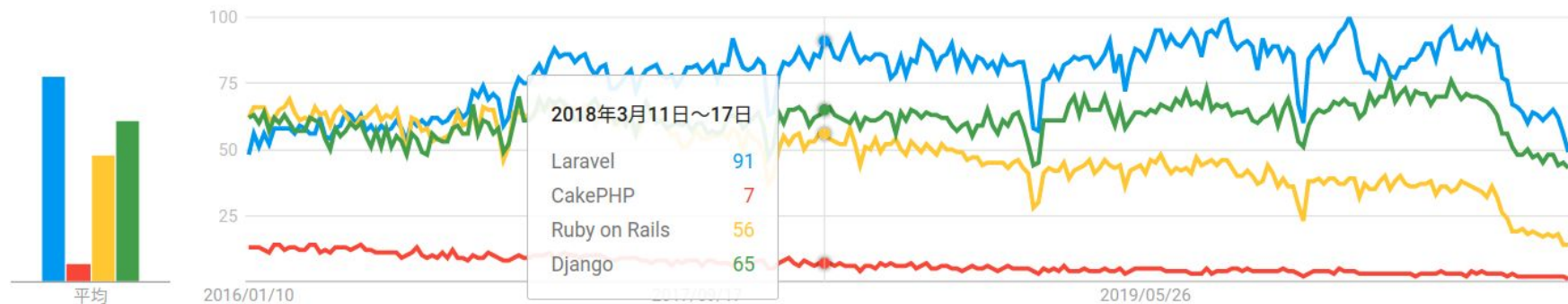
---

- 今最もアツいPHPフレームワーク！
  - Webサービスで使用する機能が豊富に用意されている.
    - DB操作関連の機能(論理削除など)
    - メール通知機能
  - 画面を柔軟に作成できるbladeテンプレート.
  - `artisan`コマンドによる手軽なライブラリ操作.
- 
- ? とにかく自由(ディレクトリ構成, 処理の記述場所など)



# Googleトレンドで見るLaravelのアツさ

人気度の動向 ?



# MVC

# 雑なMVCの説明

---

MVCとは、役割分担の概念！

- これまでは適当なPHPファイルに処理を全部書いていた！
- ファイル数が増える...
- あの処理、どのファイルに書いたっけ...??
- バグあったけど次の処理が他の人の書いたファイルに... (チーム開発)

-> いずれ詰む未来が見える...

-> せや、処理毎に別々のところに書いて管理や分類をしやすくしたろ！

# 雑なMVCの説明

---

## MVCの3つくらいでええやろ！

- Model
  - データの操作を行う役割.
  - データの「作成」「参照」「更新」「削除」し, その結果を出力
- View
  - ブラウザの表示領域を作成する役割.
  - 受け取ったデータ処理の結果をhtmlに埋め込み, 出力する.
- Controller
  - リクエストとレスポンスを行う. その間のことをいい感じにやる.
  - ModelとViewに司令を出す, 処理の流れを決める役割.

# 雑なMVCの説明

---

## その他

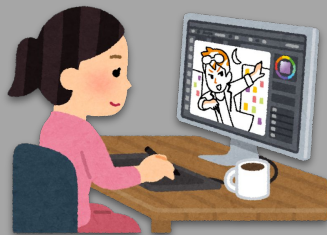
- Route

- アクセスされたURLに応じて処理を振り分ける.
- URLとControllerの関数の対応を決める.
- 実際の処理はControllerを通じてModelやViewが実行する.

※【参考】他のフレームワークでは「MVT」も存在！

- Model: MVCのModel
- View: MVCのController
- Template: MVCのView

# MVCのイメージ



View



Route



Controller



Model

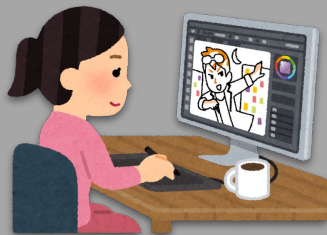


DB



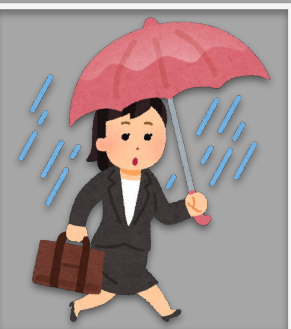
ブラウザ

# MVCのイメージ



View

データの一覧ページ見たい！



ブラウザ



Route



Controller



Model



DB

# MVCのイメージ



View



Route



ブラウザ



Controller

関数Aを実行してね！



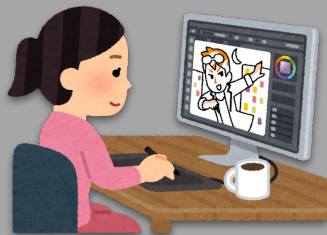
Model



DB



# MVCのイメージ



View



Route



ブラウザ

いい感じにデータ取ってくる  
関数実行して！



Controller



Model



DB

# MVCのイメージ



View



Route



ブラウザ

whereしてorderByして...  
ほらよ！



Controller



Model



DB

# MVCのイメージ



View

このデータ表示しといて！



Controller



Model



DB



Route



ブラウザ

# MVCのイメージ



View

くられッ！！

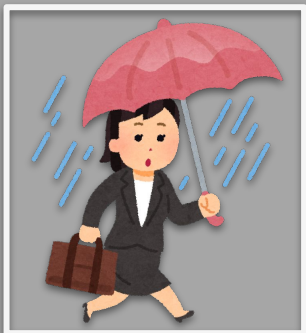


Controller



Model

やったぜ！



ブラウザ



Route



DB

# Laravelのコツ

## (フレームワーク全般)

# (Laravelに限らず)フレームワークのコツ

---

習うより慣れろッ！

- 初見で理解することは不可能！
- まずはアプリケーションを作って動かそう！
- 処理を見ながら、どこでどの処理を行っているのかを考えよう！
- MVC間でのデータの流れを追えるようになるとGood！！

慣れゲー & 周回ゲー！

- 何度も作ることで処理のイメージが湧き、余裕が生まれる.
- 「そういえば前回もこの処理書いたな」でOK.
- 10周くらいするとだいぶわかる！

# (10周くらいして)自分でつくる時のコツ

---

フレームワーク > 越えられない壁 >>> 人間

- フレームワークは人類の叡智！
- 素人が安易に改造すると台無し.

余計なことはしない！

- 細かい設定などは弄らない！
- 自分で実装せずにフレームワークに用意された機能を使い！
- ゴリ押しで書くな！（よくある処理はドキュメントに載ってる）

## ... ということでtodoリストを実装

---

1. 下記の資料にアクセス！！
  - a. <https://github.com/taroosg/laravel8-crud-tutorial>
2. 【超重要】下記の手順でAWS Educateアカウントにログイン
  - a. <https://aws.amazon.com/jp/education/awseducate/>にアクセス
  - b. 「AWS Educate にサインイン」をクリック
  - c. ページ上段「My Classroom」をクリック
  - d. 今回のクラスに入る-> AWS console
3. 今後作業途中で再度ログインが必要になっても上記手順でログイン！