



# 補足資料その1



# この講座で扱う事

PHP・Laravelでできる事/変数/関数/配列/連想配列/if文/foreach文/フォーム/HTTP/GET/POST/SSL/XSS/CSRF/ファイル書込/データベース接続/PDO/phpMyAdmin/プリペアドステートメント/バインド/プレースホルダ/トランザクション/セッション/クッキー/クラス/トレイト/名前空間/オートロード/composer/Laravel/MVC etc..

とにかくわかりやすさを重視！



# 講座作成の動機



# ホームページ+α

---

- ・ 日付を表示
- ・ お問い合わせフォーム
- ・ 新着情報
- ・ ブログ

その日の天気を表示

- ・ ・ 気象庁からデータを持ってくる

# フロントエンドとバックエンド

クライアントサイドとサーバーサイド



クライアント



サーバー



# PHPの年表



1995 PHP1.0

2004 PHP5.0.0

2014 PHP5.6.0

2015 PHP7.0

2017 PHP7.2

2020 PHP8.0

毎年アップデート中

# Webアプリ(ユーザーのインプット)

---

家計簿、勤怠管理システム、在庫管理システム、タスク管理システム、ブログ、マッチングサイト、ランキングサイト、ショッピングサイト、レビュー(口コミ)サイト、画像投稿(共有)サイト、動画投稿(共有)サイト、会員専用サイト などなど

# フレームワーク (枠組み) のメリット

---

よく使う機能をあらかじめ用意してある

- ・ 車輪の再発明を防げる
- ・ 開発スピード大幅アップ

セキュリティにもしっかり配慮



# よく使う機能の一例

---

ログイン機能(管理者と利用者それぞれ)

ソーシャルログイン(Twitter, Facebook)

データベースとのやりとり

日付計算 (Carbon) などなど

2段階認証

API連携(ぐるなびなど外部サイトとの連携)


# PHPフレームワークのトレンド

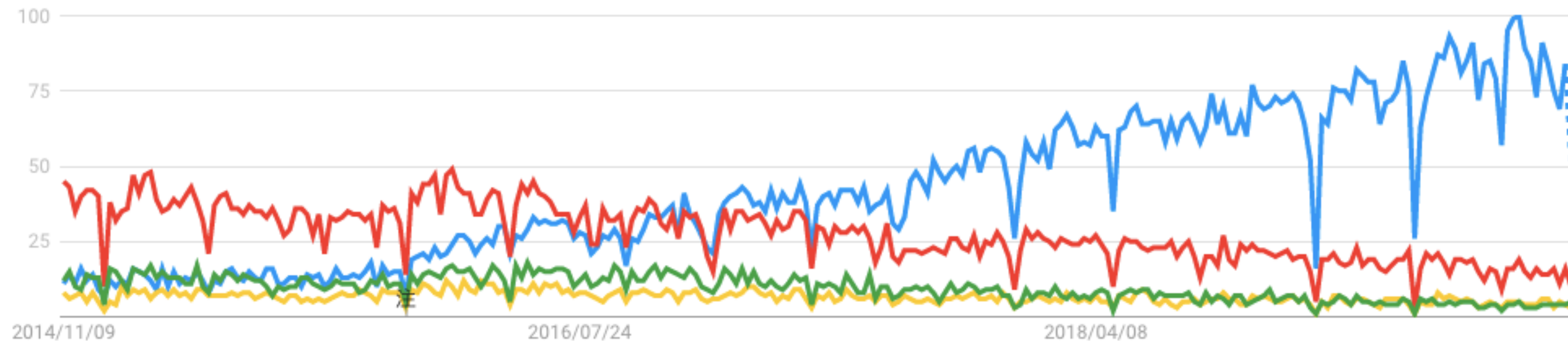
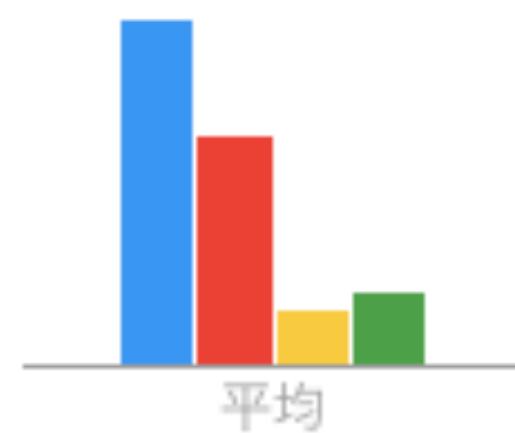
Laravel  
トピック

CakePHP  
トピック

CodeIgniter  
トピック

FuelPHP  
トピック

人気度の動向 





# Laravelの年表

---

2011年 Laravel 1.0

2017年1月 Laravel5.4 PHP5.6.4以上

2017年8月LTS Laravel5.5 PHP7以上

2019年9月LTS Laravel6.0 PHP7.2以上

2022年2月LTS Laravel9.0 PHP8.0以上

約半年毎にリリース


-> Laravel8より1年毎にリリース



# 開発環境をつくる



# 開発環境をつくる




XAMPP(win)/MAMP(mac)

・ ・ Apache/MySQL(MariaDB)/PHP(7.2以上)

Google Chrome ・ ・ ブラウザ

Visual Studio Code ・ ・ エディタ



# PHPの書き方・ 表示方法



# PHPの書き方・表示方法



ファイル名は xxx.php  
HTMLの中に書ける

<?php echo “test”; ?>  
php開発時は 末尾 ?> の省略も多い



# 変数と定数



# 変数と定数

変数 ・ ・ \$ 定数 ・ ・ const (多), define (少)

数字 ・ ・ \$value = 123;

文字 ・ ・ \$name = “山田”;

文字内に ”” があれば ” で囲む

文字連結は .

# 変数と定数

先頭は文字かアンダーバー  
大文字小文字は分ける  
できるだけ意味のわかる名詞・英語で

定数は一般的に全て大文字

変数のデバッグは `var_dump()` で





# 配列と連想配列



# 配列 (1 行)



左から3番目なら  
`$array[2];`



# 配列 (2行・3列)



横が行  
縦が列



# 配列 (3行、4列)



仕組みは同じ



# 連想配列

---

名前と値がセット。

```
$test = [  
  name => '本田',  
  height => '170cm',  
  hobby => 'サッカー'  
];
```

# 配列と連想配列の違い

配列 ・ 数字(順番固定)と値がセット。

`$array[1];`

連想配列 ・ キーと値がセット。

キー => 値

`key => value`

`$array["key"];`



# 連想配列 (学校に例えてみる)



# 連想配列

## 1組



0:本田



1:香川



2:長友



3:乾



4:大迫

## 2組



0:川島



1:柴崎



2:槇野



3:長谷部



4:酒井



# 連想配列

## 2年

### 1組



0:遠藤



1:内田



2:松井



3:岡崎



4:中村

### 2組



0:今野



1:駒野



2:大久保



3:楢崎



4:闘莉王

## 1年

### 1組



0:本田



1:香川



2:長友



3:乾



4:大迫

### 2組



0:川島



1:柴崎



2:槇野



3:長谷部



4:酒井

# 連想配列

福岡小学校

2年

1組

2組

熊本小学校

2年

1組



0:遠藤

1:内田

2:松井

3:岡崎

4:中村

2組



0:今野

1:駒野

2:大久保

3:檜崎

4:関莉王

1年

1組



0:本田

1:香川

2:長友

3:乾

4:大迫

2組



0:川島

1:柴崎

2:榎野

3:長谷部

4:酒井



# 課題：連想配列あれこれ





# 演算子(計算や判定)



# 演算子(計算や判定) 抜粋

四則演算子 `+`, `-`, `*`, `/`, `%`

比較演算子 `>`, `>=`, `+=`, `===`, `!==`

論理演算子 `and`, `&&`, `or`, `||`, `xor`

`!==` `null` や `empty`, `isset` などによく使う

`==` は使わない `===` を使う

`++` でインクリメント(1ずつ増える)



# 条件分岐と繰り返し



# 条件分岐など



プログラムが得意なもの  
条件分岐(信号機など)と繰り返し

If (elseはできるだけ使わない)  
foreach (配列の抽出)

for, switch, while, do-while



# 関数の考え方



# 関数 ・ function



関数 ・ function ・ 機能

# 関数・・・何らかの処理





# 数学の関数と似ている


入力(インプット)



出力(アウトプット)

$$f(x) = 5 * x + 10$$

# 関数は2種類



組み込み関数 ・ ・ 準備してある関数

ユーザー定義関数 ・ ・ 自由に作れる関数



# 関数のつくり方 (基本)

```
function 関数名 (引数)
{
  ～処理～
  return 戻り値;
}
```

関数名は名詞か動詞+名詞

# 関数あれこれ



計算・文字の整形・日付時刻  
配列・ファイル・CSV・画像  
ネットワーク・クッキー・セッション  
メール・データベース・セキュリティ など  
など

関数リファレンス

<https://www.php.net/manual/ja/funcref.php>



# 関数 文字列 一部

mb\_strlen() 文字列長さ取得

str\_replace() 置換

implode() 文字列結合

explode() 文字列分割

preg\_match() 正規表現で検索

substr() 指定文字から文字列取得

trim() 指定文字を削除

<https://www.php.net/manual/ja/book.mbstring.php>

<https://www.php.net/manual/ja/book.strings.php>

# 関数 配列 一部

`in_array()` 指定値が含まれているか

`array_slice()` 配列から取り出す

`array_merge()` 配列同士の結合

`array_push()` 配列の追加

`array_map()` 配列全てに関数を適用

`array_shift()` 配列先頭から取り出す

<http://html2php.starrypages.net/php/array-funcs>

<https://www.php.net/manual/ja/book.array.php>



# 組み込み関数 in ユーザー定義関数

例) 郵便番号チェック (本来は正規表現

- ・ ハイフンを削除
- ・ 7文字かどうか

関数名は camelCase か snake\_case で。付録として、よく使う関数と、知っておきたい英単語を付けますので、ご参考にしてみてください。

# できるだけ関数を使おう

昔話です。

ゴルフ予約システム  
1ファイル関数なしで約3,000行



とにかく読みづらい。。  
できるだけ関数化+ファイル分割を。





# 変数のスコープ

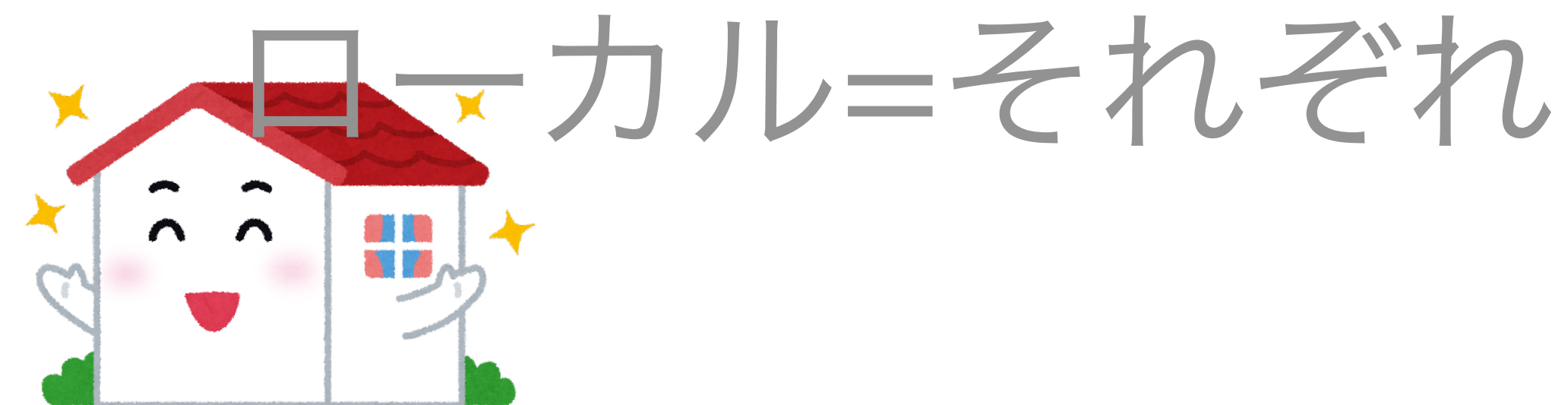
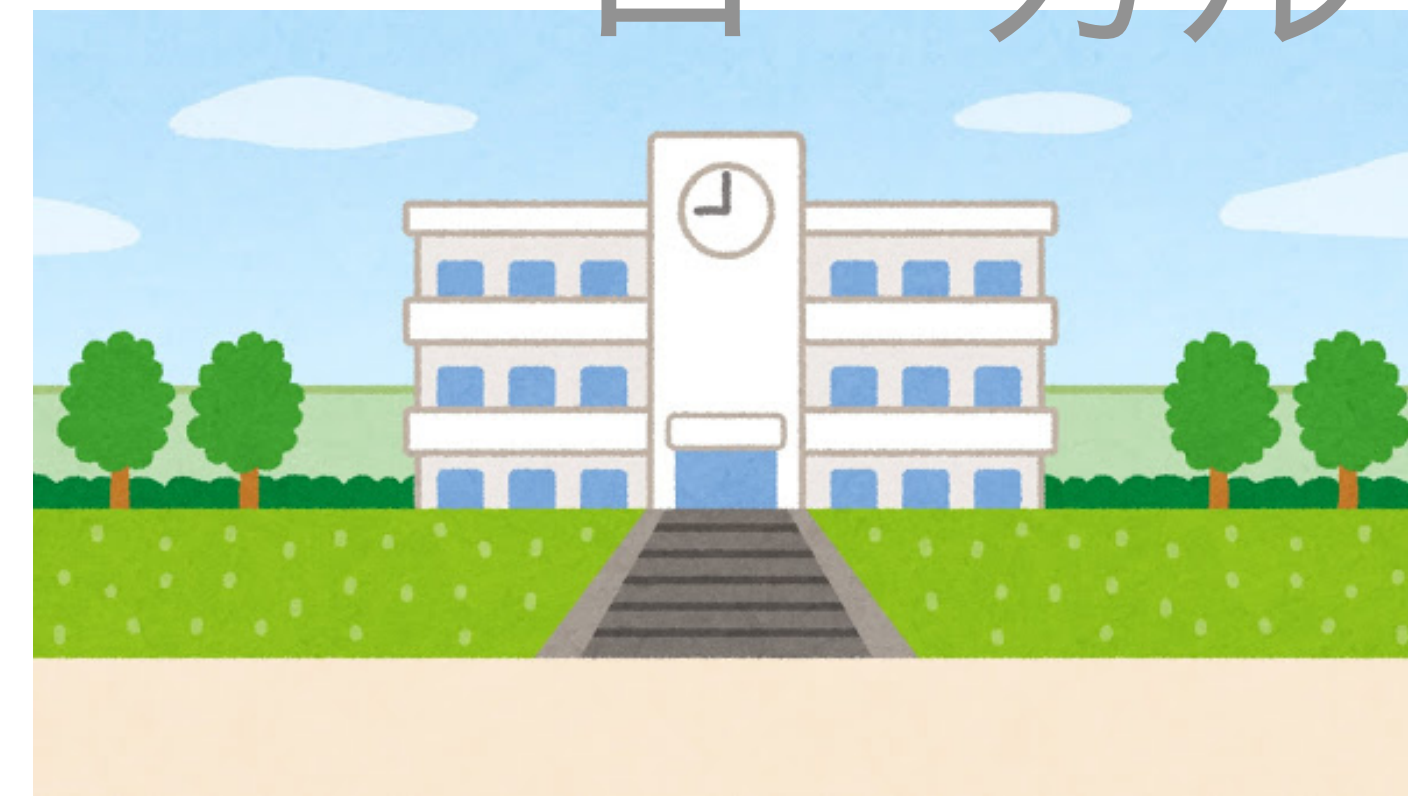


# 変数のスコープ(有効範囲)

グローバル = 全世界



ローカル






# 変数のスコープ(有効範囲)

```
<?php
```

```
$globalVariable = “グローバル”; //関数の外はグローバル  
変数
```

```
function checkScope(){  
    $localVariable = “ローカル”; //関数内はローカル変数  
    return ;  
}
```

呼び出すなら global ではなく 関数の引数を使う



# ファイルの読み込み



# ファイルの読み込み

```
require(); require_once(); //エラー  
include(); include_once(); //警告
```

\_\_DIR\_\_ \_\_FILE\_\_マジック定数

他の方法で名前空間(namespace)もあり

- ・Laravelなどのフレームワークはこちらが主流