

Laravel 超入門

...

滝澤正大 岡田州平

目次

1. Webサイトについて
2. 開発言語
3. フレームワークの導入
4. Laravelについて
5. MVCについて
6. Laravelの機能紹介
7. フロントエンドの実装について

滝澤 正大



出身:群馬

年齢:23歳

iOS/Androidネイティブ開発
TensorFlow/SuperResolution



@Shota16787947



@ShotaTakizawa

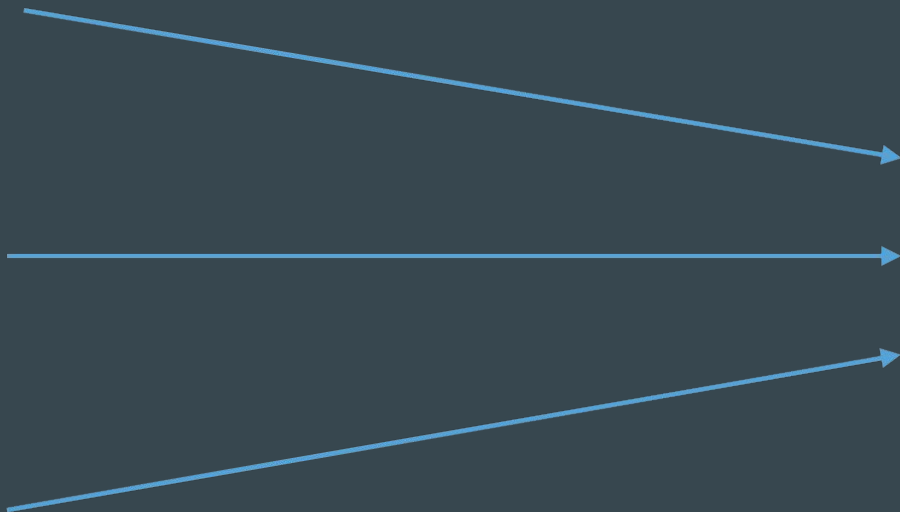


所属:デジタル・ヒュージ・テクノロジー

Webサイト



クライアントサイド

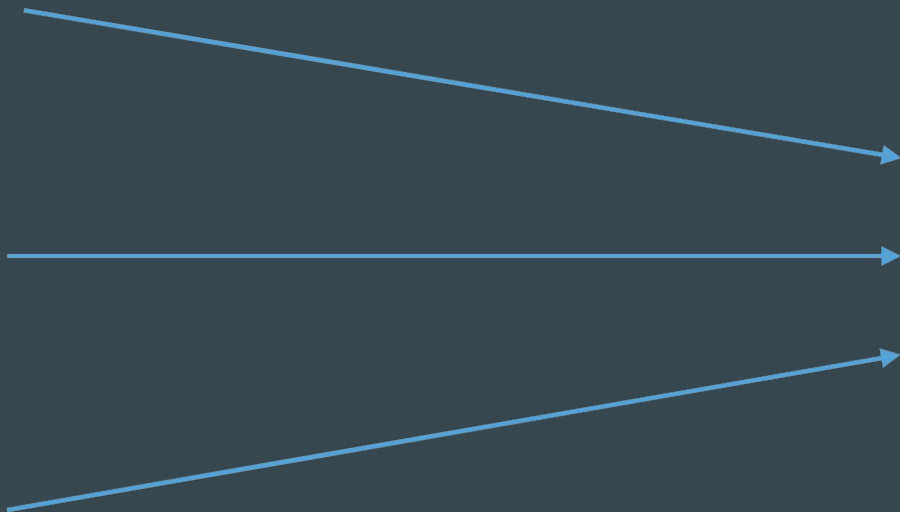


サーバサイド

Webサイト



クライアントサイド



サーバサイド

開発言語

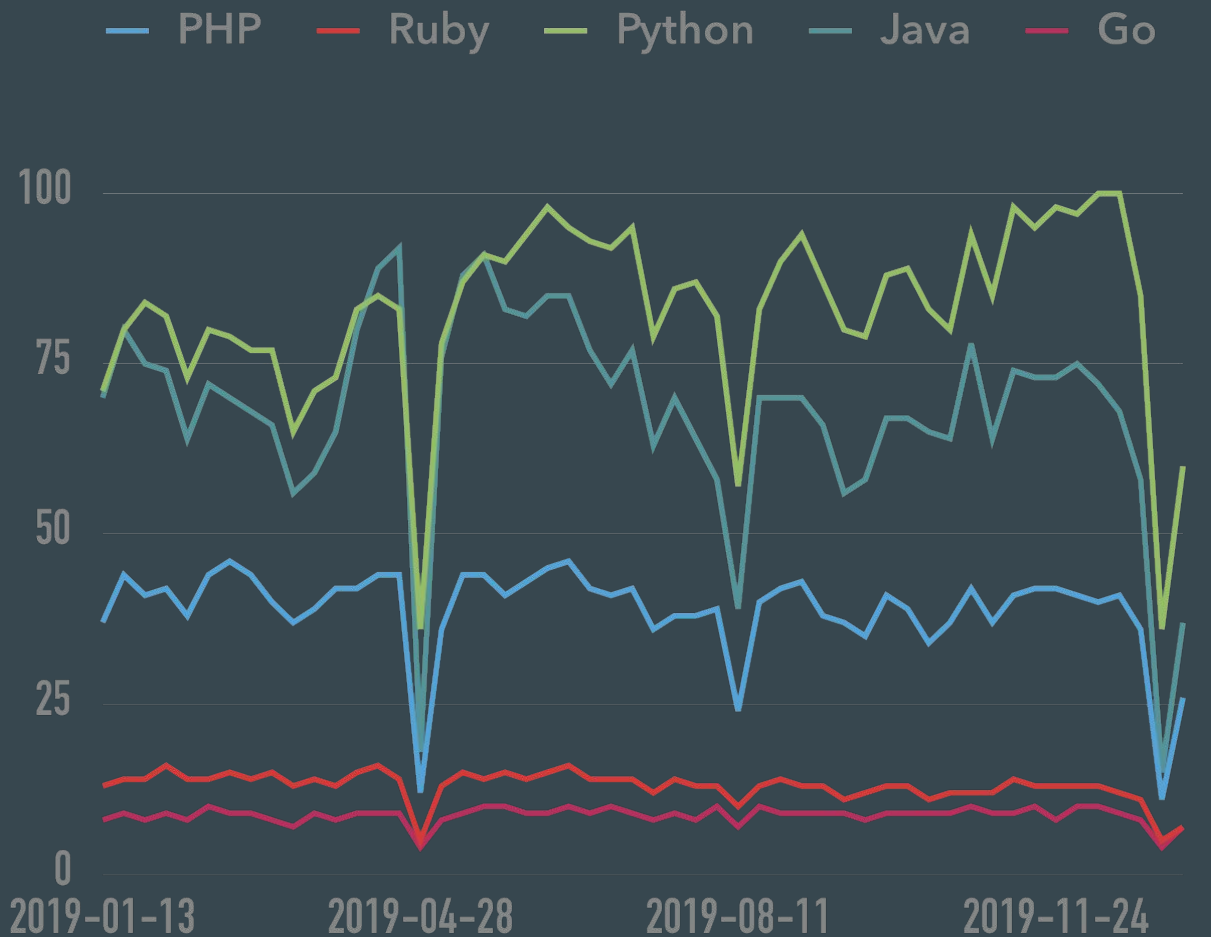
開発言語の選定

・人気の言語

人気の言語は情報量も多いため
躰いた際に検索で引かかる率が高い

・将来性

コミュニティなどが活発で、今後も
開発が進んでいくか



PHPとは

- ・動的なWebページ生成が可能
- ・Webアプリ・サービスを作ることに特化した言語
元は個人の動的Webを実現するためのCGIツールであった.
- ・FacebookやWikipediaもPHPで開発された
- ・WordPressの開発にも使われている
プラグインの作成に使用する.

PHPコード

```
<html>
  <body>
    <p>
      <?php
        $message = 'Hello World';
        print($message);
      ?>
    </p>
  </body>
</html>
```

DBを使う

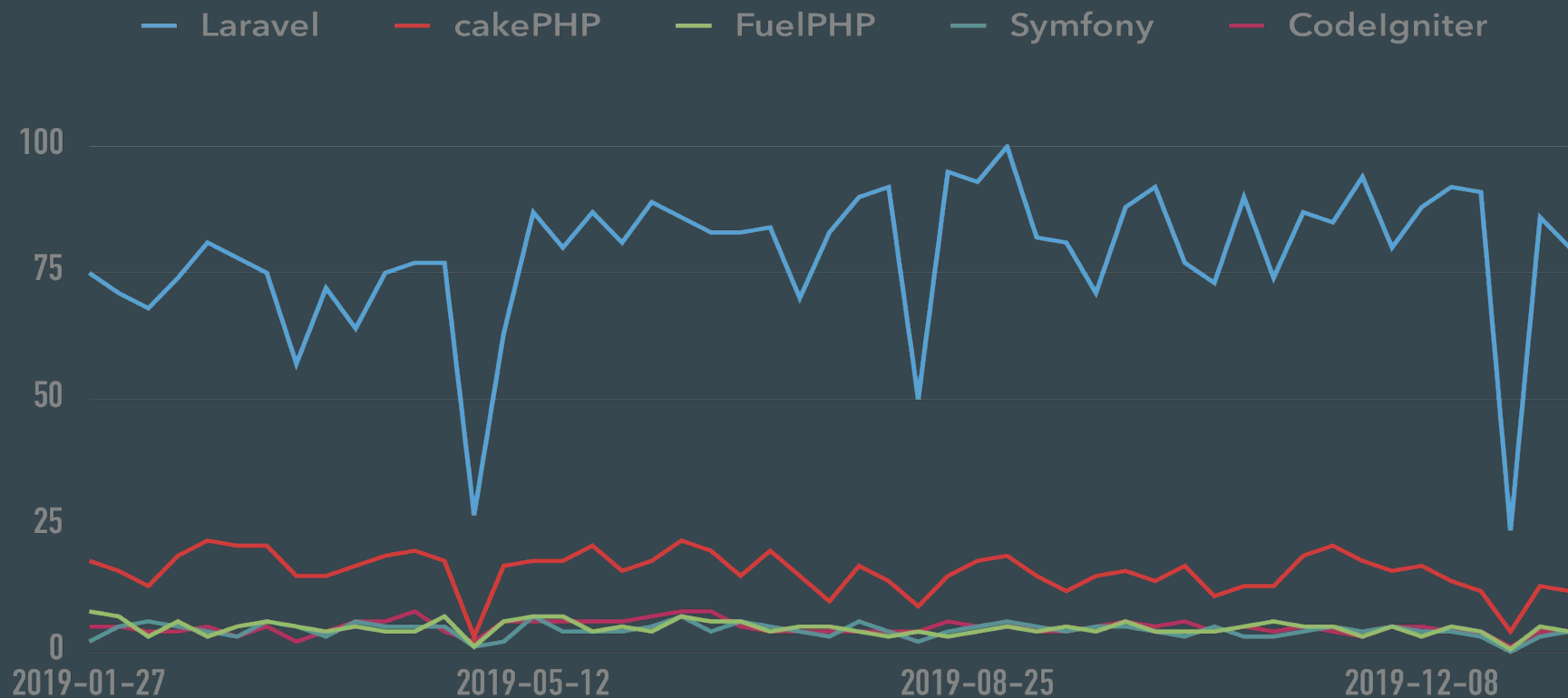
```
$host = 'example.com';  
$dbname = 'example';  
$user = 'xxx';  
$pass = 'xxx';  
  
$db = new PDO("mysql:host={$host};dbname={$dbname}", $user, $pass);  
  
$q = $db->query('SELECT xx FROM yy');  
  
while ($row = $q->fetch()) {  
    print("$row[zz]"); // データを表示  
}
```

開発でぶつかった問題

- ・マークアップと混ざってしまう
最終的なコードが汚くなってしまう.
- ・phpタグをたどるのが億劫
データを用意する部分なのか表示する部分なのかわからなくなる.
- ・バグが発生したときに、該当箇所を探すのが大変
バグの原因が明確でも、どの部分に書いたか探すのが大変.

フレームワークの導入

人気のFW



Laravelの特徴

- ・学習コストが低い

柔軟に記述できるため、雑な設計でも動いてしまうのは注意.

- ・公式ドキュメントの情報が読みやすい

公式の日本語情報があるのはありがたい.

- ・MVCアーキテクチャ

機能別に分けて書けるため、処理が追いやすい.

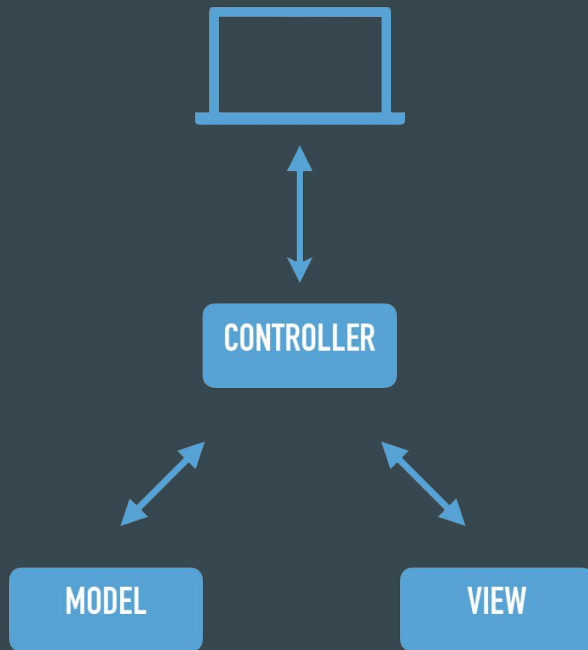
MVC

UIをもつアプリケーションのアーキテクチャ

Model: アプリケーションのデータの管理など

View: ユーザに表示する画面の生成など

Controller: ユーザの入力値をもとに, 使用する
ModelとViewを決定する



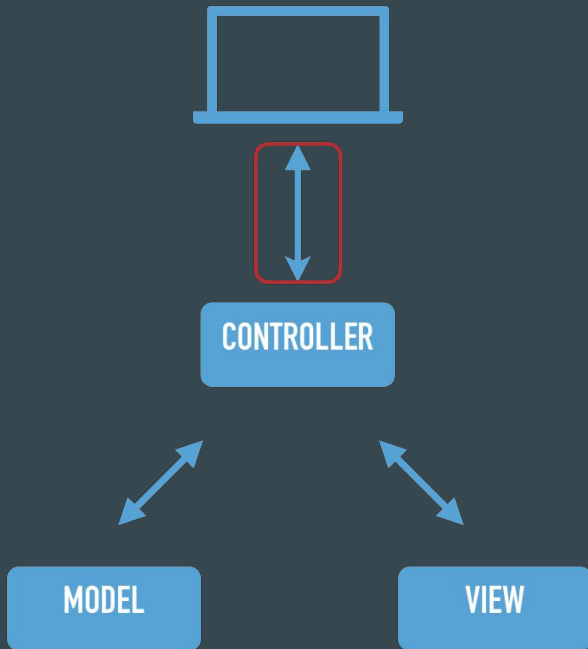
Laravel

Routing

リクエストされたURLとControllerをつなぐ

```
// http://アドレス/foo で呼び出される
Route::get('foo', function ()
{
    return 'Hello World';
});

// コントローラの呼び出し
Route::get('foo', 'FooController@index');
```



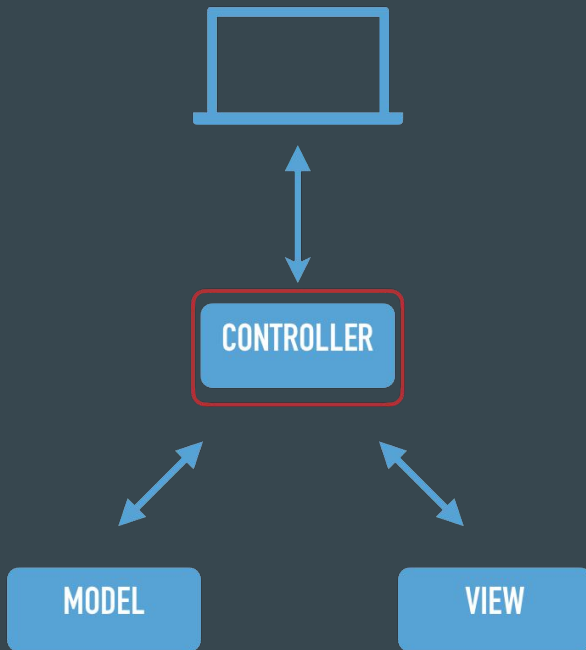
Controller

ModelとViewを用いて表示画面を生成する

```
class ExampleController extends Controller
{
    public function index(Request $request)
    {
        // モデル(データ)の用意
        $model = ExampleModel::all()

        // モデルをもとにビューを作成
        $view = view('example', ['model'=>$model]);

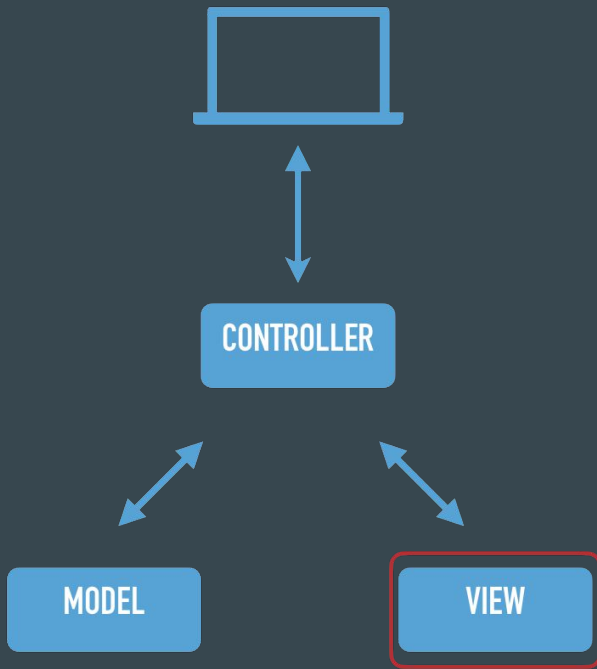
        // 作成したビューをレスポンスとして返す
        return $view;
    }
}
```



View

Bladeというテンプレートエンジンを使用
テンプレートとデータを合わせたものを出力する

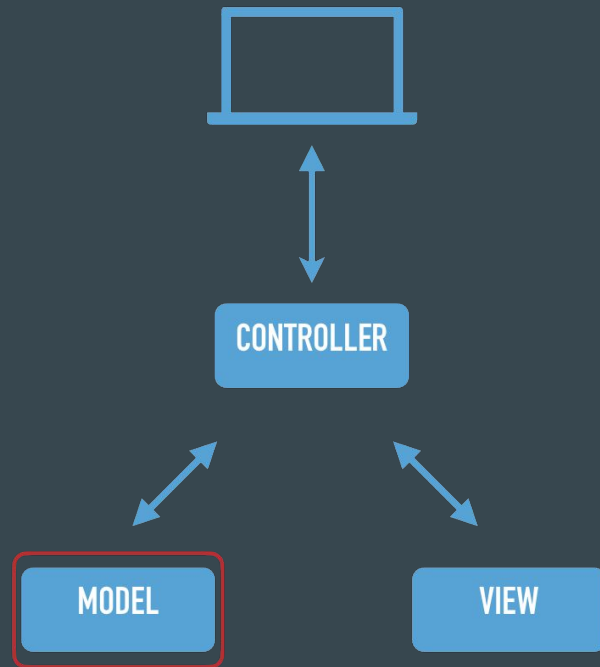
```
<html>
  <body>
    <table>
      {{-- 渡されたデータの表示 --}}
      @foreach ($items as item)
        <tr>
          <td>{{ $item->name }} </td>
          <td>{{ $item->email }} </td>
        </tr>
      @endforeach
    </table>
  </body>
</html>
```



Model

主にデータベースとの連携を行う。

LaravelにはEloquentというORMが実装されている



DBへの接続

DB_CONNECTION=pgsql

DB_HOST=example.com

DB_PORT=5432

DB_DATABASE=xxx

DB_USERNAME=xxx

DB_PASSWORD=xxx

DBへの接続

```
# 空のデータを用意
$data = new Example;

# データを設定
$data->name = 'xxx';

# DBにインサート
$data->save();
```

Laravel

ドキュメント章別ページ

前章	リリースノート	アップグレードガイド	貢献ガイド	APIドキュメント
準備	インストール Valet	設定 デプロイ	ディレクトリ構造	Homestead
構成の概念	ライフサイクル 契約	サービスコンテナ	サービスプロバイダ	ファサード
基礎	ルーティング リクエスト セッション	ミドルウェア レスポンス バリデーション	CSRF保護 ビュー エラー処理	コントローラ URL生成 ログ
フロントエンド	Bladeテンプレート	多言語化	スカフォールド	アセットコンパイラ
セキュリティ	認証 暗号化	API認証 ハッシュ	認可 パスワードリセット	メール確認
より深く知る	Artisanコンソール イベント 通知	ブロードキャスト ファイルストレージ パッケージ開発	キャッシュ ヘルパ キュー	コレクション メール タスクスケジュール
データベース	データベースの準備 シーディング	クエリビルダ Redis	ベジネーション	マイグレーション
Eloquent ORM	Eloquentの準備 APIリソース	リレーション シリアライズ	コレクション	ミューテタ
テスト	テストの準備 データベース	HTTPテスト モック	コンソールテスト	ブラウザテスト
公式パッケージ	Cashier Passport	Dusk Scout	Envoy Socialite	Horizon Telescope
言語 ファイル	auth.php	pagination.php	passwords.php	validation.php

<https://readouble.comより>

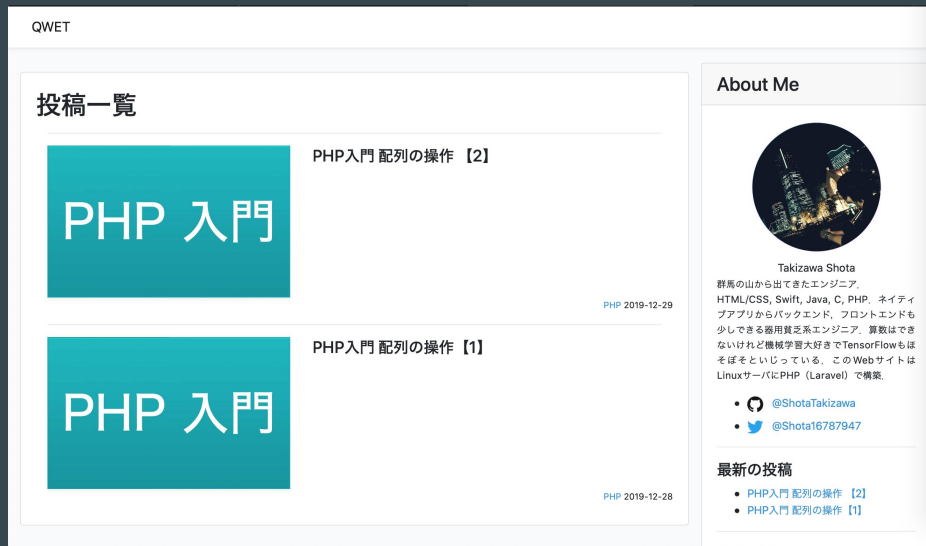
Webサイト

Laravel × PostgreSQLで作成

メモリ 1GBのサーバでも(今の所)
快適に動作

~~作って満足して記事が書けていない~~

<https://qwet.info>



フロントエンド

自己紹介

名前:岡田州平

経歴:2年間教育業界に携わる。
エンジニアとして半年経過

趣味:本を読むこと

直近の目標:一人で稼ぐ力を身につけること

所属:デジタル・ヒュージ・テクノロジー

ブログ:<https://kusanagi.dht-jpn.co.jp/2020/01/okada-shuheii20200111/>



DHT
Digital Huge Technology

フロントエンド開発の複雑化

- ・1991年にWebが登場

CSSの装飾, 静的なページが多い.

- ・1995年にJavaScript (JS) が開発される

動的なページが作成できるようになる.

JSの実装に絡んだブラウザでセキュリティーホールが見つかる.

⇒ フロントエンドに求められるもの

HTML, CSS, JSを使った見た目の補助

フロントエンド開発の複雑化

- ・2005年にAjax (Asynchronous Javascript + XML) の登場
JSを使って、非同期にサーバからXMLデータを取得.
 - ・2009年にNode.jsの登場
サーバサイドJSの実行環境.
 - ・2010年にJQueryの登場
JSのライブラリ. プラグイン機能が備えてあり, プラグインの開発も可能.
- ⇒ フロントエンドに求められるもの
Ajaxをベースとした開発.

フロントエンドの複雑化

- ・フレームワークの登場

設計の構造化が考慮されるようになった.

- ・双方向バインディングの流行

データとViewを自動で更新してくれる。

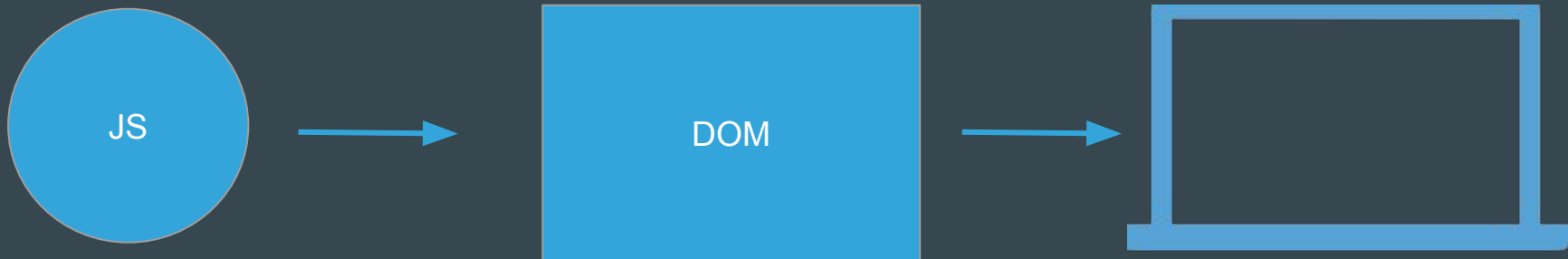
- ・仮想DOMの流行

DOMの書き換え時に, 変更部分のみ書き換えることにより高速化する.

⇒ フロントエンドに求められるもの

アプリケーションのプレゼンテーション全般

仮想DOM



//idの取得

```
var h1 = document.getElementById("title")
```

//取得したidに変更をくわえている

```
h1.textContent = "タイトル変更"
```

//タイトルにidを追加

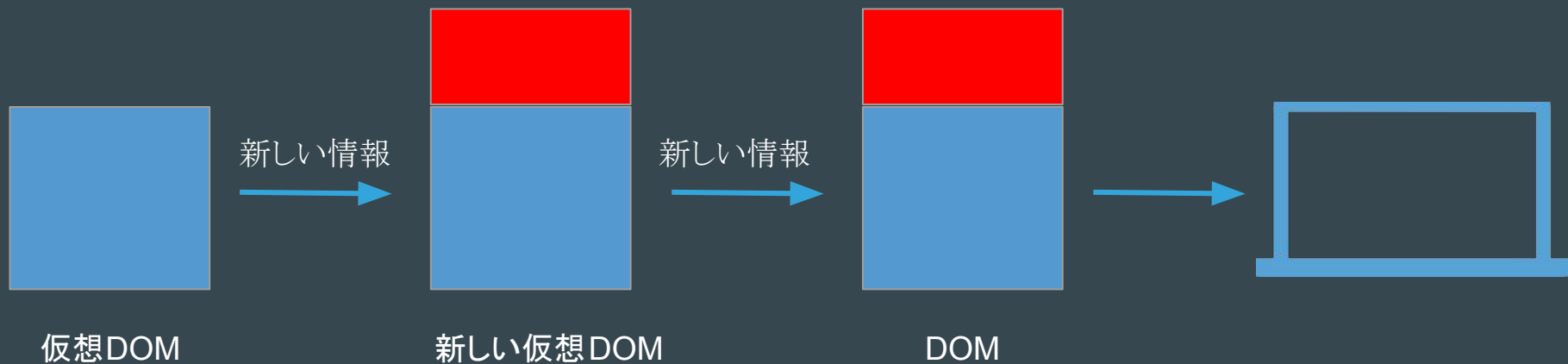
```
<h1 id = "title">タイトル</h1>
```

//Javascriptを反映後

```
<h1 id = "title">タイトル変更</h1>
```

仮想DOM

必要な部分のみDOMに反映させるから高速化する。



Vue.jsの特徴

- Javascriptのフレームワーク

- Webアプリケーションを簡単に作ることができる

- 現在のWebアプリケーション開発の流行にのっている

- 双方向バインディング

- 仮想DOM

Vue.jsのセットアップ

laravel/uiをインストール

\$ composer require laravel/ui --dev

Vueのセットアップ

\$ php artisan ui vue

npm経由で開発環境の構築

\$ npm install && npm run dev

※ #: 説明 \$: 入力コマンド