

サイバー大学IT総合学部

専門応用科目

JavaScriptフレームワークによるWebプログラミング

# 第9回 Vue.js概要

小園井康志

# 第9回 学習目標

- JavaScriptフロントエンドフレームワーク  
Vue.jsの概要を理解し説明できる
- Vue.jsのインストール方法を理解し説明できる
- Vue.jsを使った簡単なプログラムを作成できる

# 第9回 授業構成

- 第1章 Vue.js概要
- 第2章 Vue.js実行環境
- 第3章 Vue.js実習

JavaScriptフレームワークによるWebプログラミング  
第9回 Vue.js概要

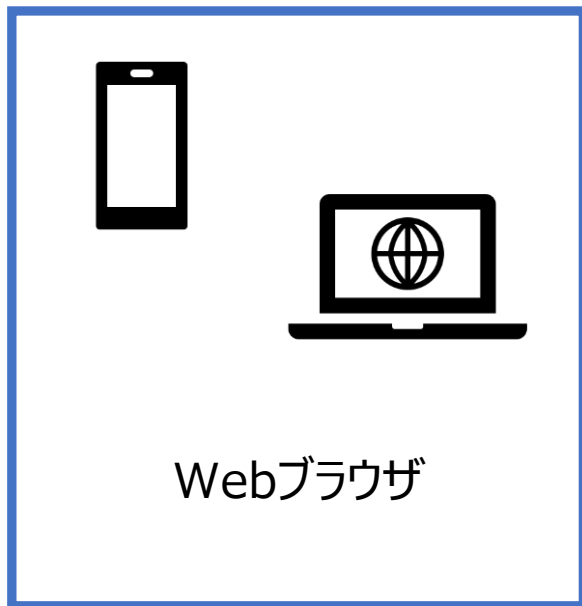
# 第1章 Vue.js概要

# 第1章 学習目標

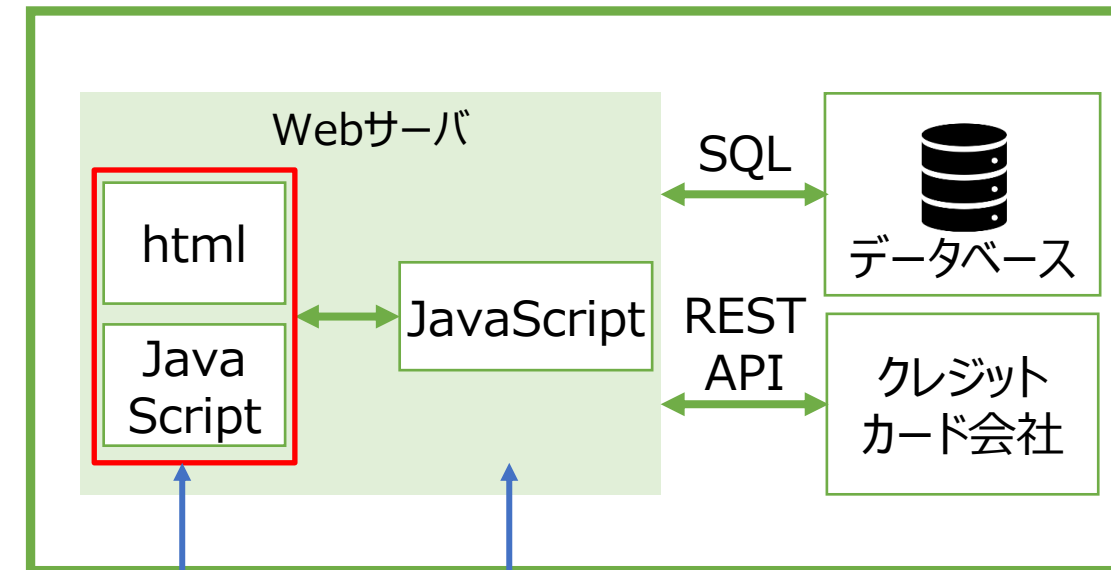
- JavaScriptフロントエンドフレームワーク  
Vue.jsの概要を理解し説明できる

# JavaScriptフレームワーク授業での全体図

端末（クライアント）



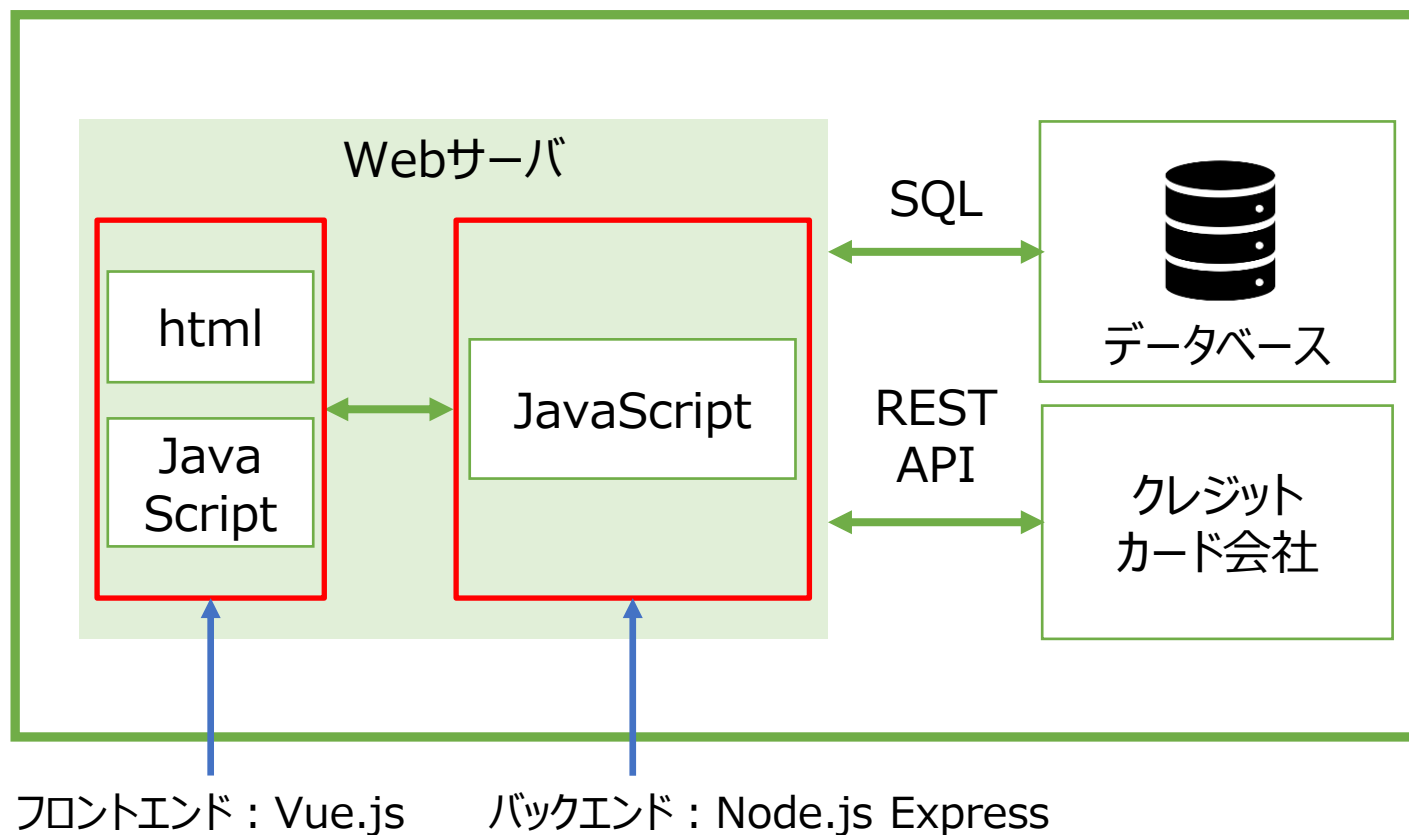
Webシステム



フロントエンド : Vue.js    バックエンド : Node.js Express

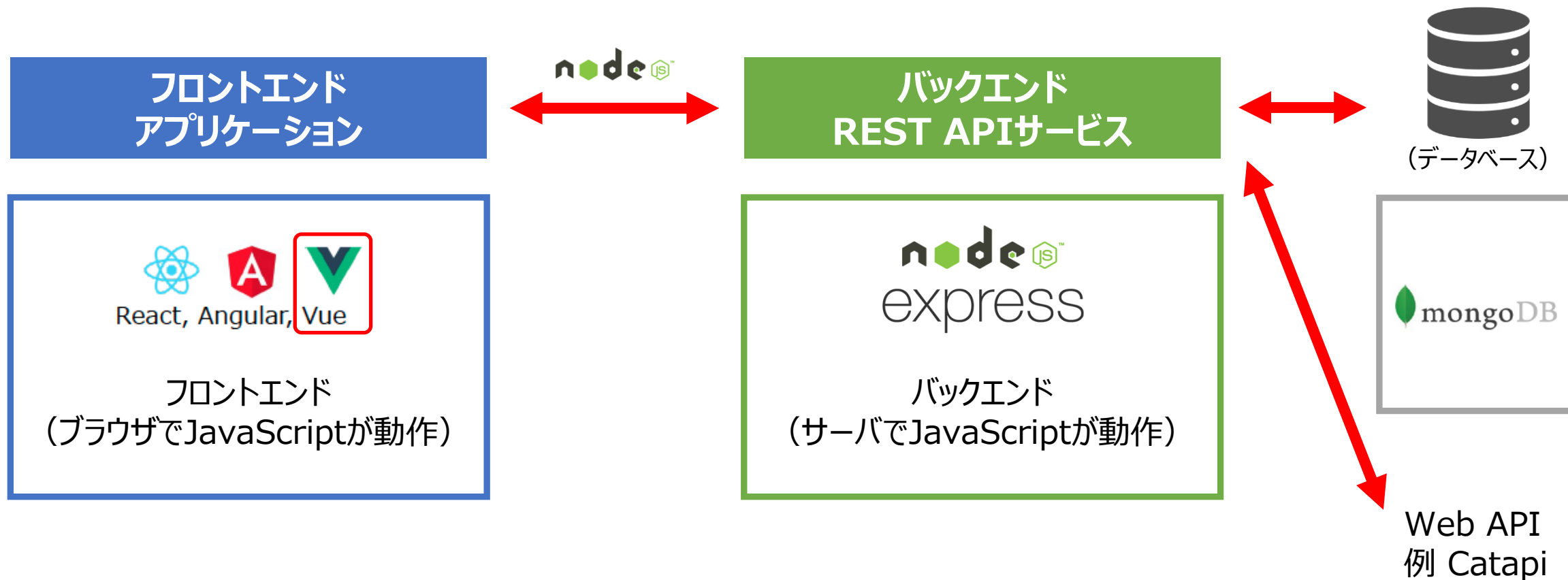
# JavaScriptフレームワーク授業での全体図

- バックエンド：前回、第8回まで
- フロントエンド：今回の9回以降



# JavaScriptフレームワーク授業での全体図

- ・フロントエンドのフレームワークとしてVue.jsを学習





# フロントエンドJavaScriptフレームワーク

フロントエンドの開発においては言語を使ってすべて0から開発するフルスクラッチ開発ではなく、フレームワークを使った開発が増えてきている。

フロントエンドのJavaScriptフレームワークの中でも多く使われているのが以下のフレームワーク

- **Angular (アングラー)**

Googleが開発したJavaScriptフレームワーク。商用サービスで多く使われている。学習コストは高い。

- **React (リアクト)**

Facebookが中心で開発ボタンやボックスなどユーザーインターフェースの開発に特化している。SPAが作りやすい。

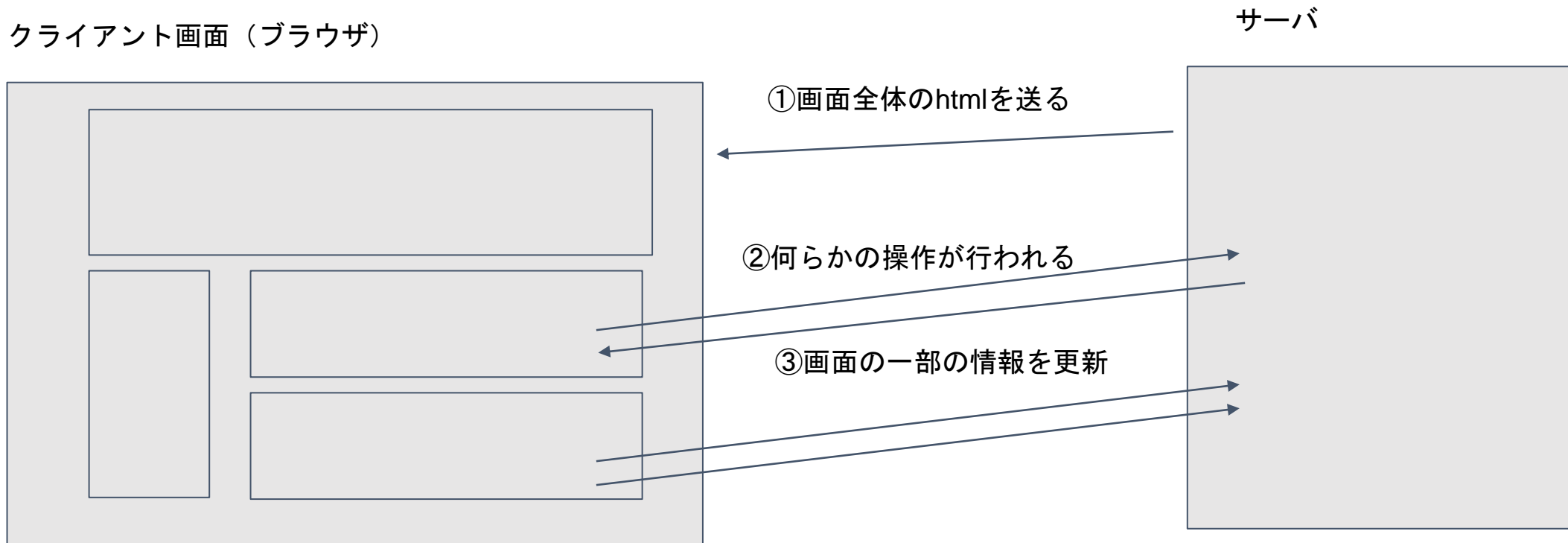
- **Vue (ビュー)**

同じくユーザーインターフェースに特化。必要なものが一通り揃っている。学習コストが比較的低いフレームワーク。

SPA: シングルページアプリケーション  
Webアプリケーションの構成法の一つで、  
ページの移動を行わず、一つの同じ  
Webページ上でアプリが動く方式。

# SPA(シングルページアプリケーション)

従来のウェブサイトではユーザの操作などですべての画面を変更するがSPAでは一部のみを更新



# Vue.js 特徴

- 学習コストが低い
  - 導入も比較的簡単
- 拡張性がある。
- 高性能、リアクティブ
- 開発で必要な機能がセットになっている。
- 様々な機能が用意されているため、操作を覚えると扱いやすく少ない記述で実装が可能
- 日本語の公式サイトもあるのでわかりやすい  
<https://jp.vuejs.org>

# 学習コストが低い

- ドキュメントが豊富

- 公式ドキュメント (<https://jp.vuejs.org/>)
- 世界中の開発者によるアクティブなコミュニティがあり、質問や課題の解決に役立つ情報やリソースが多数存在している。
- コミュニティガイド：  
<https://ja.vuejs.org/about/community-guide.html>

- インストールが簡単

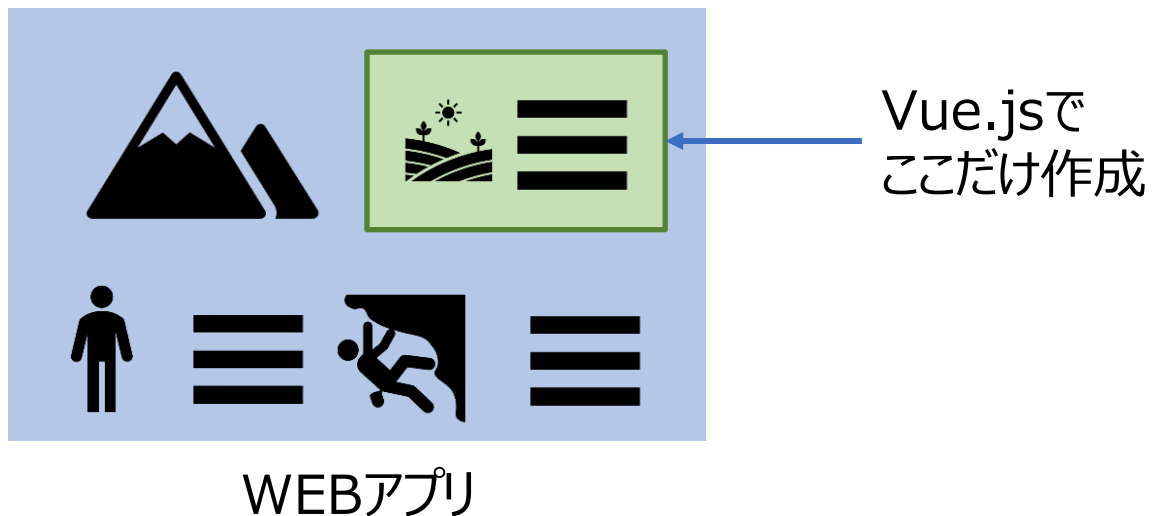
- CDNでファイルを指定して読み込むだけ

- 参照サイト

<https://coosy.co.jp/blog/vuejs-benefit/>

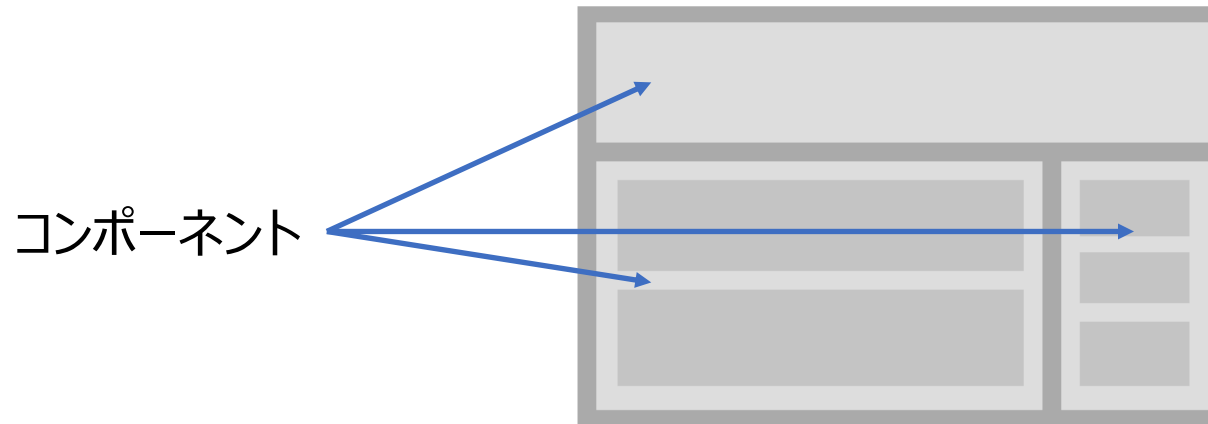
# 拡張性の高さ

- シンプルな設計になっており他のライブラリーと組み合わせることも可能。
- 既存プロジェクトの一部に適用することも可能なため部分的にVue.jsを使うことも可能



# コンポーネント

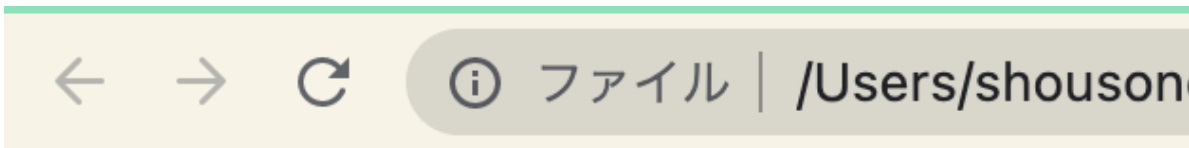
- コンポーネントは部品、構成要素と言った意味。  
プログラムを再利用可能な部品として作っておいて  
それらを配置すれば必要なアプリが素早く作成できる。



# リアクティブ

- リアクティブは“反応する”という意味。  
プログラムの中のデータが更新されると自動的に  
そのデータを利用しているアプリの画面の表示も更新される。  
元データを更新するとそれに反応して表示が変わる。

例：タイマの時間が変わると（毎秒）表示が変わる



現在の日時: 2023/4/5 13:32:45

# リアクティブ

- 右記のコードで  
time.htmlというファイルを作成、  
ブラウザで実行すると確認できる

コードはこちらを参照  
<https://github.com/osonoi-hot/vue>

```
<script src "https://unpkg.com/vue@next"></script>
<div id "app">
  <span>現在の日時:
  {{currentDateTime.toLocaleString()}}</span>
</div>
```

```
<script>
Vue createApp
  data function
    return
      currentDateTime new Date

  created function
    let self this
    this timer setInterval function
      self currentDateTime new Date
      1000

  beforeUnmount function
    clearInterval this timer

  mount '#app'
</script>
```



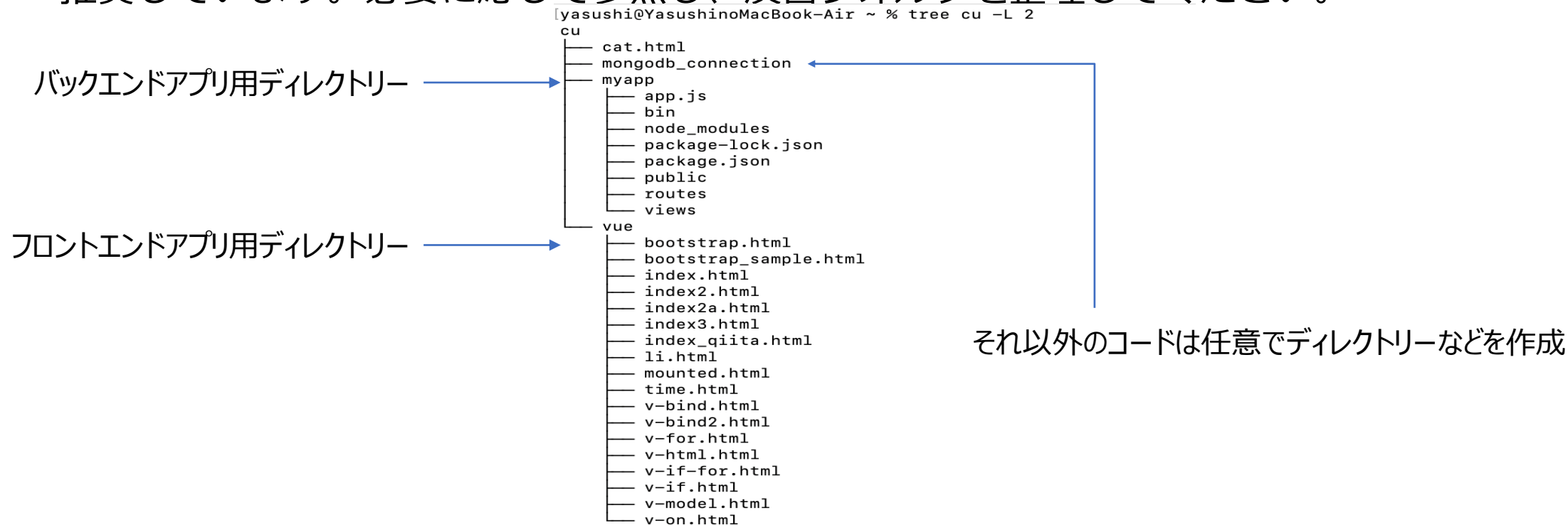
# 実機演習：time.htmlの実行例

動画を全画面で視聴してください

# 実習時間

- ・ 10分程度を目安に動画を止めて前ページまでの実習をしてください。
- ・ 作業が終わったらビデオを再開して学習を進めてください。

※第1回4章で説明した通り、実習のファイル作成時には以下のディレクトリ構造を推奨しています。必要に応じて参照し、演習フォルダを整理してください。



# Vue.jsのバージョン

- この授業ではVue3を使用
- Vue3とVue2は動作が異なる
- Vue2は2023年末でサポートが終了予定
- <https://v3-migration.vuejs.org/ja/>

## Vue 3 移行ガイド

**Vue 2 のサポートは 2023 年 12 月 31 日をもって終了します。**

EOL の日付までに Vue 3 へのアップグレードが不可能な場合は、**Extended LTS** の詳細をご覧ください。

このガイドは主に、Vue 2 の経験があり、Vue 3 との変更点について学びたいユーザーのためのものです。Vue 3 を試す前に最初から最後まで読まなければならないものではありません。Vue 3 を学ぶには、**新しいドキュメント**を読むのがおすすめです。

# テンプレートとディレクティブ

- テンプレート

- Htmlをベースとするテンプレート構文が用意されており、それを利用することでさまざまな表示をhtmlの中に組み込むことができる。

- ディレクティブ

- Vue.jsが提供している特別な属性
- 日本語ではv属性と呼ばれている
- 値などが変わるとリアクティブに表示を変えるなどの動作をすることができる

# 第1章 まとめ

- Vue.js概要では以下を学習した。
- Vue.jsの特徴
  - 拡張性
  - コンポーネント
  - リアクティブ
  - Vue.jsのバージョン
  - テンプレートとディレクティブ（詳細は第10回以降）

JavaScriptフレームワークによるWebプログラミング  
第9回 Vue.js概要

# 第1章 Vue.js概要

終わり

JavaScriptフレームワークによるWebプログラミング  
第9回 Vue.js概要

## 第2章 Vue.js環境

## 第2章 学習目標

- Vue.jsのインストール方法を理解し説明できる
- 開発者ツールの設定を理解し説明できる



# Vue.jsのインストール

- インストールの方法はいくつかある
  - CDN (Content Delivery Network) からhtmlファイルに読み込み使う
  - Vue.jsのファイルをダウンロードしてhtmlファイルでそのファイルを指定して使う
  - npmを利用してインストール
  - Vue CLIを使ってプロジェクトを作る
- 今回の実習では一番シンプルなCDNを利用する方法を学習する

CDN : Content Delivery Network (コンテンツ デリバリー ネットワーク) 、インターネット上で公開されるWebサイトでソフトウェアなどのコンテンツのデリバリー、配信のために利用されるサービス

# Vue.jsのインストール


- CDNで位置を指定して読み込む

htmlファイルのscriptに以下の内容を入れます

```
<script src=https://unpkg.com/vue@3.2.0/dist/vue.global.js></script>
```



CDNの場所



Vue.jsのバージョン

GitHubリポジトリでアルファ版も含めたバージョン情報を確認できます。  
<https://github.com/vuejs/core/blob/main/CHANGELOG.md>

# Vue.jsのインストール

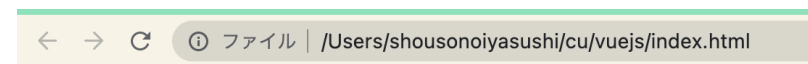
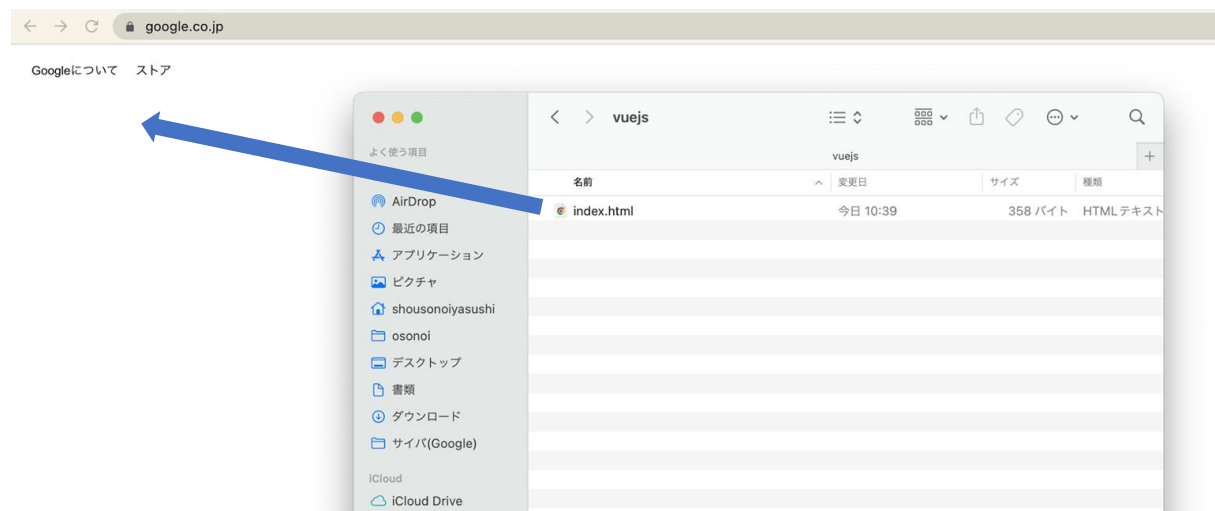
- 読み込みができるか試す
- 以下のプログラムをindex.htmlとして作成
- ブラウザで表示

```
<!DOCTYPE html>
<html>
  <head>
    <title>Vue.js のサンプル</title>
    <script
src="https://unpkg.com/vue@3.0.0/dist/vue.global.js"></script>
  </head>
  <body>
    <h1>Vue3</h1>
  </body>
</html>
```

コードはこちらを使ってください <https://github.com/osonoi-hot/vue>

# Vue.jsのインストール

- 作成したファイルをドラッグ&ドロップでブラウザに移動



**Vue3**

ブラウザで上のように表示される

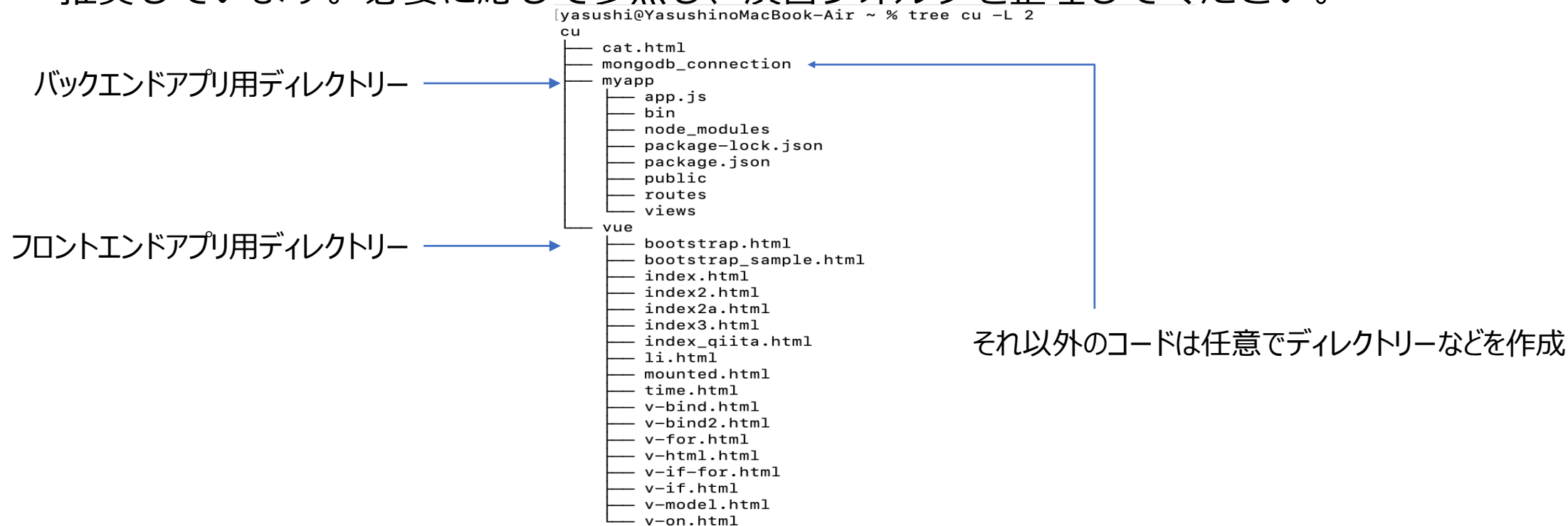
# 実機演習：Vue.jsのインストール

動画を全画面で視聴してください

# 実習時間

- ・ 10分程度を目安に動画を止めて前ページまでの実習をしてください。
- ・ 作業が終わったらビデオを再開して学習を進めてください。

※第1回4章で説明した通り、実習のファイル作成時には以下のディレクトリ構造を推奨しています。必要に応じて参照し、演習フォルダを整理してください。



# Vue.jsサンプルコード2

- Vue.jsを利用したコードを作成
- Index2.htmlというファイル名で作成

コードはこちらを使ってください

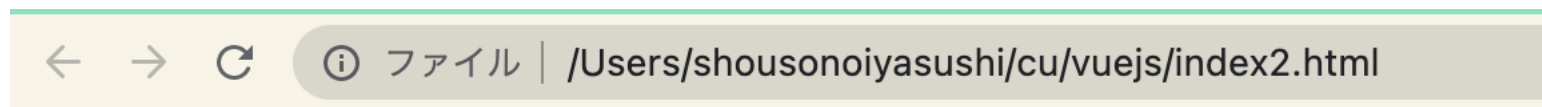
<https://github.com/osonoi-hot/vue>

```
<!DOCTYPE html>
<html>
  <head>
    <title>Vue.js のサンプル</title>
    <script src="https://unpkg.com/vue@3.0.0/dist/vue.global.js"></script>
  </head>
  <body>
    <h1>Vue3</h1>
    <div id="app">
      {{ message }}
    </div>
    <script>
      const appdata = {
        data() {
          return {
            message: "Hello Vue"
          }
        }
      }

      Vue.createApp(appdata).mount('#app')
    </script>
  </body>
</html>
```

# Vue.jsサンプルコード

- 動作結果
- コードは2章で学習



## Vue3

Hello Vue



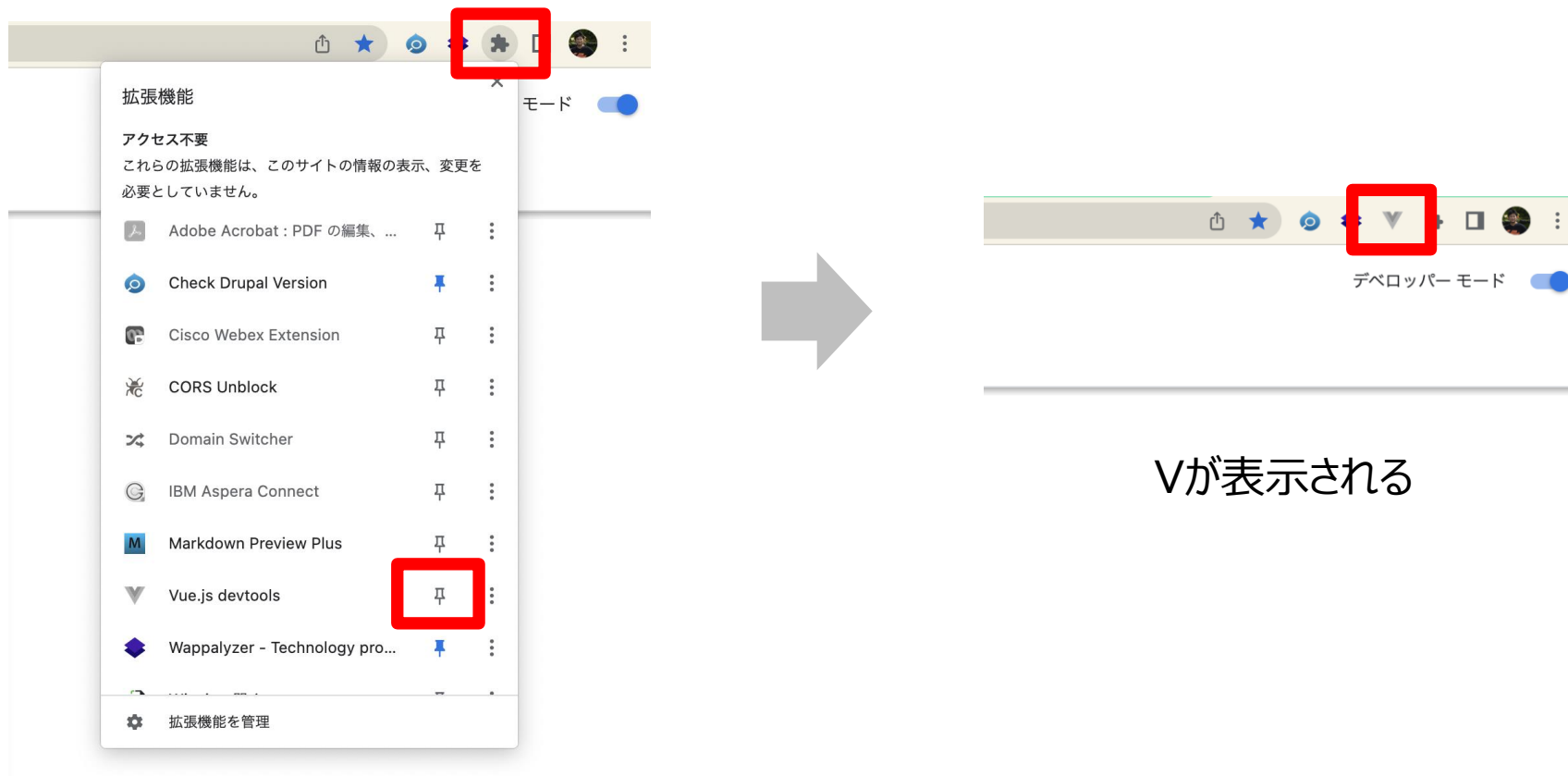
# 開発ツール、Devtoolsのインストール

- DevtoolsはVue.jsの開発をサポートするChromeブラウザの拡張機能。Vue.jsのプログラムの中で使用されているdataの中身を確認することなど開発に便利なツール
- Chromeウェブストアからインストールする
  - <https://chrome.google.com/webstore/detail/vuejs-devtools/nhdogjmejiglipccpnnnanhbledajbpd>



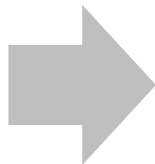
# 開発ツール、Devtoolsのインストール

- ブラウザーの右上の設定メニューをクリックVue.jsを“固定”にする



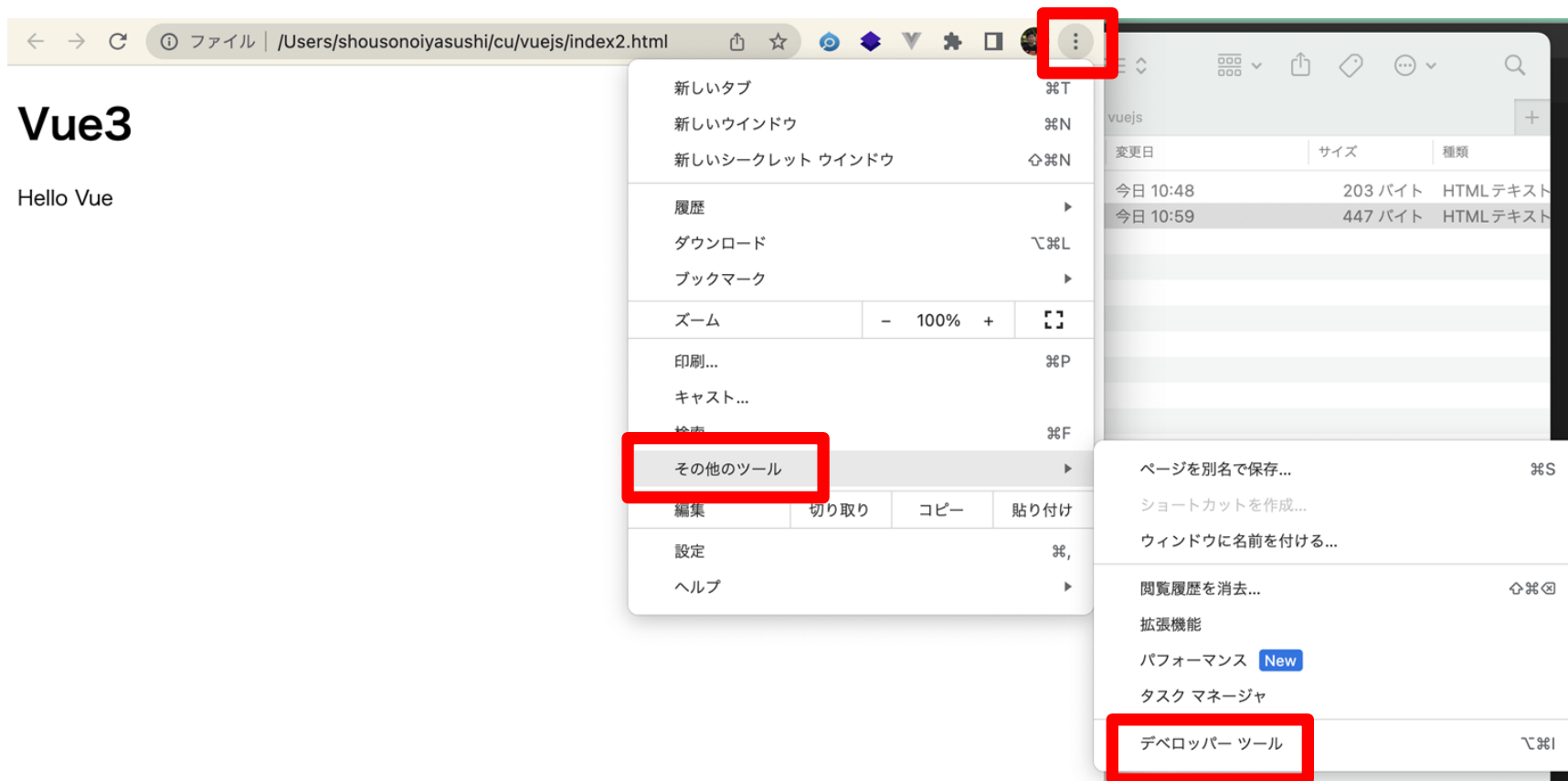
# 開発ツール、Devtoolsのインストール

- Vを右クリック（macはcontrolキーを押しながらクリック）、ファイルのURLへのアクセスを許可するをオンにする



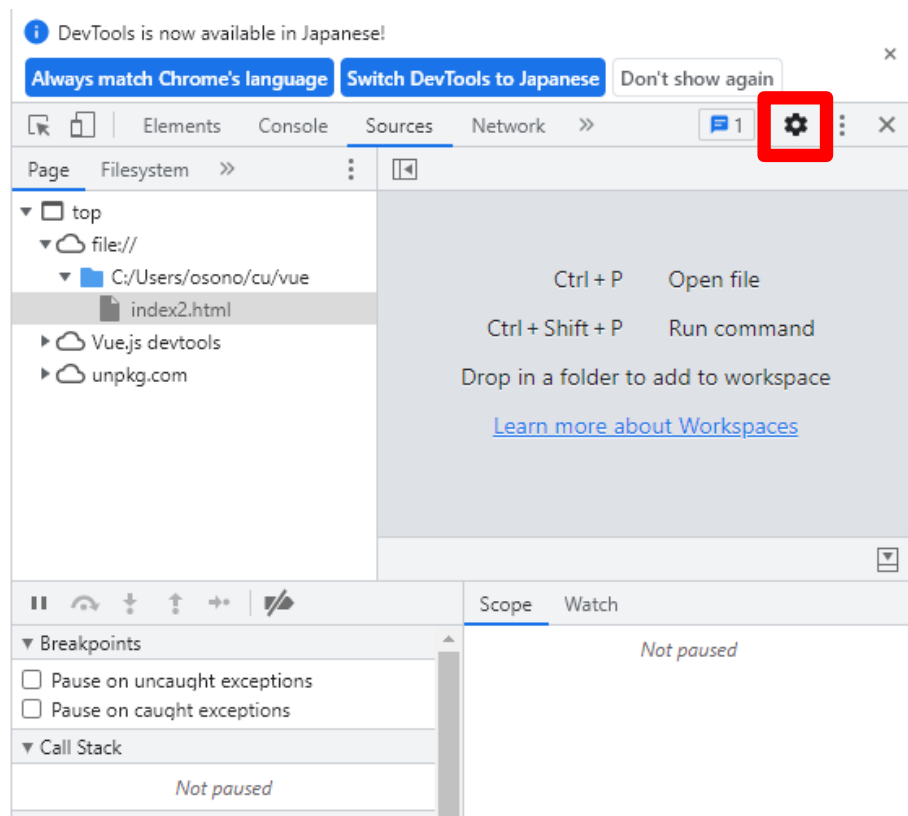
# 開発ツール、Devtoolsの設定

- デベロッパーツールを表示

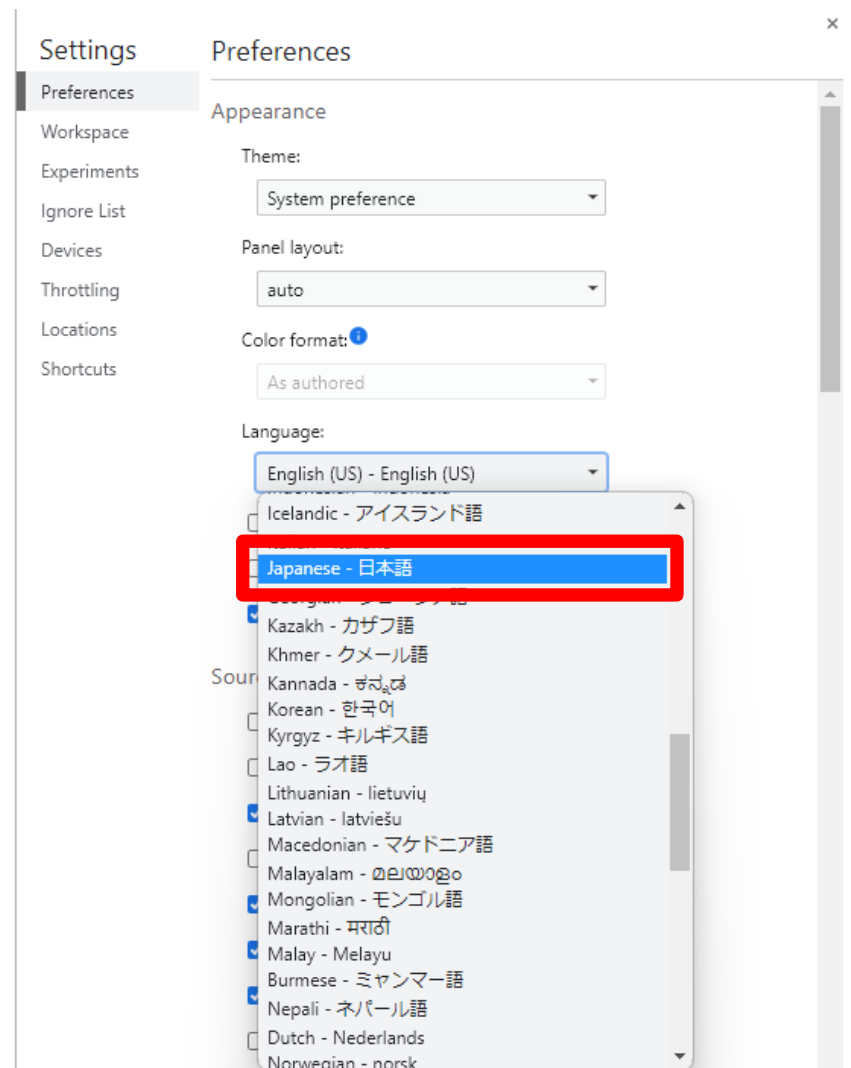
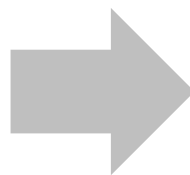


# 開発ツール、Devtoolsの設定

## • 言語の変更（日本語）



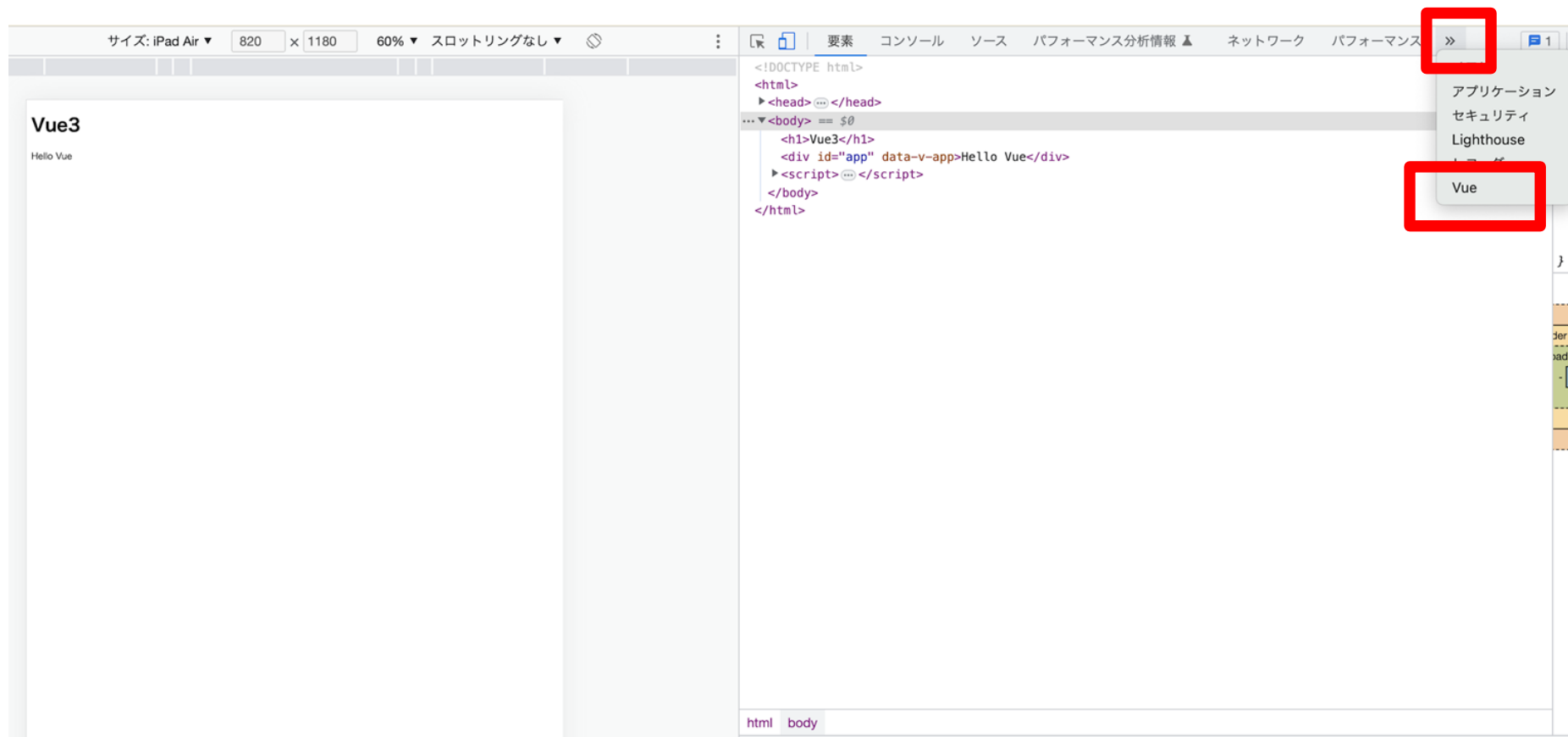
開発者ツールで歯車アイコンをクリック



Preference、Languageで日本語を選択

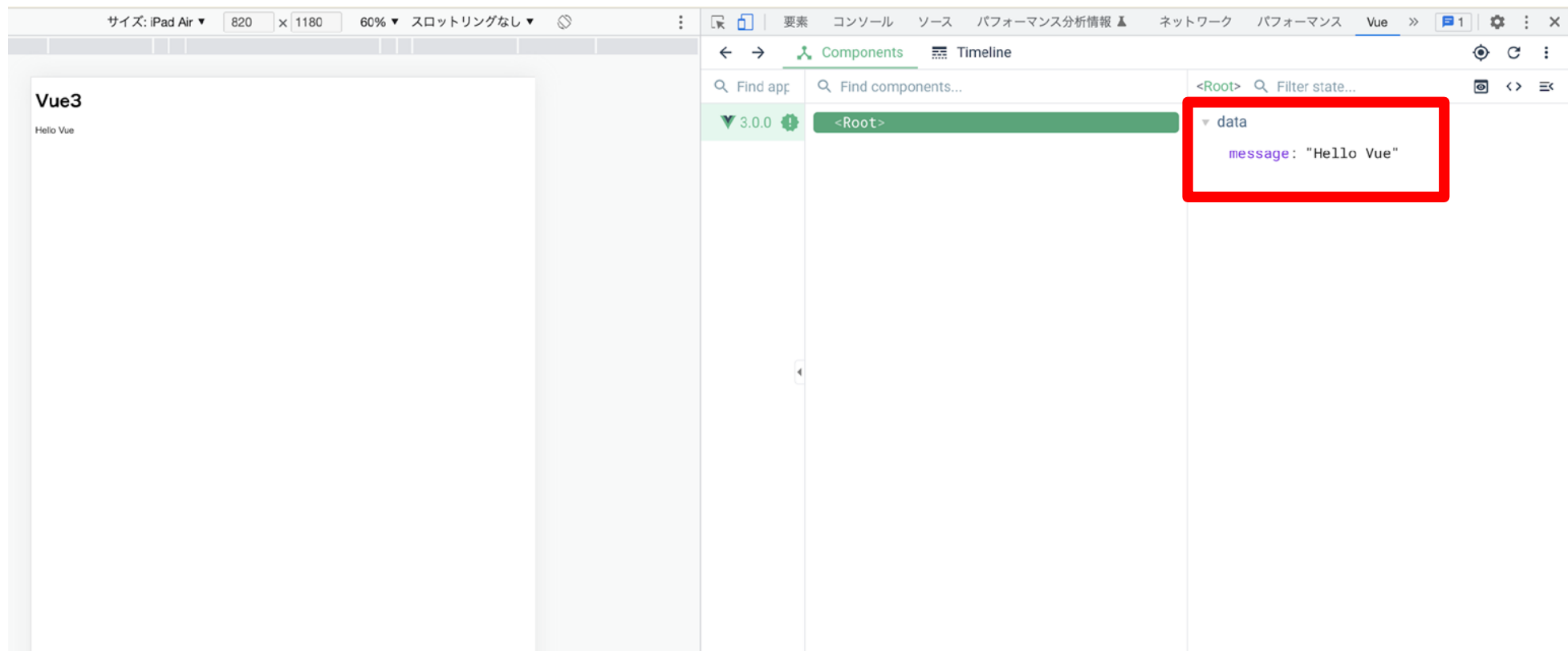
# 開発ツール、Devtoolsの設定

- Vueを表示



# 開発ツール、Devtoolsの設定

- VueのDataが表示される



# 実機演習：開発ツール、Devtoolsの設定

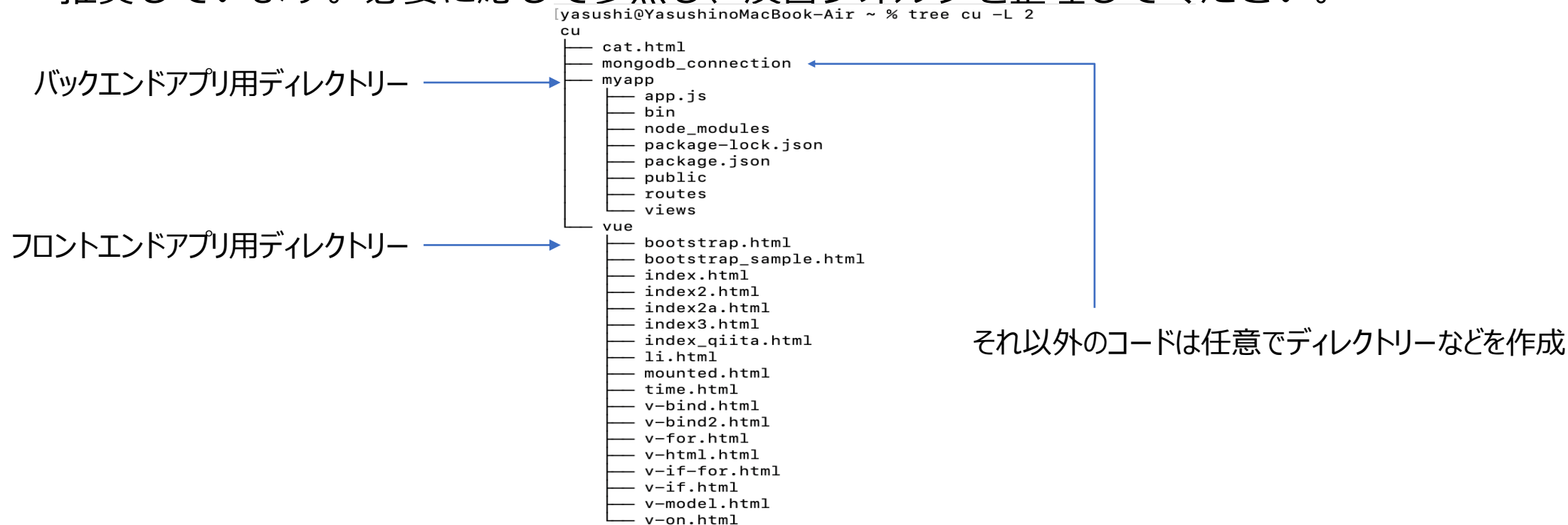
動画を全画面で視聴してください



# 実習時間

- ・ 10分程度を目安に動画を止めて前ページまでの実習をしてください。
- ・ 作業が終わったらビデオを再開して学習を進めてください。

※第1回4章で説明した通り、実習のファイル作成時には以下のディレクトリ構造を推奨しています。必要に応じて参照し、演習フォルダを整理してください。



## 第2章 まとめ

- Vue.jsのインストール方法を理解し説明できる
  - サンプルコードを作成した
- 開発者ツールの設定を理解し説明できる
  - ツールをインストール、設定してプログラムの動作を確認した

JavaScriptフレームワークによるWebプログラミング  
第9回 Vue.js概要

## 第2章 Vue.js環境

終わり

JavaScriptフレームワークによるWebプログラミング  
第9回 Vue.js概要

## 第3章 Vue.js実習

# 第3章 学習目標

- Vue.jsを利用したプログラムを理解し作成できる

# プログラム解説

- 第2章のコード  
(index2.html)

```
<!DOCTYPE html>
<html>
  <head>
    <title>Vue.js のサンプル</title>
    <script src="https://unpkg.com/vue@3.0.0/dist/vue.global.js"></script>
  </head>
  <body>
    <h1>Vue3</h1>
    <div id="app">
      {{ message }}
    </div>
    <script>
      const appdata = {
        data() {
          return {
            message: "Hello Vue"
          }
        }
      }

      Vue.createApp(appdata).mount('#app')
    </script>
  </body>
</html>
```

# プログラム解説

```
<!DOCTYPE html>
<html>
  <head>
    <title>Vue.js のサンプル</title>
    <script src="https://unpkg.com/vue@3.0.0/dist/vue.global.js"></script>
  </head>
  <body>
    <h1>Vue3</h1>
    <div id="app">
      {{ message }}
    </div>
    <script>
      const appdata = {
        data() {
          return {
            message: "Hello Vue"
          }
        }
      }

      Vue.createApp(appdata).mount('#app')
    </script>
  </body>
</html>
```

Vue.jsの読み込み



表示のためのタグ、テンプレート  
(node.jsで使ったテンプレートと同じ役割)



Vue3のスクリプト



htmlは2つの部分に分かれている

# プログラム解説

- Vue.jsの基本形

```
<div id="app">  
  </div>
```

テンプレート表示

```
<script>
```

```
  Vue.createApp(appdata).mount('#app')
```

```
</script>
```

JavaScriptデータを準備

Id="#app"で指定したエレメントに  
スクリプトをマウント、連携する



# プログラム解説

## • Vue.jsの基本形

```
const appdata = {  
  data() {  
    return {  
      message: "Hello Vue"  
    }  
  }  
}
```

Const appdata = { \*\*\*\*\* }

でappdataにデータ、オブジェクトを格納  
\*\*\*\*\*はデータ、オブジェクト

Data関数はreturnの中身を返す

# Vue.js プログラム実習

- 以下のオブジェクトを出力するプログラムを考えてみる

```
message: 'Hello Vue',  
count: 77,  
user: {  
  lastName: 'Osonoi',  
  firstName: 'yasushi',  
  country: 'Japan',  
},  
colors: ['Red', 'Green',  
'Blue']  
}
```

# Vue.js プログラム実習

- 右側のプログラムを index3.html として 作成保存

コードはこちらを使ってください

<https://github.com/osonoi-hot/vue>

```
<!DOCTYPE html>
<html>
  <head>
    <title>Vue.js のサンプル</title>
    <script src="https://unpkg.com/vue@3.0.0/dist/vue.global.js"></script>
  </head>
  <body>
    <h1>Vue3</h1>
    <div id="app">
      <p>{{ message }}</p>
      <p>{{ count }}</p>
      <p>{{ user.lastName }}</p>
      <p>{{ colors[1] }}</p>
    </div>
    <script>
      const appdata = {
        data() {
          return {
            message: 'Hello Vue',
            count: 77,
            user: {
              lastName: 'Osonoi',
              firstName: 'yasushi',
              country: 'Japan',
            },
            colors: ['Red', 'Green', 'Blue']
          }
        }
      }

      Vue.createApp(appdata).mount('#app')
    </script>
  </body>
</html>
```

# Vue.js プログラム実習

- 実行結果



## Vue3

Hello Vue

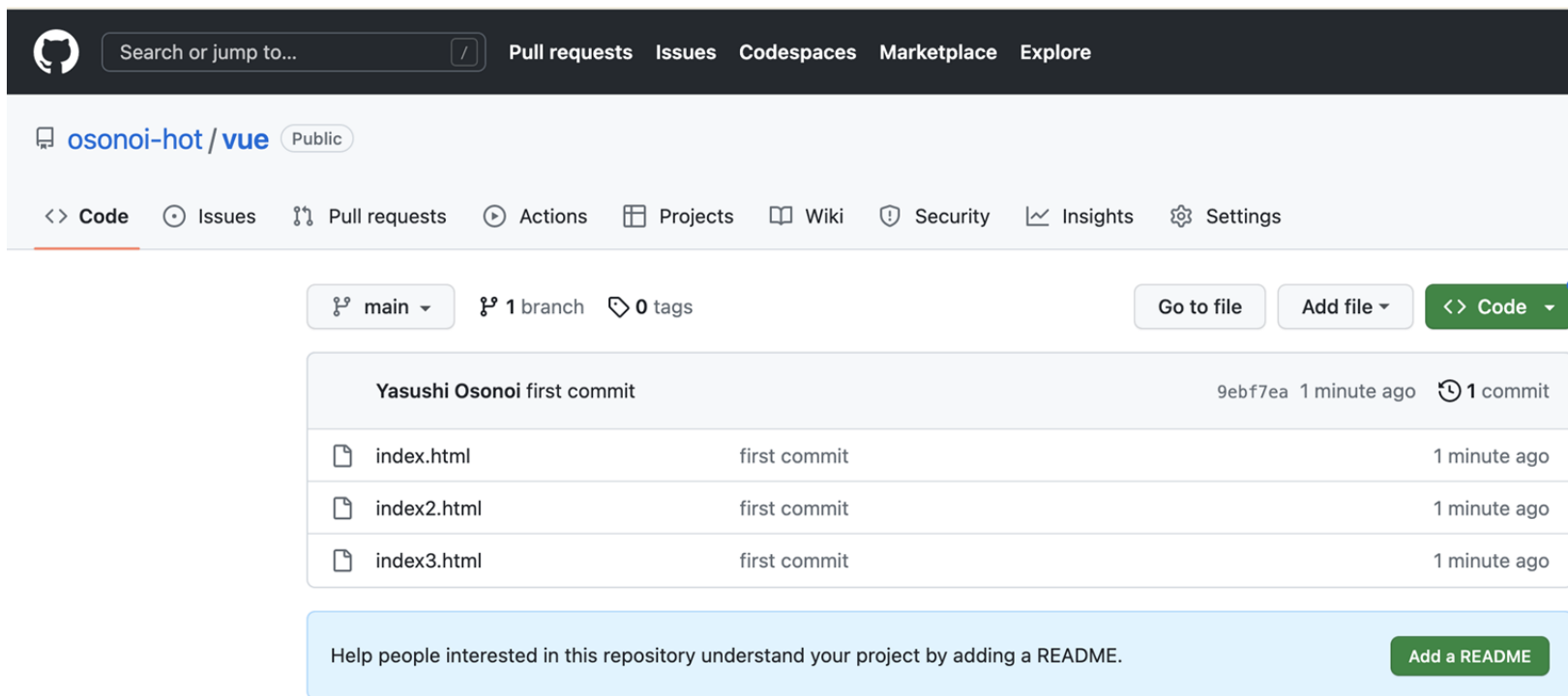
77

Osonoi

Green

# GitHubにコードをpush

- 新しいリポジトリを作成してpushする



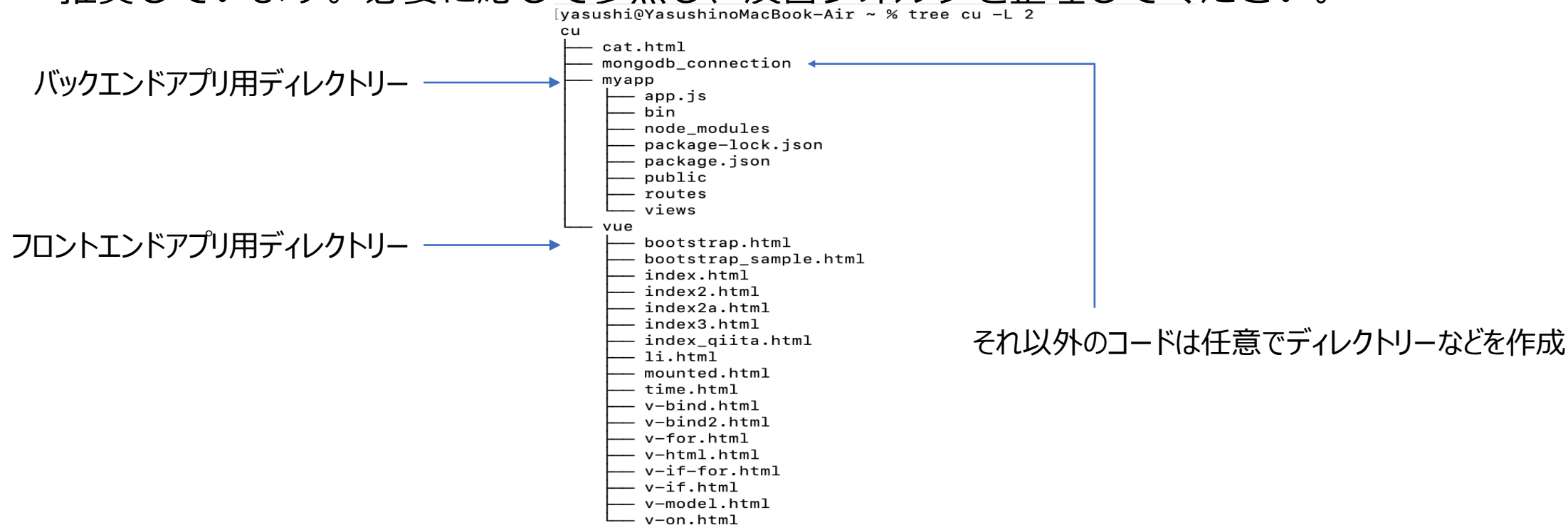
# 実機演習：GitHubにコードをpush

動画を全画面で視聴してください

# 実習時間

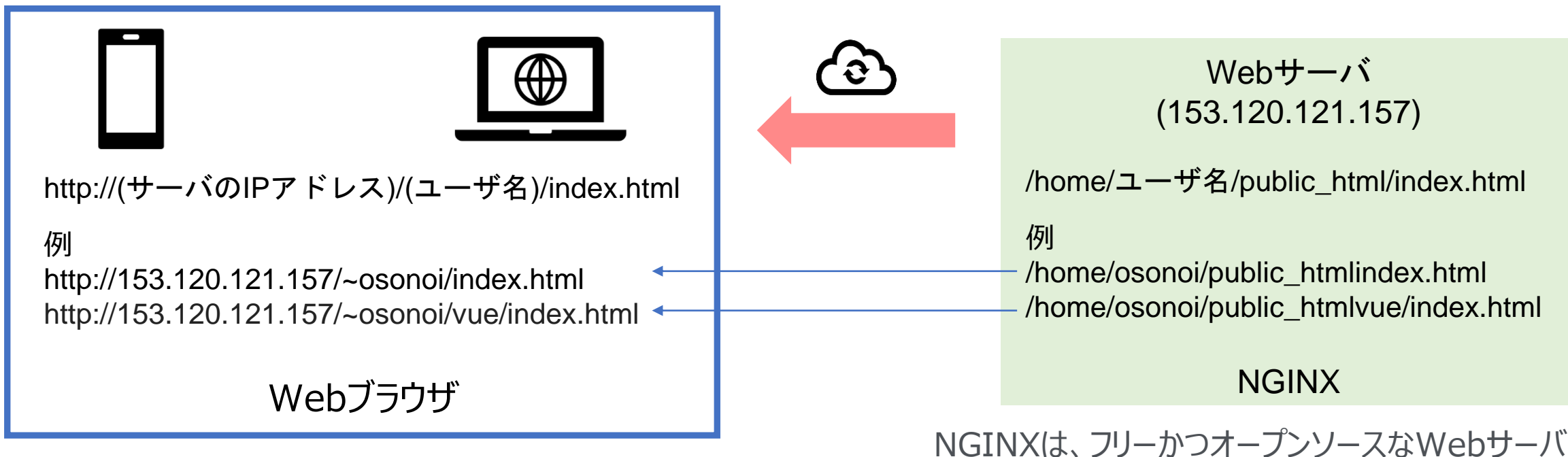
- ・ 10分程度を目安に動画を止めて前ページまでの実習をしてください。
- ・ 作業が終わったらビデオを再開して学習を進めてください。

※第1回4章で説明した通り、実習のファイル作成時には以下のディレクトリ構造を推奨しています。必要に応じて参照し、演習フォルダを整理してください。



# サーバにアップロード

- サーバのディレクトリとWeb URLアドレスの関係





# サーバにアップロード

- sshでサーバにログイン
- 下記要領でpublic\_html ディレクトリを作成

```
$ chmod 711 /home/(ご自分のユーザ名)
```

```
$ mkdir ~/public_html
```

```
$ chmod 755 ~/public_html
```

- GitHubのソースコードをクローン

```
$ git clone
```

- 例

```
$ git clone https://github.com/osonoi-hot/vue.git
```

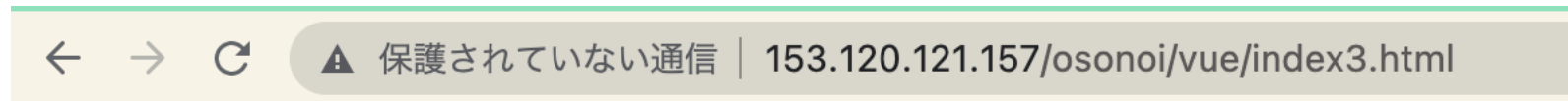
- ブラウザでアクセス

- アドレスはご自分のIPアドレス + ~ (ご自分のユーザ) + vue + file 名

例 http://153.120.121.157/~osonoi/vue/index2.html

# サーバにアップロード

- ブラウザで確認



## Vue3

Hello Vue

77

Osonoi

Green

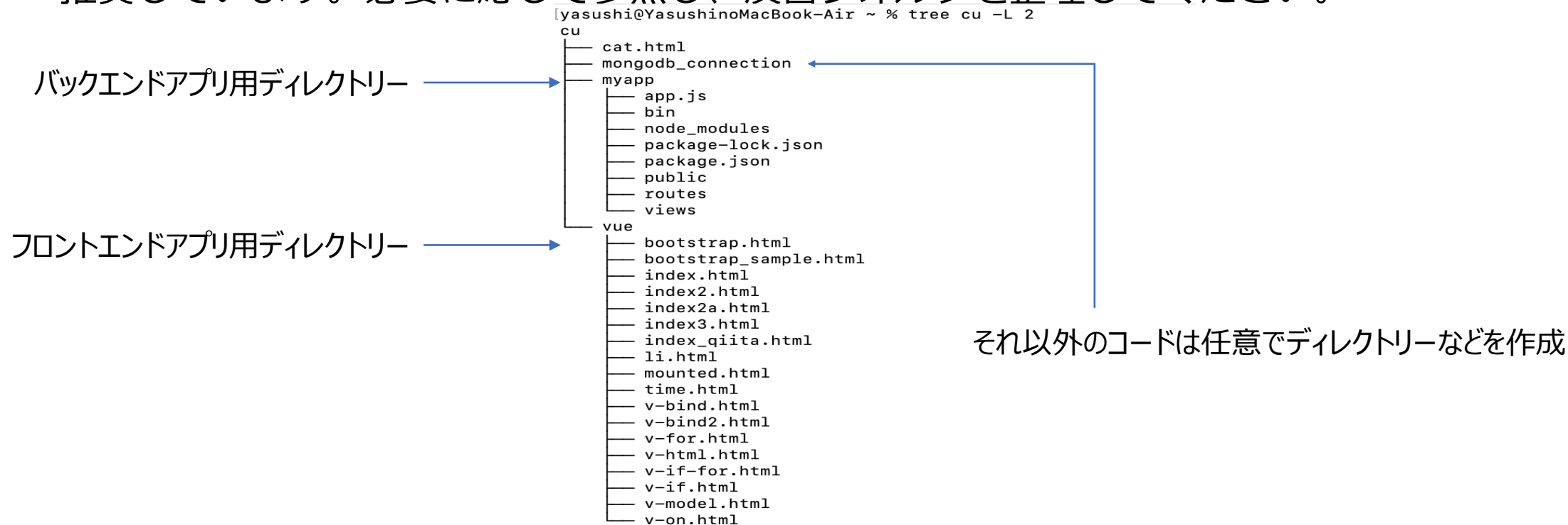
# 実機演習：サーバにアップロード

動画を全画面で視聴してください

# 実習時間

- ・ 10分程度を目安に動画を止めて前ページまでの実習をしてください。
- ・ 作業が終わったらビデオを再開して学習を進めてください。

※第1回4章で説明した通り、実習のファイル作成時には以下のディレクトリ構造を推奨しています。必要に応じて参照し、演習フォルダを整理してください。



## 第3章 まとめ

- Vue.jsの基本的なプログラム、動作を確認した
- サーバで動作確認を行った

## 第9回 まとめ

- JavaScriptフロントエンドフレームワークVue.jsの概要を学習した
- Vue.jsのインストール方法を学習した
- 開発者ツールの設定を行った
- Vue.jsを利用したプログラムを学習し作成した

JavaScriptフレームワークによるWebプログラミング  
第9回 Vue.js概要

## 第3章 Vue.js実習

終わり