

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2回 JavaScriptとフレームワーク

小園井康志

## 第2回 学習目標

- ・ Web開発に使われるJavaScriptおよびフレームワークについて理解し概要を説明できる
- ・ バックエンドの実行環境を作成することができる

# 第2回 授業構成

- 第1章 JavaScriptフレームワーク
- 第2章 nodejs実行環境構築（ローカル）

JavaScriptフレームワークによるWebプログラミング  
第2回 JavaScriptとフレームワーク

# 第1章

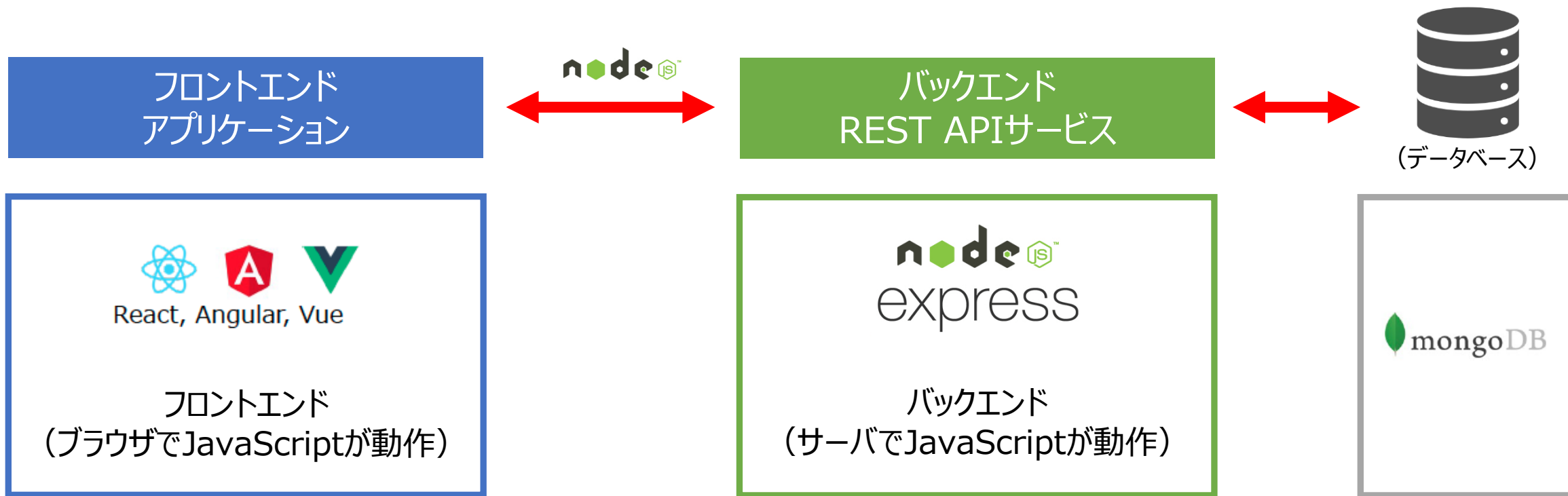
# JavaScriptフレームワーク

# 第1章 学習目標

- Web開発に使われるJavaScriptのフレームワークについて理解し、概要を説明できる

# JavaScriptフレームワークの例

- JavaScriptを多用したモダンなアーキテクチャ

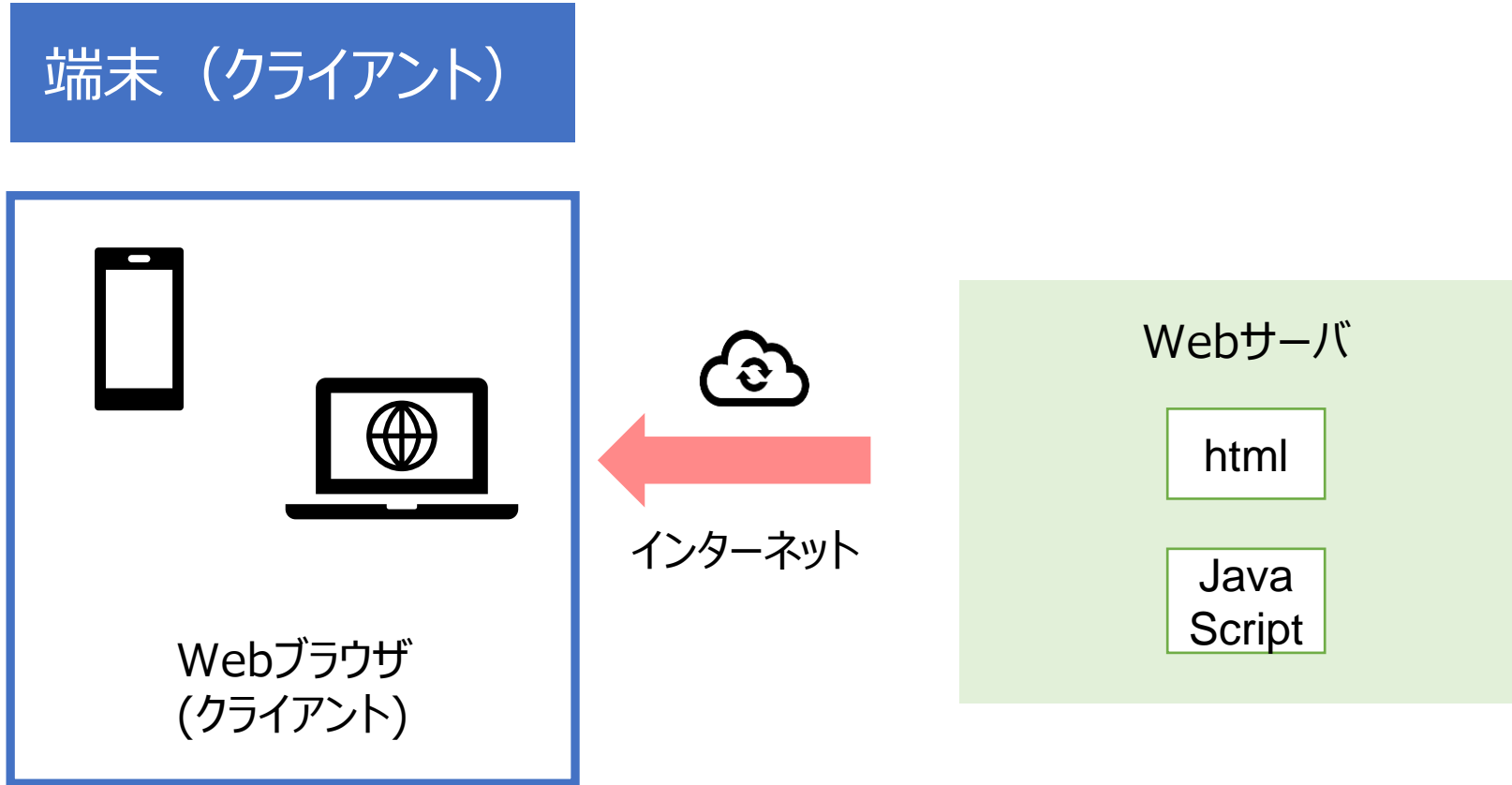


# フロントエンドとバックエンド

- フロントエンド（クライアントサイド）  
ユーザがブラウザで見るものを実現するコード
- バックエンド（サーバサイド）  
アプリケーションのロジックを組み立てるコード  
ログインページにおけるユーザ認証  
スケジュールに従ってタスクを実行  
外部サービスとの連動

# Webサイト作成とは違うWeb開発

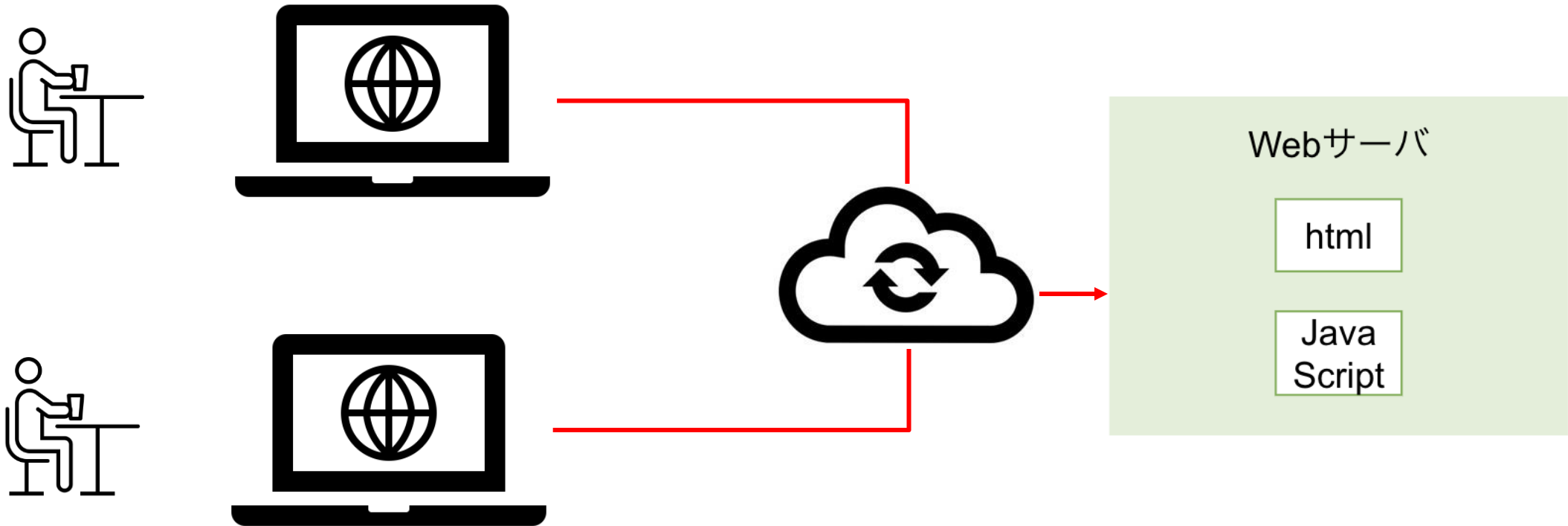
- Webサイト





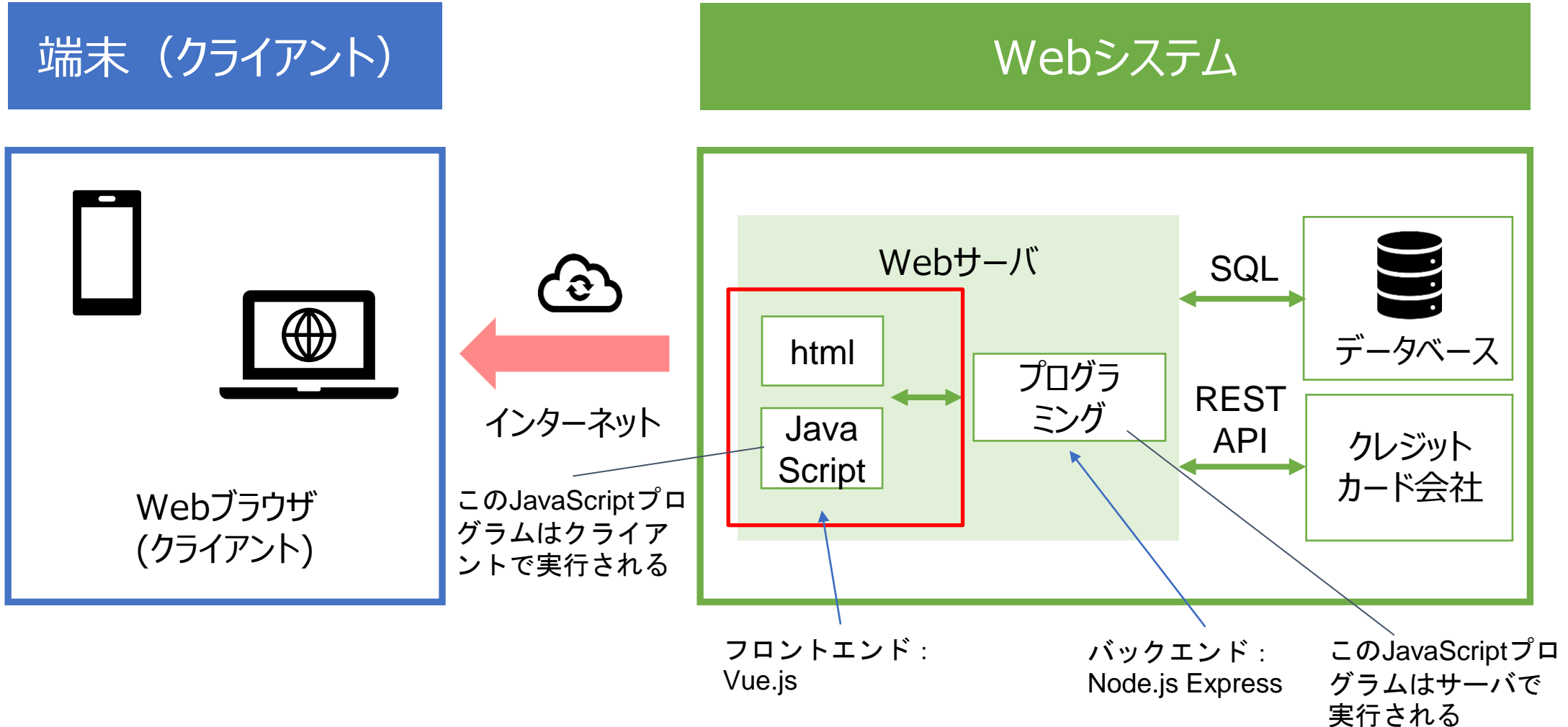
# 静的サイト

- 誰が見ても同じページが表示される



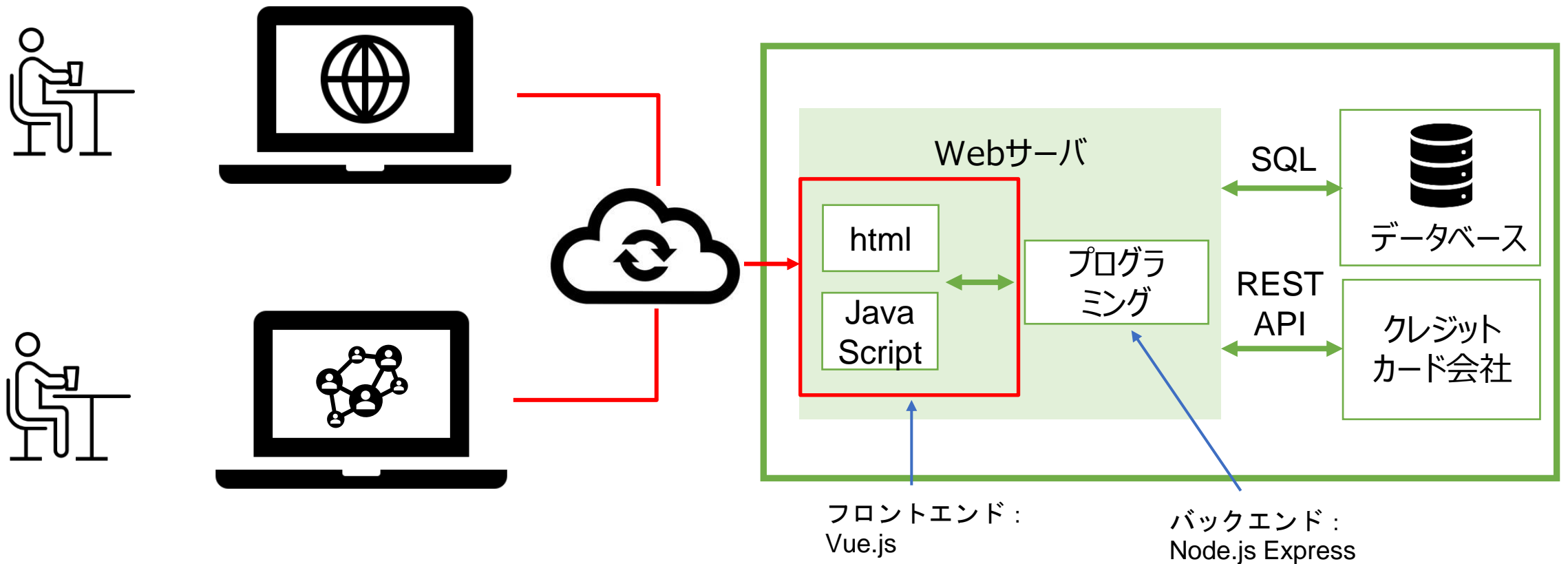
# Webサイト作成とは違うWeb開発

- Webアプリケーション



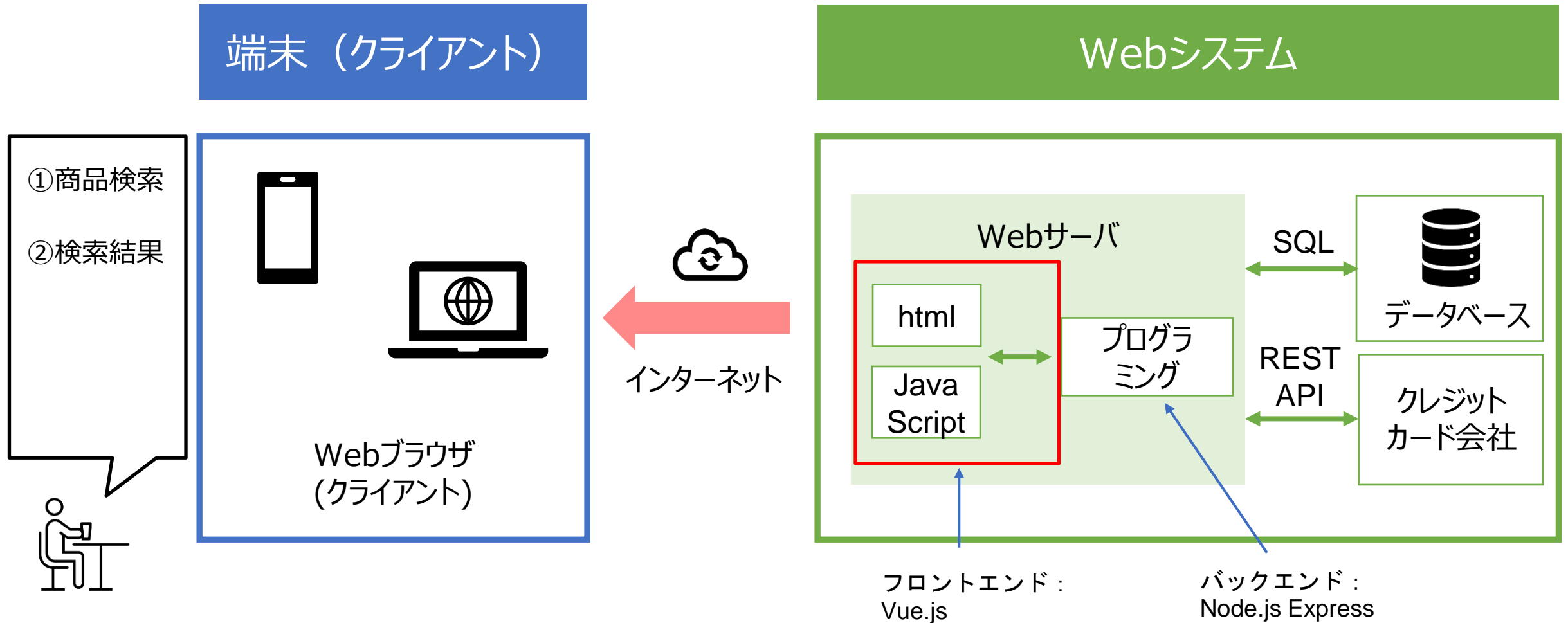
# 動的サイト

- ・ 見る人の状況によって見えるページが違う
- ・ 内容はプログラミングによってコントロール



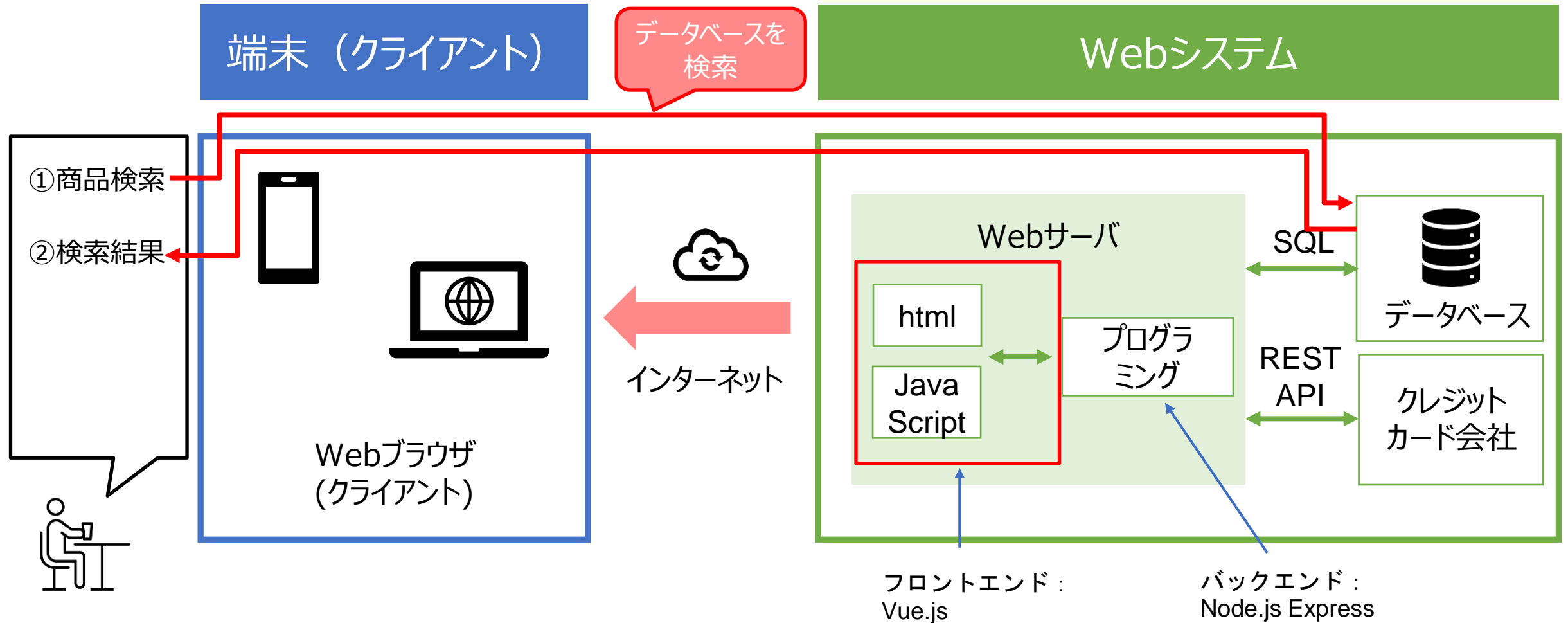
# アプリの例

- オンラインショップで商品を買う場合



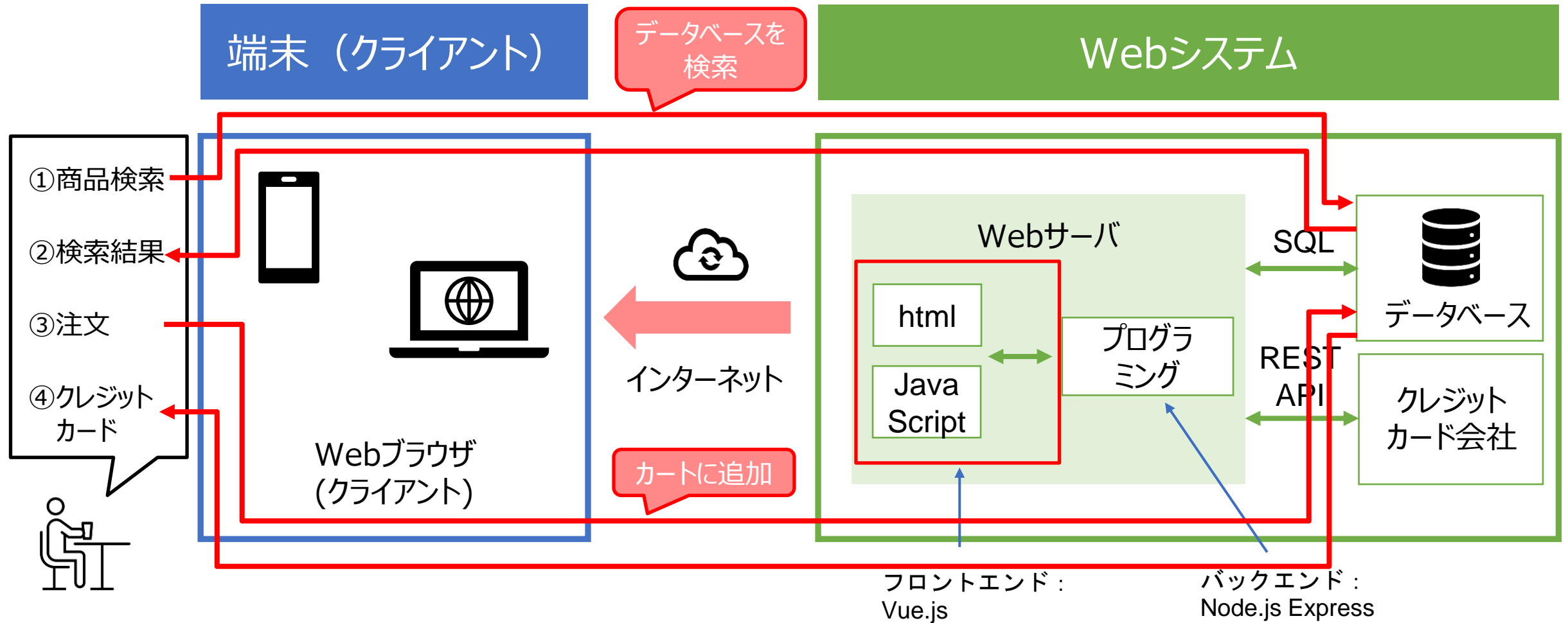
# アプリの例

- オンラインショップで商品を買う場合



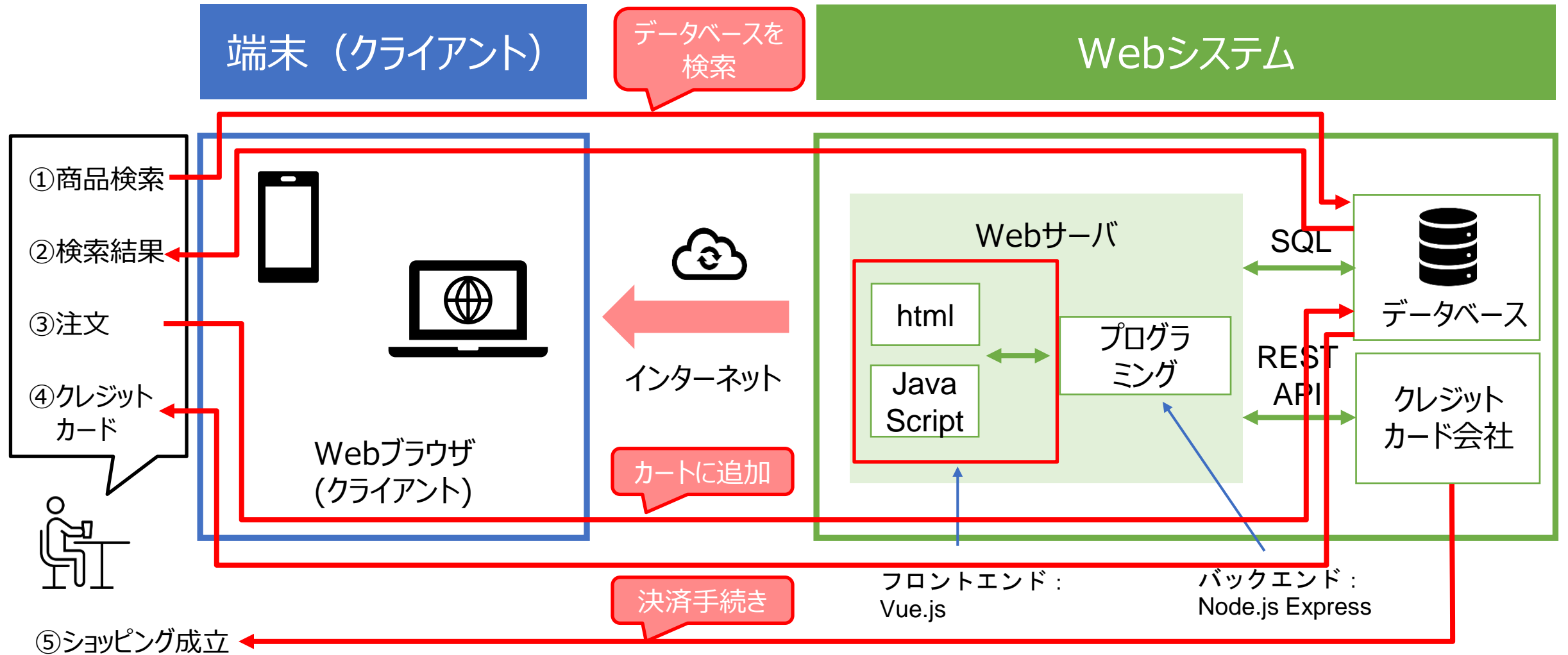
# アプリの例

- オンラインショップで商品を買う場合



# アプリの例

- オンラインショップで商品を買う場合



# 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。



# フレームワーク比較

	Angular	React	Vue
開発元	Google + コミュニティ	Facebook + コミュニティ	コミュニティ
派生フレームワーク	Angular Universal	Next.js, Gatsby.js特化	Nuxt.js
長所	MVVM（Model, View, ViewModel）と呼ばれるフルスタックの機能を提供する。学習コストは高いものの、標準が確立されているため、統制の取れた開発が可能。Googleの各種サービス他、商用事例が多い。	React自体はViewにしておりシンプルであり、学習は容易。React NativeによりiOS、Android等のネイティブアプリケーションも開発可能。Facebookグループ他、商用事例が多い。スキルの有るエンジニアにとっては応用が効きVueよりも使いやすい。	React, Angularと比較して、新しいフレームワークであり、両者の長所・短所を研究し、良い所取りをしている （Reactに近い） 近年、ユーザー数が急激に伸びている。学習が容易で、シンプルな記述方法、開発効率を重要視している。
短所	フルスタックであるがゆえに初期学習コストが大きい。 過去のバージョン（Angular1）と互換性が無く、情報収集の際は注意が必要。	View以外の部分に関してはサードパーティライブラリに依存しているが、それらも人気が高いがゆえにトレンド、デファクトスタンダードが変化する。	他のフレームワークよりも細かな制御やパフォーマンスで一步劣る場合がある。

# バックエンドJavaScript実行環境 Node.js

Node.js公式サイト

<https://nodejs.org/en/>

ライアンダーサル氏によって2009年に作成された、  
V8 JavaScriptエンジン上に構築されたJavaScript実行環境。

\* V8 JavaScriptエンジン： Googleが開発するオープンソース  
JavaScriptエンジン

# Node.jsの特長

- フロントエンドで使われるJavaScriptがバックエンドでも使用できる。  
Webの開発においてはフロントエンドはJavaScript、バックエンドは別の言語という組み合わせが多かったがNode.jsならどちらも同じ言語で開発ができる。
- Webサーバも作成できる。  
他のフレームワークでは一般にnginxなどのWebサーバを構築して、そこにフレームワークで作成したものを配置するが、Node.jsではWebサーバのプログラムも作成することができる。  
フレームワークで仕組みが用意されているので実際には数行プログラムを書くだけで実行できる。

# Node.jsの特長

- シングルスレッドで動作、ノンブロッキングI/O、イベント駆動型プログラミングモデル（コールバック関数などを使う）
  - \* イベント駆動型プログラミング：例えばマウスを動かすと、それに反応するゲームプログラムなどである。このようなプログラムでは、フローの代わりにイベントに応じた動作を記述する。
- C10K問題（1000クライアント）を効率的に解決可能
  - \* C10K問題：クライアントが約1万台に達すると、Webサーバのハードウェア性能に余裕があるにも関わらず、レスポンス性能が大きく下がる問題。

# バックエンドJavaScriptフレームワーク Express

Express公式サイト

<http://expressjs.com/ja/>

ExpressはNode.jsで利用できるWebアプリケーションフレームワーク。

Node.jsに比較的近い使い心地であり、小さくてパワフル。

Generatorと呼ばれるプロジェクト作成を簡単にするプログラムが用意されている。

# 第1章 まとめ

JavaScriptおよびフレームワークについて、  
以下の内容を学習した。

- JavaScriptフレームワークにおける、  
フロントエンド、バックエンドの動き
- 主なフロントエンドJavaScriptフレームワークとその特長  
Angular, React, Vue
- バックエンドJavascript実行環境  
Node.js
- バックエンドJavaScriptフレームワーク  
Express

JavaScriptフレームワークによるWebプログラミング  
第2回 Web開発の変遷とJavascript

# 第1章

# JavaScriptフレームワーク

終わり

JavaScriptフレームワークによるWebプログラミング  
第2回 JavaScriptとフレームワーク

## 第2章 Node.js実行環境作成



# はじめに

- ・ このチャプターは“mac環境用”になります。  
Windows環境で受講されている方は、  
“Windows環境用”のチャプターを受講してください

## 第2章 学習目標

- ・ 次回以降のバックエンド実習実行環境として、Node.jsをインストールする。

# コマンドラインの使い方 (Mac)

## 前回の復習

- LaunchPadを起動
- "その他"をクリック、"ターミナル"をクリックして起動する



# Node.jsインストール (Mac)

- MacではWindows環境とは違いMacのパッケージ管理ソフト Homebrewを使ってインストールする事をお勧め  
(後ほどバージョンを変えたい時なども比較的簡単にできる)

手順は

1. Homebrewのインストール
2. Homebrewでnvm (Node Version Manager) をインストール
3. nvmでNode.jsをインストール

\*以降の説明では、ターミナルでの入力は以下の表現で表します。

% コマンド

# Homebrewのインストール

- ・ターミナルを立ち上げて以下のコマンドを入力

```
% /bin/bash -c "$(curl -fsSL
```

```
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

- ・コマンドはHomebrewのWebサイトを参照

<https://brew.sh/ja/>



# Homebrewの動作確認

- 以下のコマンドを入力

```
% brew -v
```

## 出力例

\*バージョンが出てくれば問題なくインストールされています  
バージョンはインストールされた時期によって異なります

```
[shousonoiyasushi@shousonoiyasushinoMacBook-Pro ~ % brew -v  
Homebrew 3.5.2  
Homebrew/homebrew-core (git revision aff7142cd05; last commit 2022-06-25)
```

# nvm (Node Version Manager) のインストール

- 以下のコマンドを入力

```
% brew install nvm
```

- インストール確認

```
% nvm
```

## 出力例

\*以下のようなメッセージが出てくればnvmのインストールはできています

```
[shousonoiyasushi@shousonoiyasushinoMacBook-Pro ~] % nvm

Node Version Manager (v0.38.0)

Note: <version> refers to any version-like string nvm understands. This includes:
- full or partial version numbers, starting with an optional "v" (0.10, v0.1.2, v1)
- default (built-in) aliases: node, stable, unstable, iojs, system
- custom aliases you define with `nvm alias foo`

Any options that produce colored output should respect the `--no-colors` option.

Usage:
  nvm --help          Show this message
  nvm --no-colors     Suppress colored output
```

\*インストールに問題があった場合には次ページを参考に”n”をインストール

# n のインストール(nvmのインストールで問題があった場合)

- 以下のコマンドを入力

```
% brew install n
```

- インストール確認

```
% n --version
```

## 出力例

\*以下のようなメッセージが出てくればnのインストールはできています

```
[~ % n --version  
v9.0.0_
```



# nvmでNode.jsをインストール

- 以下のコマンドを入力

```
% nvm install v18.4.0
```

```
% n lts (nをインストールした場合)
```

- インストール確認

```
% node --version
```

```
% npm --version
```

## 出力例

\*以下のようにバージョン情報が出てくればインストールは問題なくできています

```
[shousonoiyasushi@shousonoiyasushinoMacBook-Pro ~ % node --version  
v18.4.0
```

```
[shousonoiyasushi@shousonoiyasushinoMacBook-Pro ~ % npm --version  
8.12.1
```

## 第2章 まとめ

- Windows、Mac環境でのNode.jsのインストール方法を学習した。
  - Windows環境の場合はVoltaを利用してインストールを行う
  - Macの場合はHomebrew及びnvmを使用する。

## 第2回 まとめ

- Web開発に使われるJavaScriptフレームワークについて、以下の点を学習した
  - JavaScriptフレームワークにおけるフロントエンド、バックエンドの動き
  - 主なフロントエンドJavaScriptフレームワークとその特長
  - バックエンドJavascript実行環境
  - バックエンドJavaScriptフレームワーク
- バックエンドの実行環境を作成する方法として、Node.jsのインストール方法を学習した
  - Windows環境の場合はVoltaを利用してインストールを行う
  - Macの場合はHomebrew及びnvmを使用する

JavaScriptフレームワークによるWebプログラミング  
第2回 Web開発の変遷とJavascript

# 第2章 Node.js実行環境作成

終わり

JavaScriptフレームワークによるWebプログラミング  
第2回 JavaScriptとフレームワーク

## 第2章 Node.js実行環境作成

# はじめに

- ・ このチャプターは“Windows環境用”になります。  
mac環境で受講されている方は、“mac環境用”の  
チャプターを受講してください


## 第2章 学習目標

- ・ 次回以降のバックエンド実習実行環境として、Node.jsをインストールする。

# コマンドラインの使い方 (Windows)

## 前回の復習

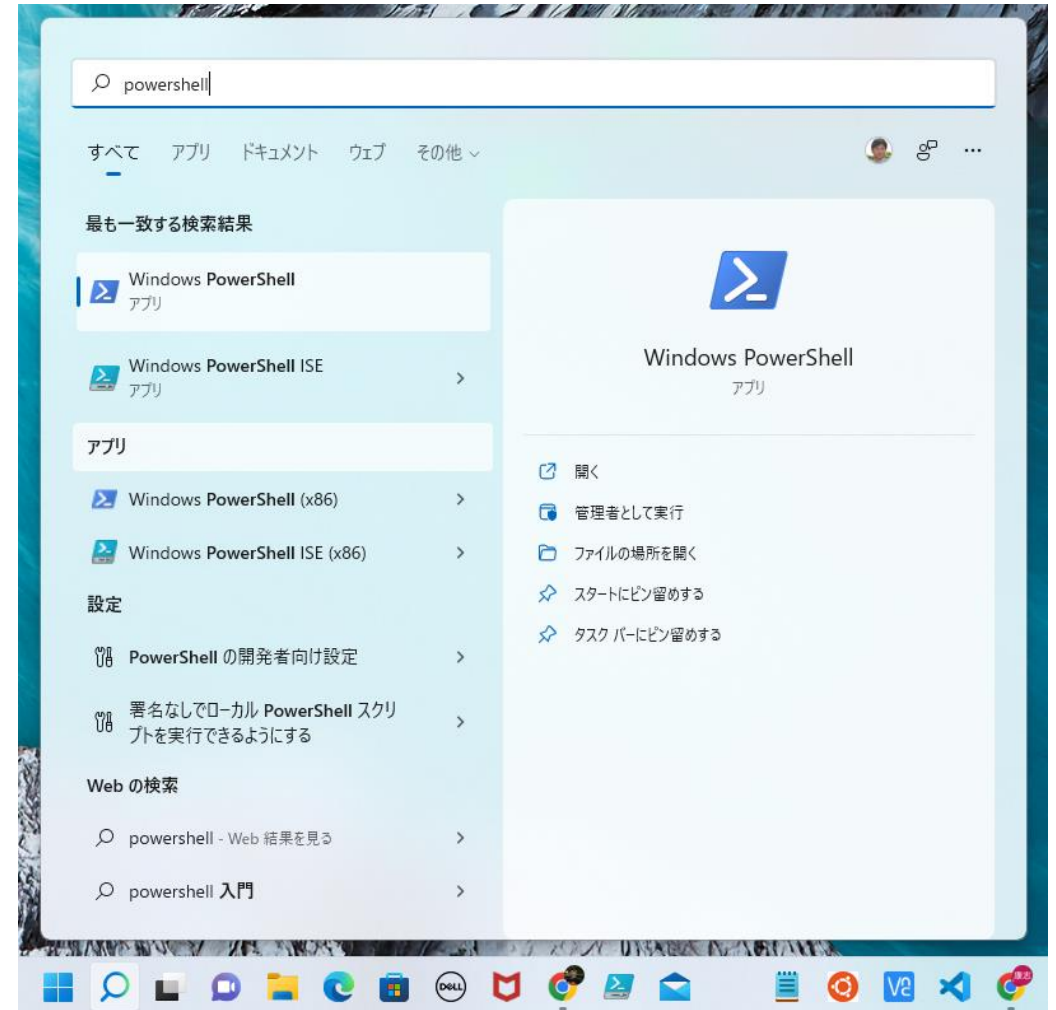
- 検索で"Powershell"を入力
- Windows PowerShell を選択



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

新機能と改善のために最新の PowerShell をインストールしてください!https://aka.ms/PSWindows

PS C:\Users\Yosono>
```





# Node.jsのインストール

- ・ WindowsへのNode.jsのインストールにはVoltaというツールを利用する
- ・ このツールはNode.jsのバージョン管理ソフトウェア
- ・ Nodejsのインストールはバージョン管理も可能

公式サイト

<https://volta.sh/>

# Node.jsのインストール

- Node.jsのバージョン管理ツールVoltaをインストールし、これを使用してnode.jsをインストールする。
- Volta公式サイト：ダウンロード

<https://docs.volta.sh/guide/getting-started>

Windows版インストーラーのダウンロードは以下リンクから

## Windows Installation

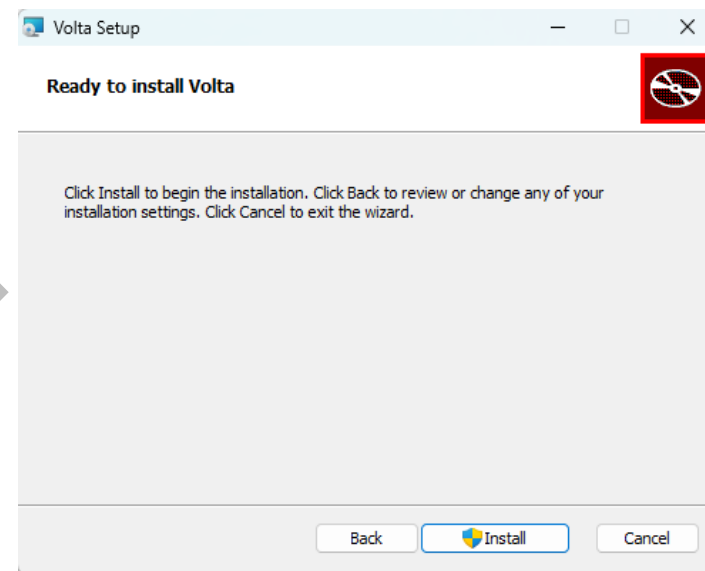
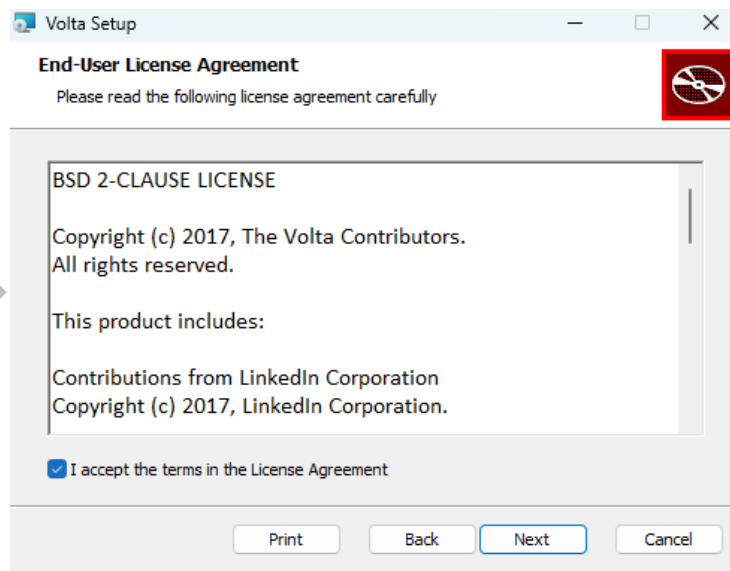
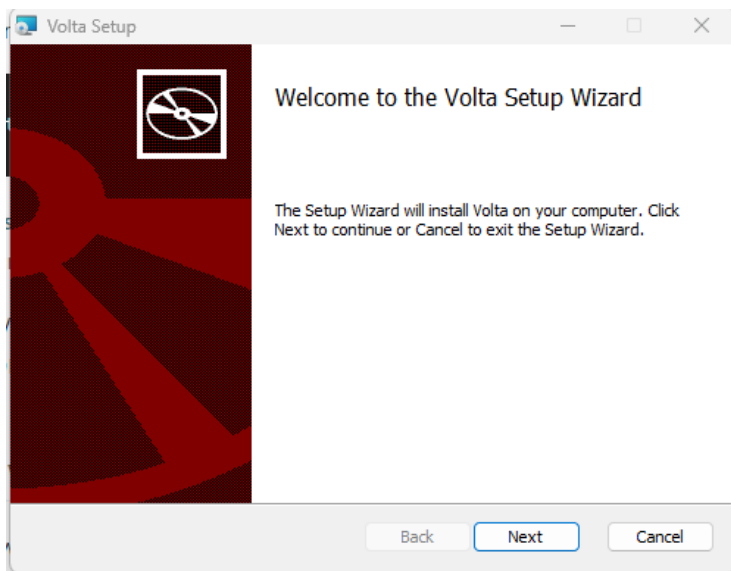
---

For Windows, [download and run the Windows installer](#) and follow the instructions.

# Voltaのインストール (Windows)

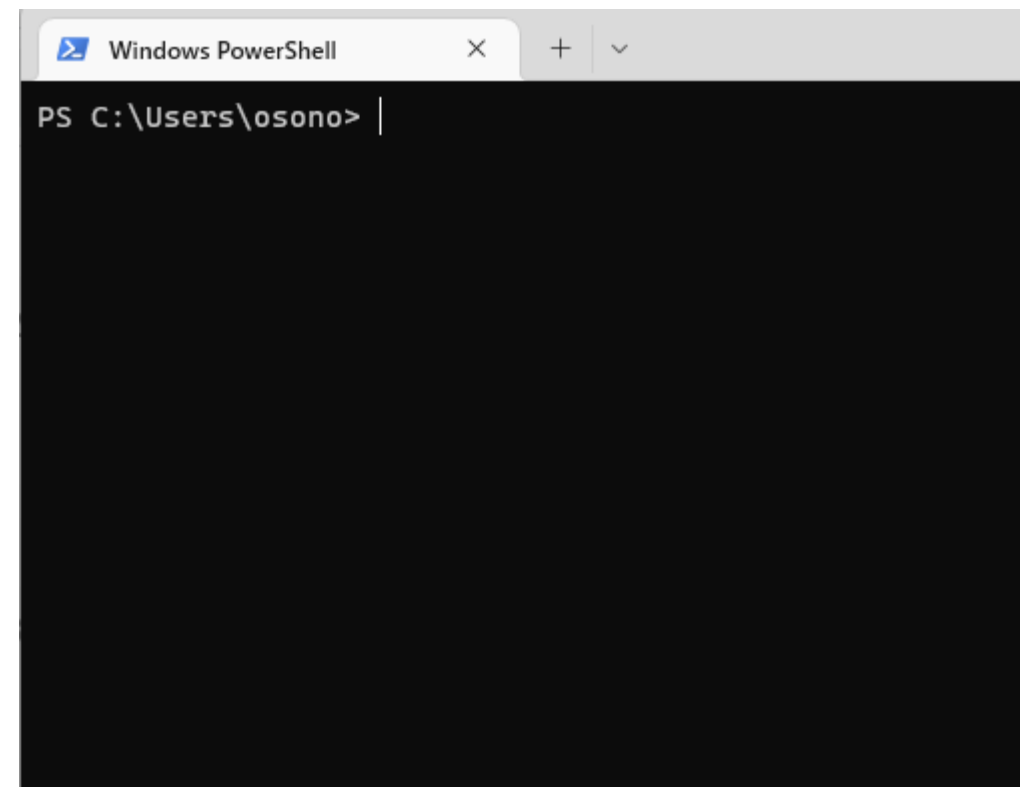
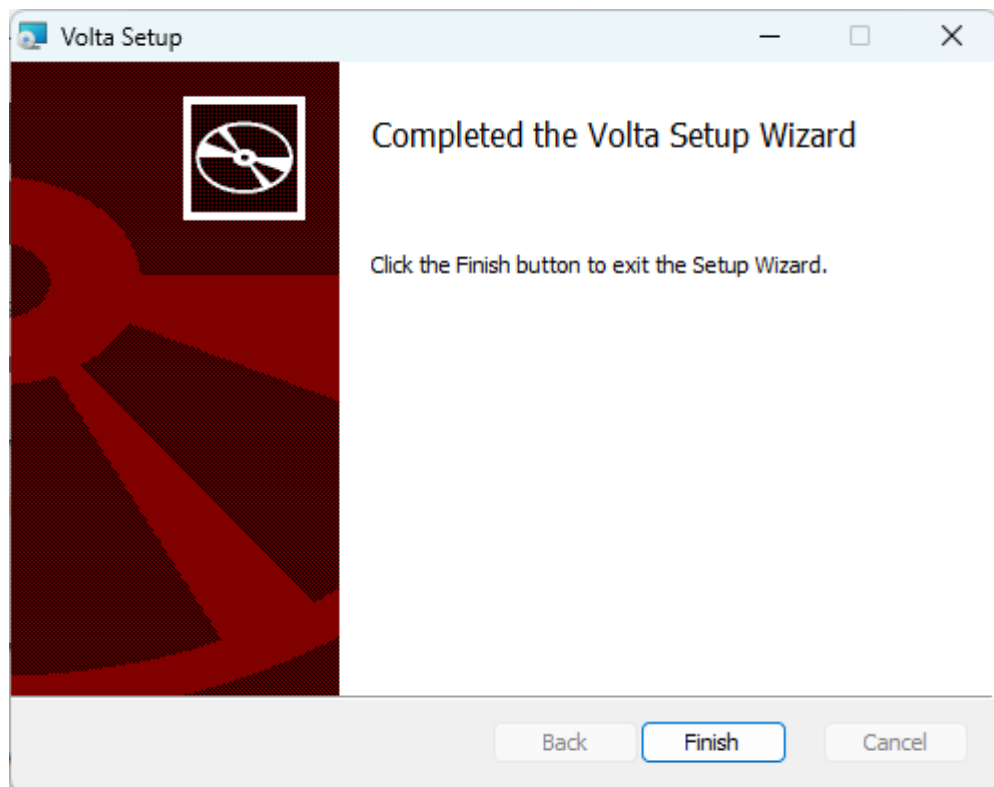
インストーラをダウンロード後、実行してウィザードに従いインストール作業を進める。

※設定は変更せず、「Next」でOK



# Voltaのインストール (Windows)

インストールできたらPowerShellを起動



# コマンドラインの使い方 (Windows)

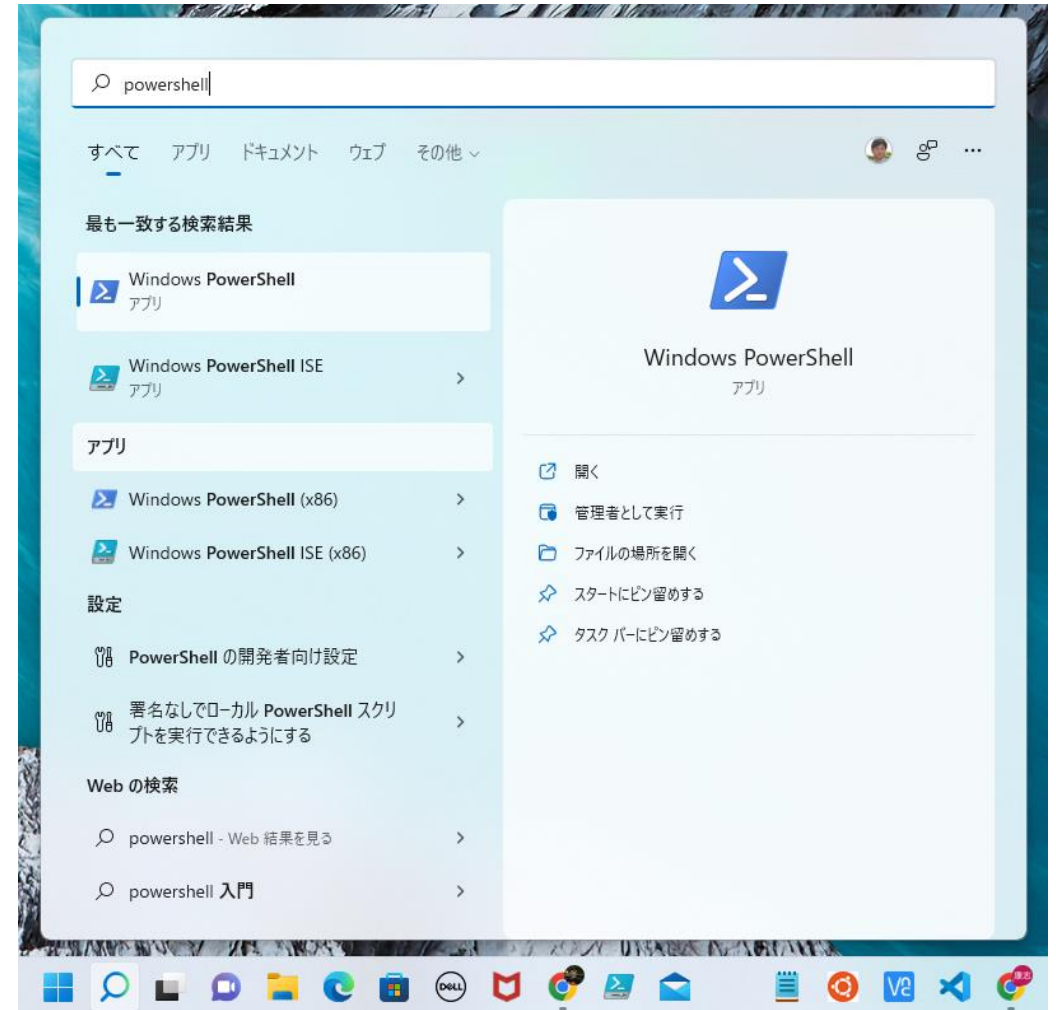
- 検索で"Powershell"を入力
- Windows PowerShell を選択



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

新機能と改善のために最新の PowerShell をインストールしてください!https://aka.ms/PSWindows

PS C:\Users\Yosono>
```



# Voltaを使ったnode.jsのインストール (Windows)

- 以下のコマンドを入力

voltaを使ってnode.jsのインストール

```
> Volta install node@18
```

- 動作確認

```
> node --version
```

```
> npm --version
```

実行例

```
PS C:\Users\osono> node --version
v18.12.1
PS C:\Users\osono> npm --version
8.19.2
PS C:\Users\osono> |
```

## 第2章 まとめ

- Windows、Mac環境でのNode.jsのインストール方法を学習した。
  - Windows環境の場合はVoltaを利用してインストールを行う
  - Macの場合はHomebrew及びnvmを使用する。

## 第2回 まとめ

- Web開発に使われるJavaScriptフレームワークについて、以下の点を学習した
  - JavaScriptフレームワークにおけるフロントエンド、バックエンドの動き
  - 主なフロントエンドJavaScriptフレームワークとその特長
  - バックエンドJavascript実行環境
  - バックエンドJavaScriptフレームワーク
- バックエンドの実行環境を作成する方法として、Node.jsのインストール方法を学習した
  - Windows環境の場合はVoltaを利用してインストールを行う
  - Macの場合はHomebrew及びnvmを使用する



JavaScriptフレームワークによるWebプログラミング  
第2回 Web開発の変遷とJavascript

# 第2章 Node.js実行環境作成

終わり