

Vue.js

入門

INDEX

■ SPAフロントフレームワーク

- I. Node.js & JavaScript
- II. フロントエンド & バックエンド
- III. フロントエンドフレームワーク
- IV. React・Angular・Vue
- V. Vue.js のインスタンス
- VI. Vue.js のコンポーネント間データ管理
- VII. Vue.js のライブラリ Vuex①
- VIII. Vue.js のライブラリ Vuex②

■ 開発環境セットアップ

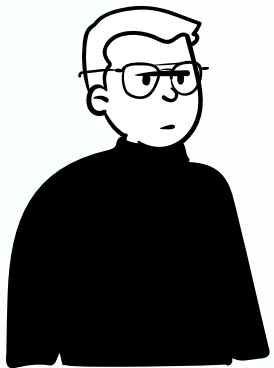
- I. Node.js インストール
- II. Vue.js インストール
- III. 付加機能インストール
- IV. Vueプロジェクト生成

■ Vue基本構文

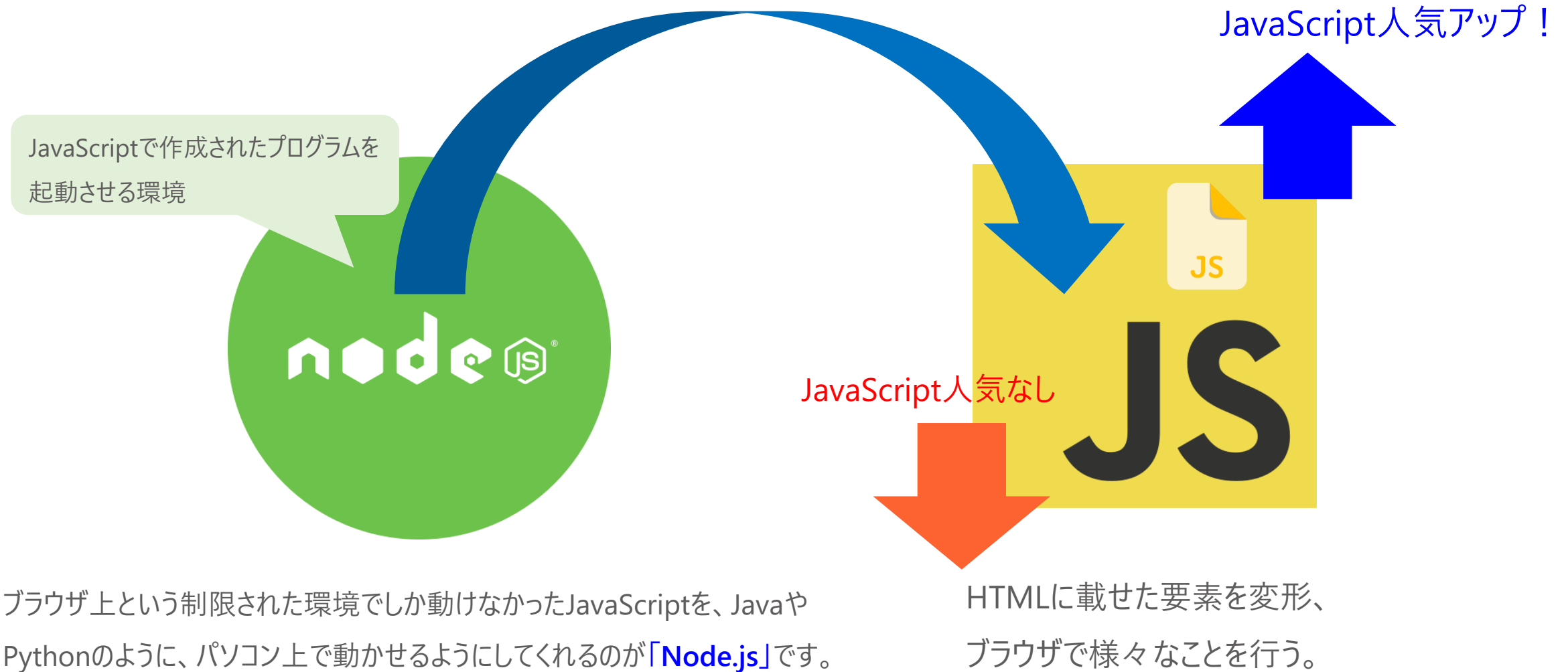
- I. データバインディング
- II. リストレンダリング
- III. 条件付きレンダリング
- IV. イベントハンドリング
- V. フォーム入力バインディング

■ ミニプロジェクト

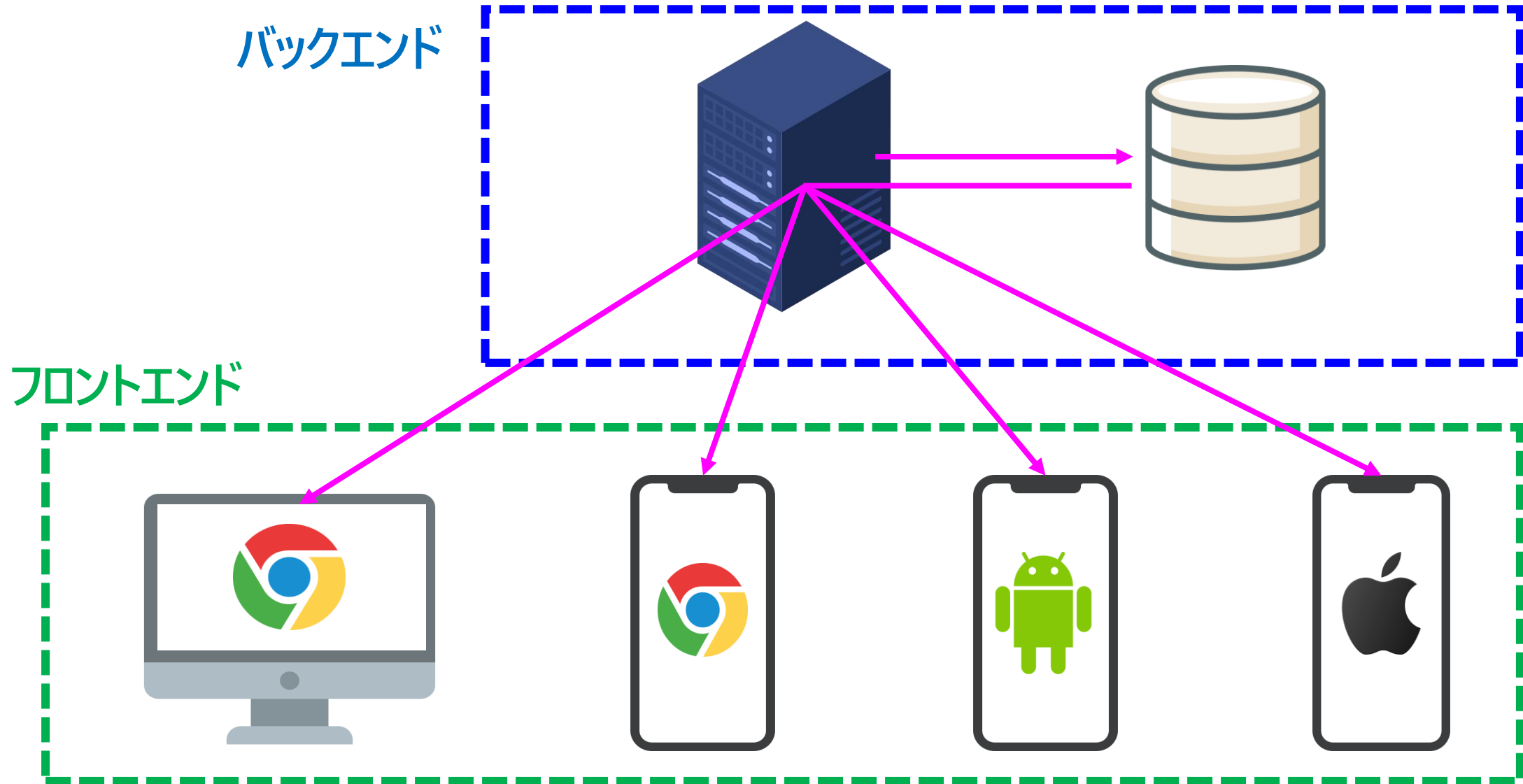
1. SPA フロントフレームワーク



1. Node.js & JavaScript



2. フロントエンド & バックエンド



3. フロントエンドフレームワーク



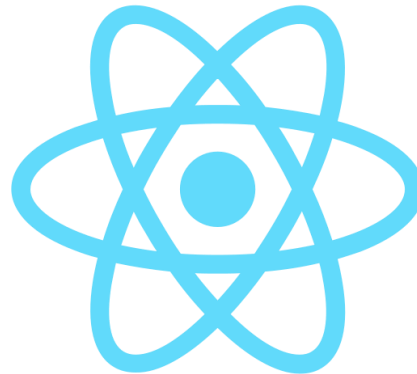
ブラウザでも動作する「JavaScript」で、
フレームワークを作成すれば、可能ではないでしょうか。



SPAフレームワーク *Single Page Application*



Angular.js



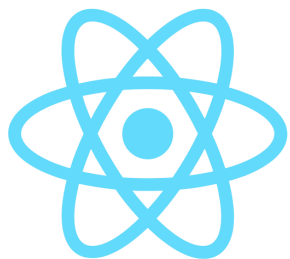
React



Vue.js

4. React ・ Angular.js ・ Vue.js

SPAフレームワーク *Single Page Application*



React



- ・Facebookが主導しているので安心。
- ・ライブラリであるので、利用しやすい。

単一のページを部分的に更新することで、ページ遷移を行わずにコンテンツを完結するWebページのことです。



Angular.js



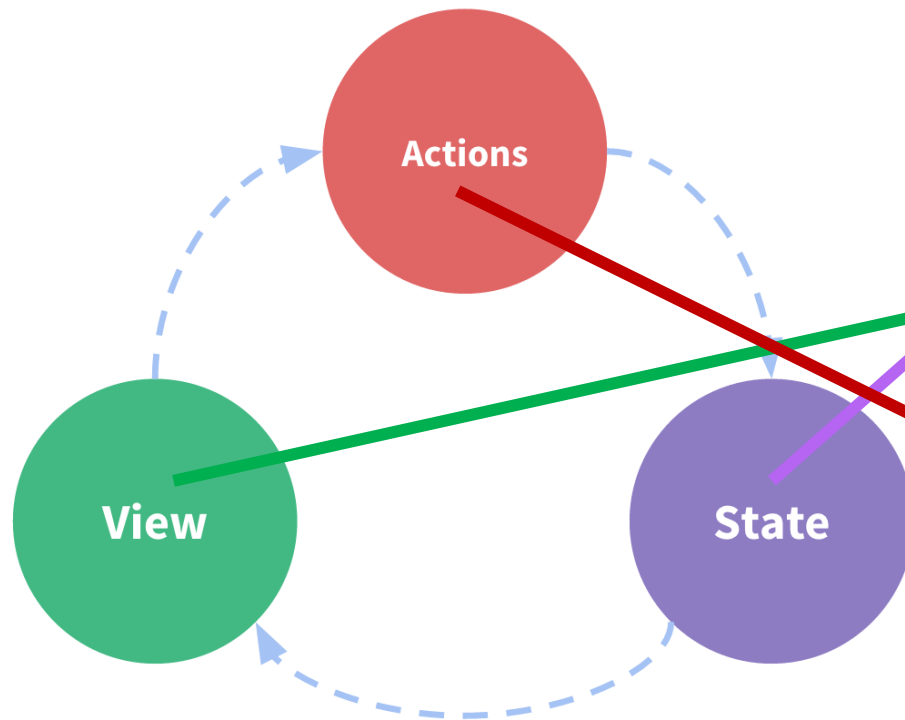
- ・ TypeScriptを補完した言語で、非常に安定するフロントエンドアプリを作ることができる。



- ・ 単一ファイルコンポーネント。
- ・ 学習コストが低い。
- ・ データの同期。

Vue.js のインスタンス

Viewインスタンス (コンポーネント)



new Vue({

State

```
data () {  
  return {  
    count: 0  
  }  
},
```

Appを駆動する
データ

View

```
template: `  
  <div>{{ count }}</div>  
`
```

状態のただの宣言
的なマッピング

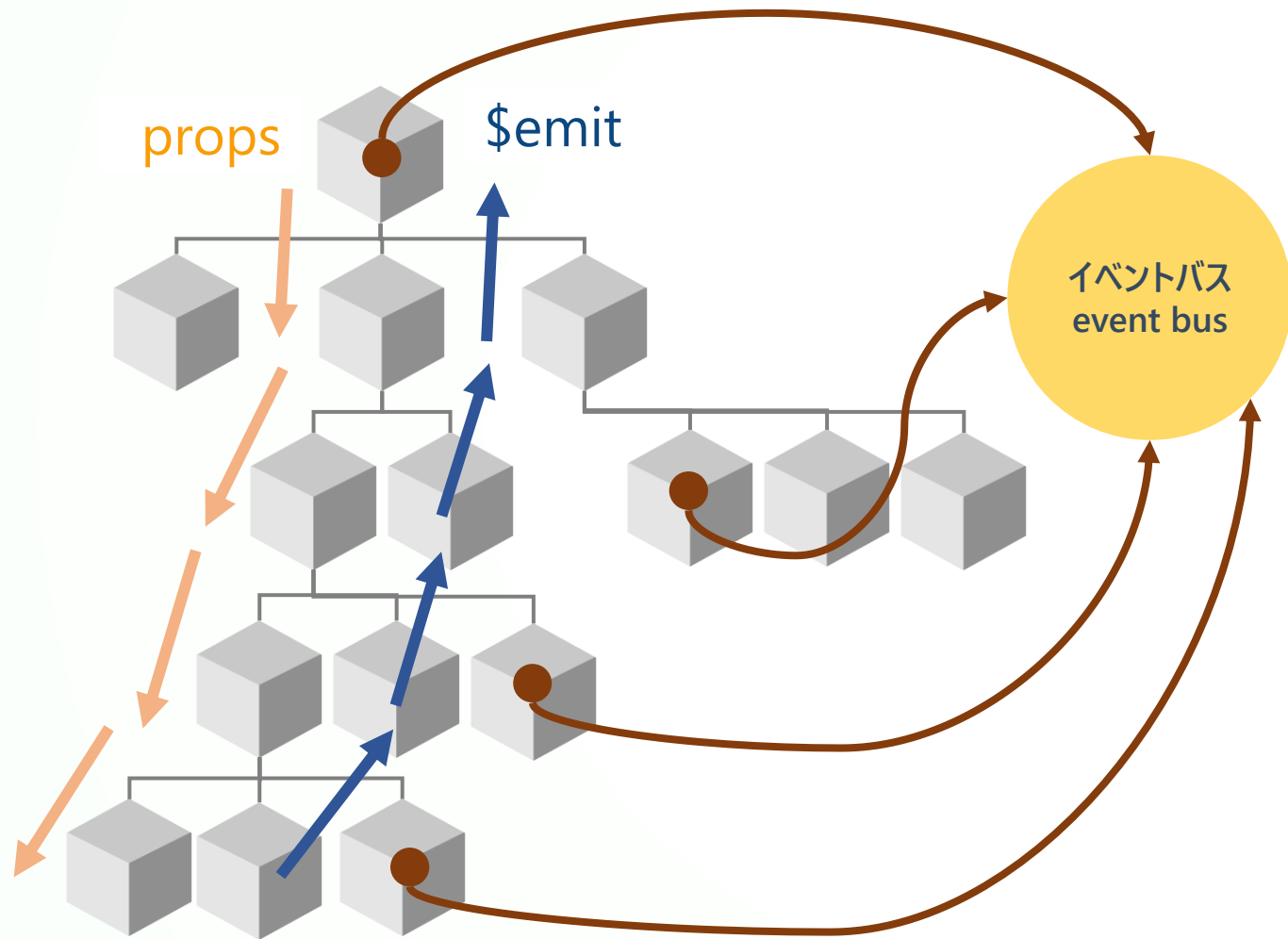
Action

```
methods: {  
  increment () {  
    this.count++  
  }  
}
```

ビューからのユーザー入力に反応
して、状態の変更を可能にする
方法

})

Vue.js のコンポーネント間データ管理

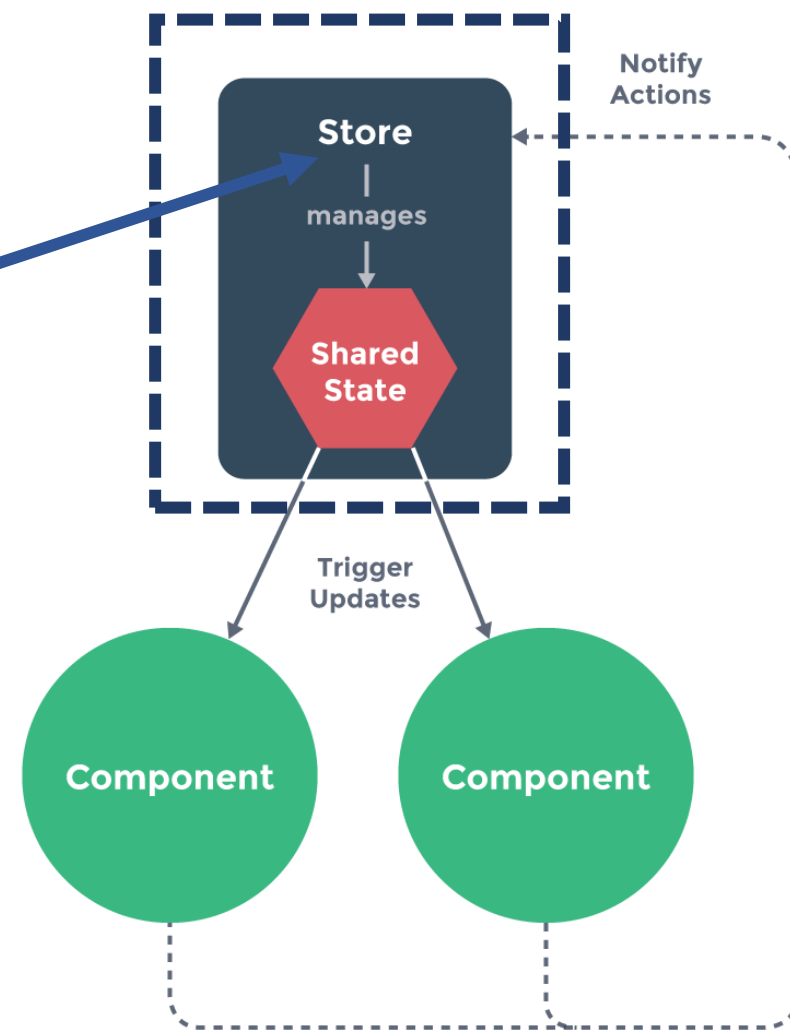
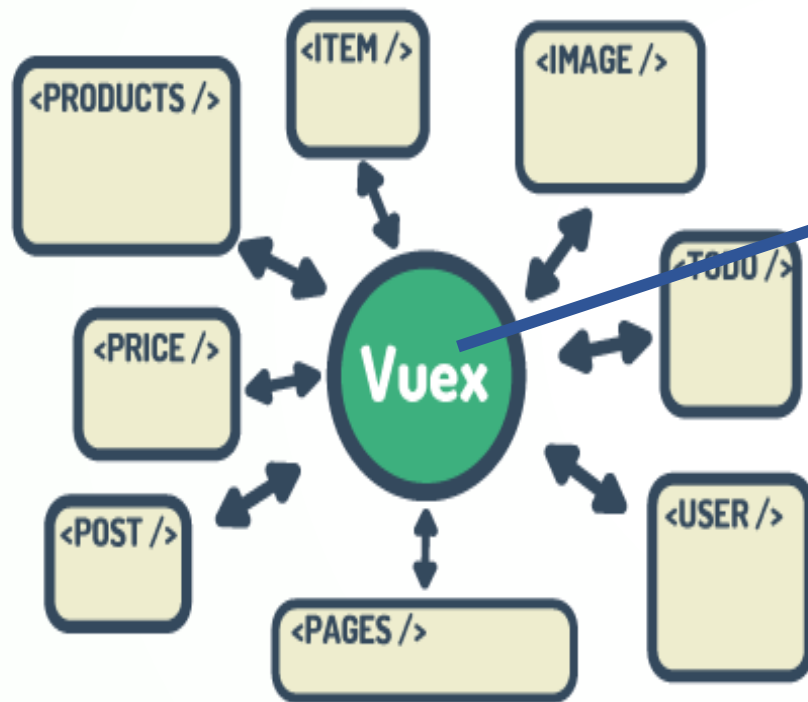


Props, \$emitにて、親子関係で、データを渡す。

状態管理インスタンス(イベントバス)を利用して、親子間以外の場合でも通信を行いデータを管理。

1つのバスを使用するため、大規模のプロジェクトでは、限界がある。

Vue.js のライブラリ Vuex①



Vue.js のライブラリ Vuex

Actions

- Mutationを実行させる役割を担う。
- actions内で、関数を作成する。
- dispatch('関数名', '引数') の方式で、呼び出す。

```
dispatch('setAccount', account);

export const actions = {
  setAccount({commit, dispatch}, account) {
    commit('currentUser', account);
    dispatch('setIsAdmin', account.uid);
  }
}
```

<POSTS />

State (データオブジェクト)

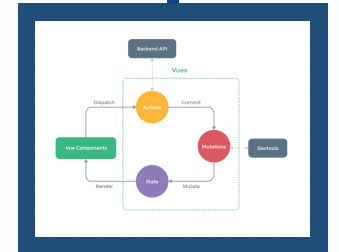
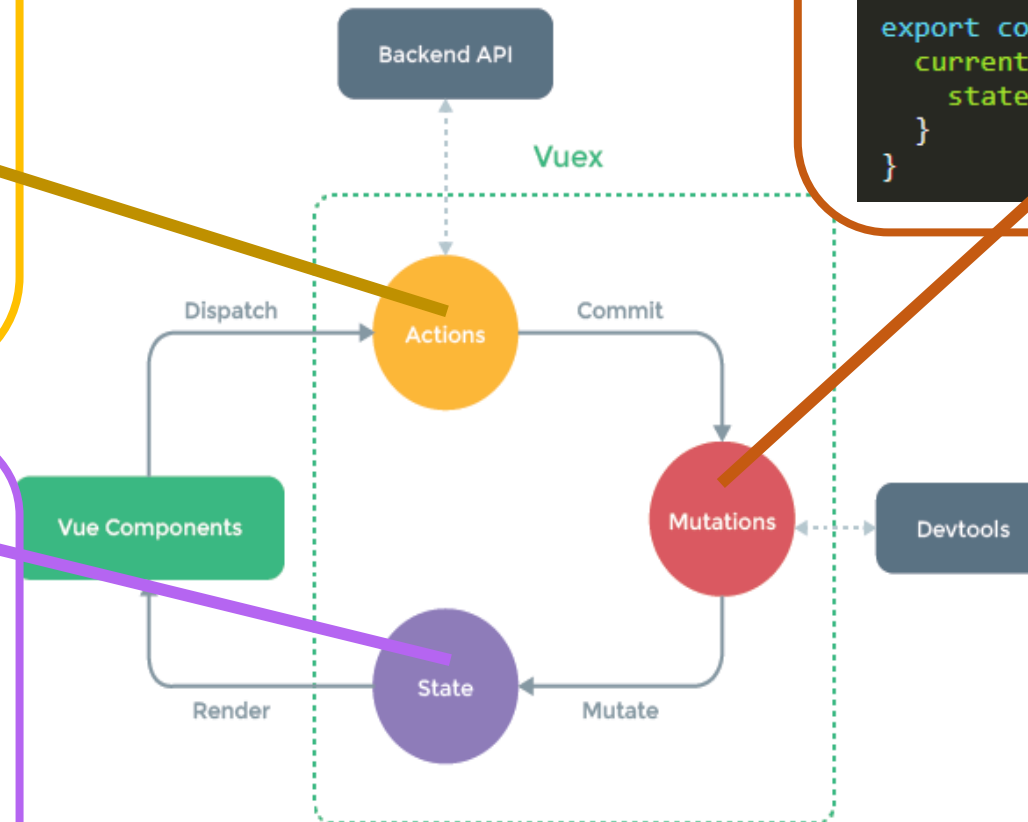
- Stateは、共通で参照するための変数を定義。
- プロジェクトのどこでも参照して使用することが出来る。
- すべてのコンポーネントで、共通の値を使用できる。

```
export const state = () => ({
  account: null,
  isAdmin: null,
  item: null
})
```

Mutation

- Stateの変更を担当する。
- 必ず、Mutationを通して、Stateを変更する必要がある。
- mutations内で、関数を作成。
- commit('関数名', '引数') の方式で、呼び出す。

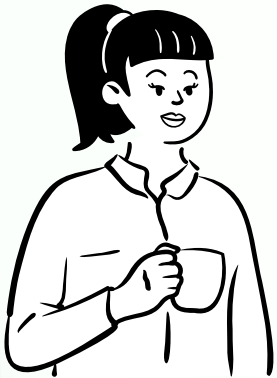
```
export const mutations = {
  currentUser(state, account) {
    state.account = account
  }
}
```



AGE />

2. 開発環境 セットアップ

Node.js インストール



Node.js® は、Chrome の V8 JavaScript エンジン で動作する JavaScript 環境です。

Node.js 18 is now available!

ダウンロード Windows (x64)

16.14.2 LTS

推奨版

18.0.0 最新版

最新の機能

[他のバージョン](#) | [変更履歴](#) | [API ドキュメント](#) | [他のバージョン](#) | [変更履歴](#) | [API ドキュメント](#)

または、LTSのリリーススケジュールをご覧ください

Node.JS公式サイトよりセットアップファイルをダウンロードする
リンク：<https://nodejs.org/ja/>



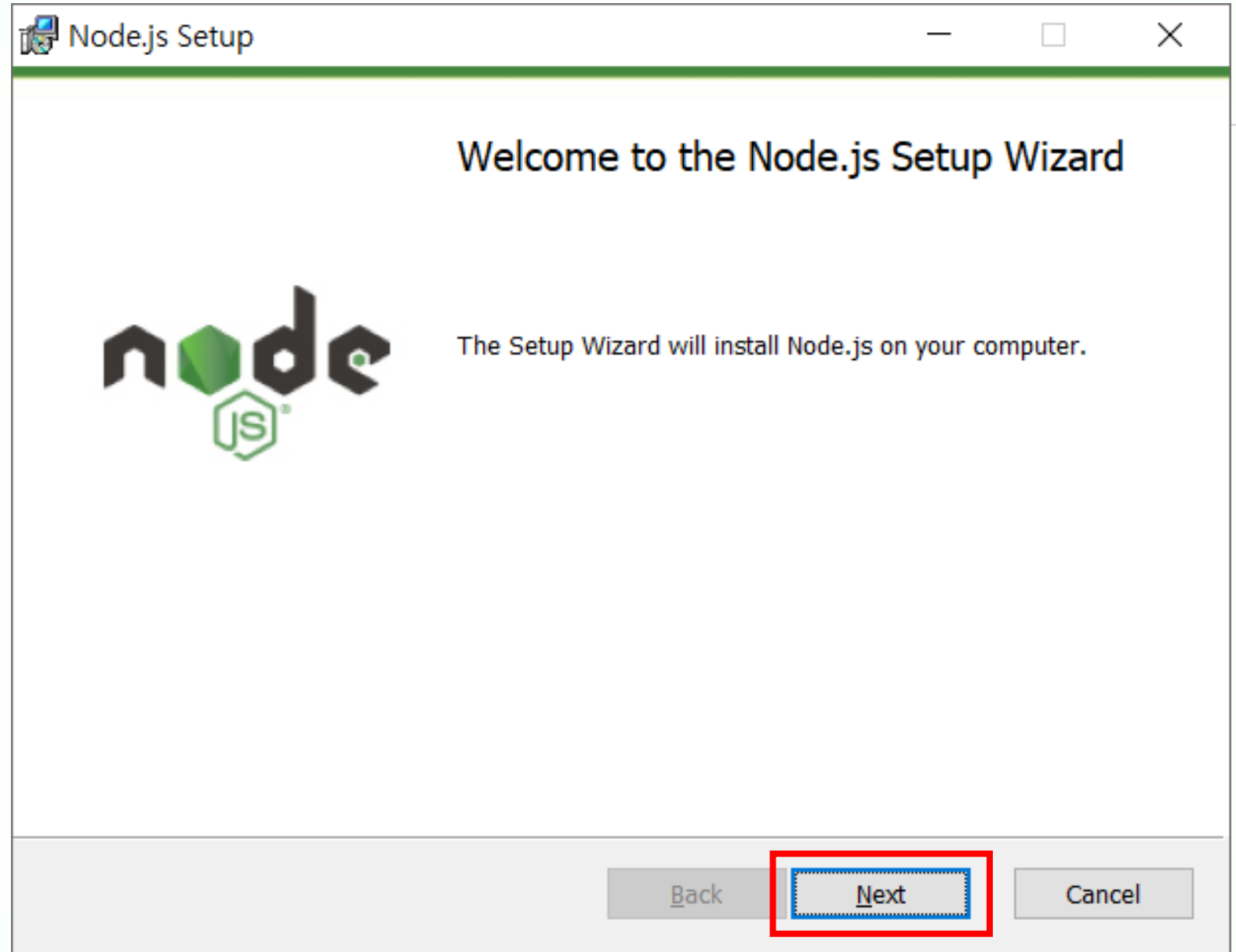
© OpenJS Foundation. All Rights Reserved. Portions of this site originally © Joyent.

Node.js is a trademark of the OpenJS Foundation. Please review the [Trademark List and Trademark Guidelines of the OpenJS Foundation](#).

- [GitHub 上で編集](#)
- [Node.js の不具合を報告](#)
- [サイトの不具合を報告](#)
- [ヘルプ](#)
- [Contributing For Nodejs.org](#)

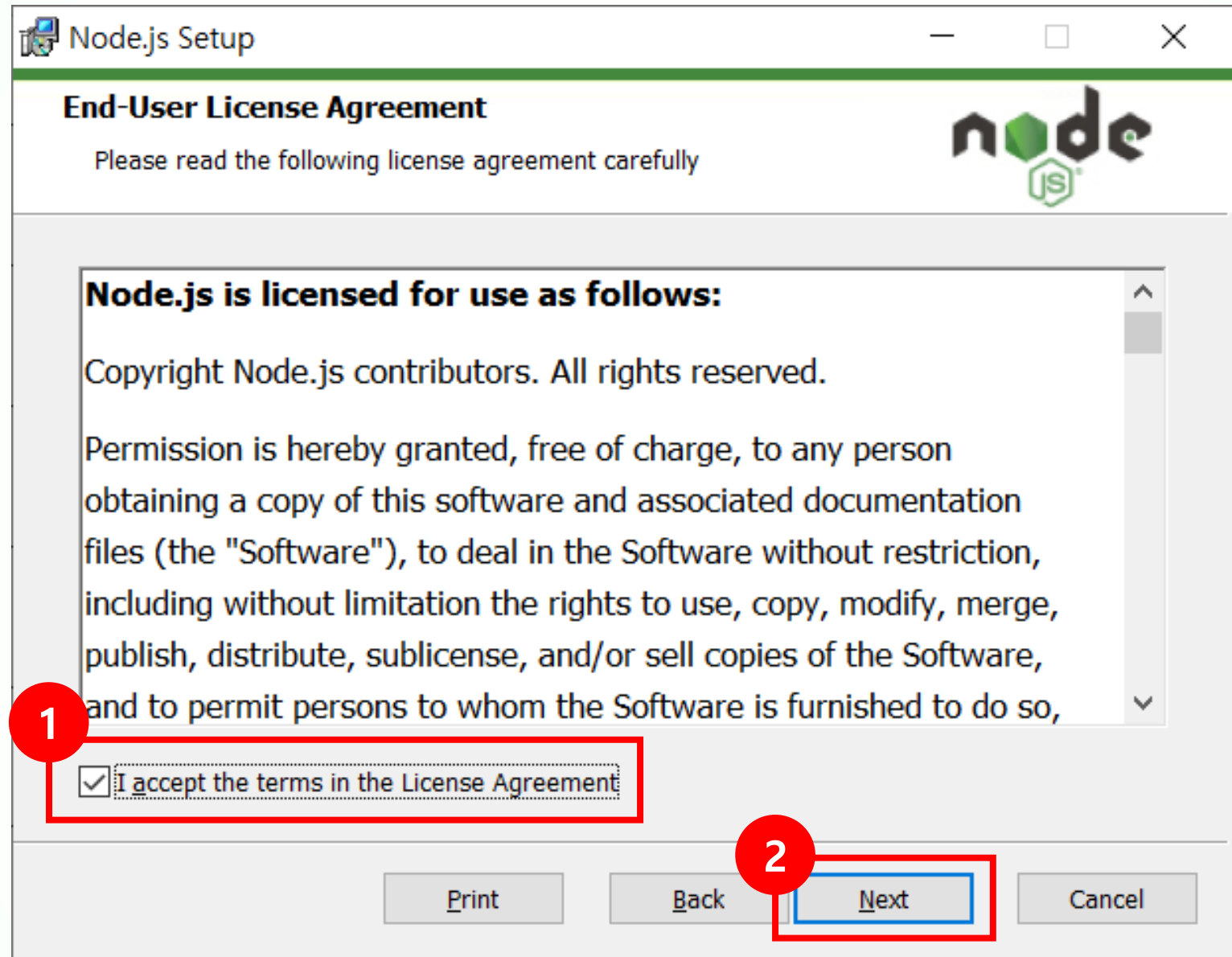
開発環境構成

①Node.jsインストール



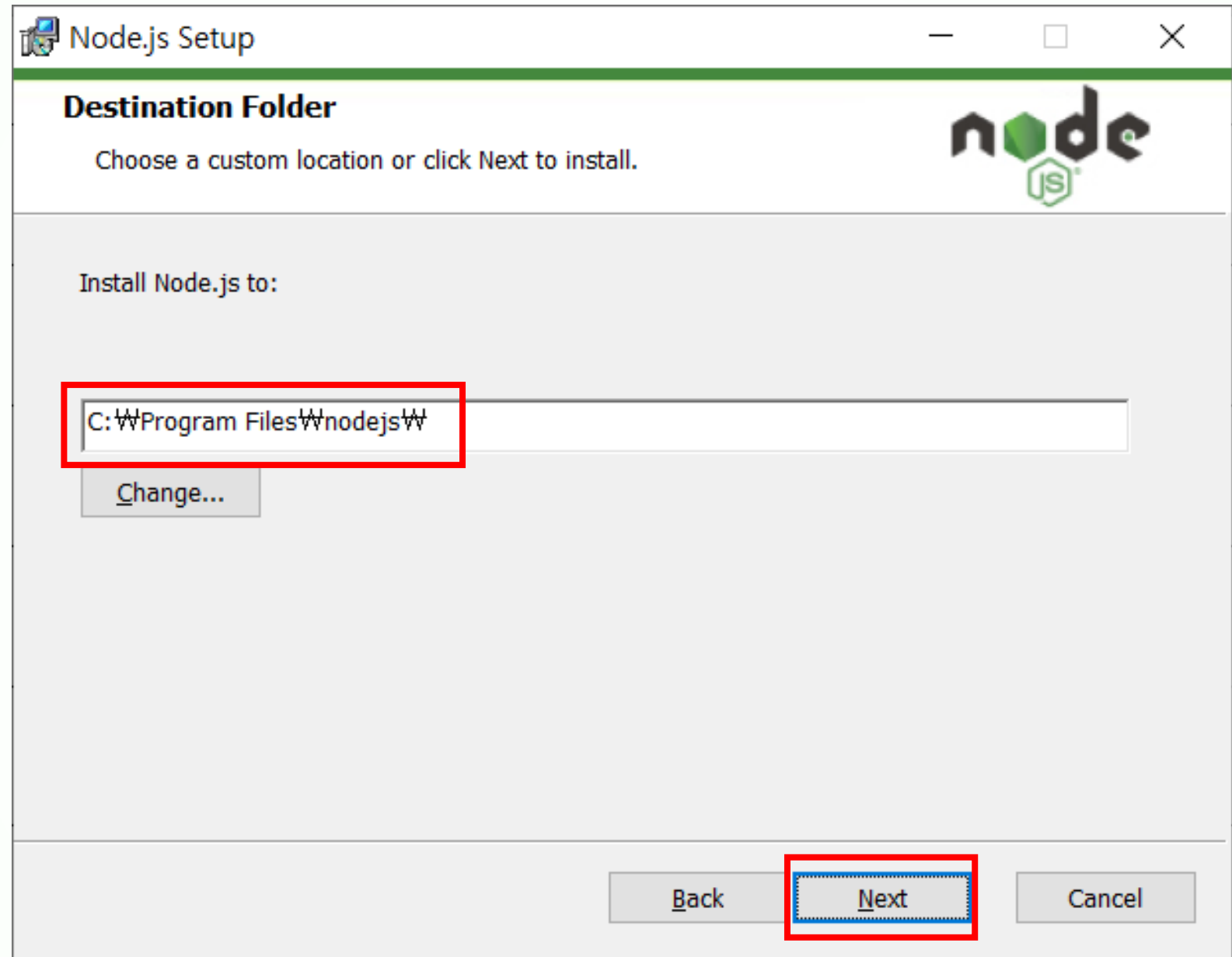
開発環境構成

①Node.jsインストール



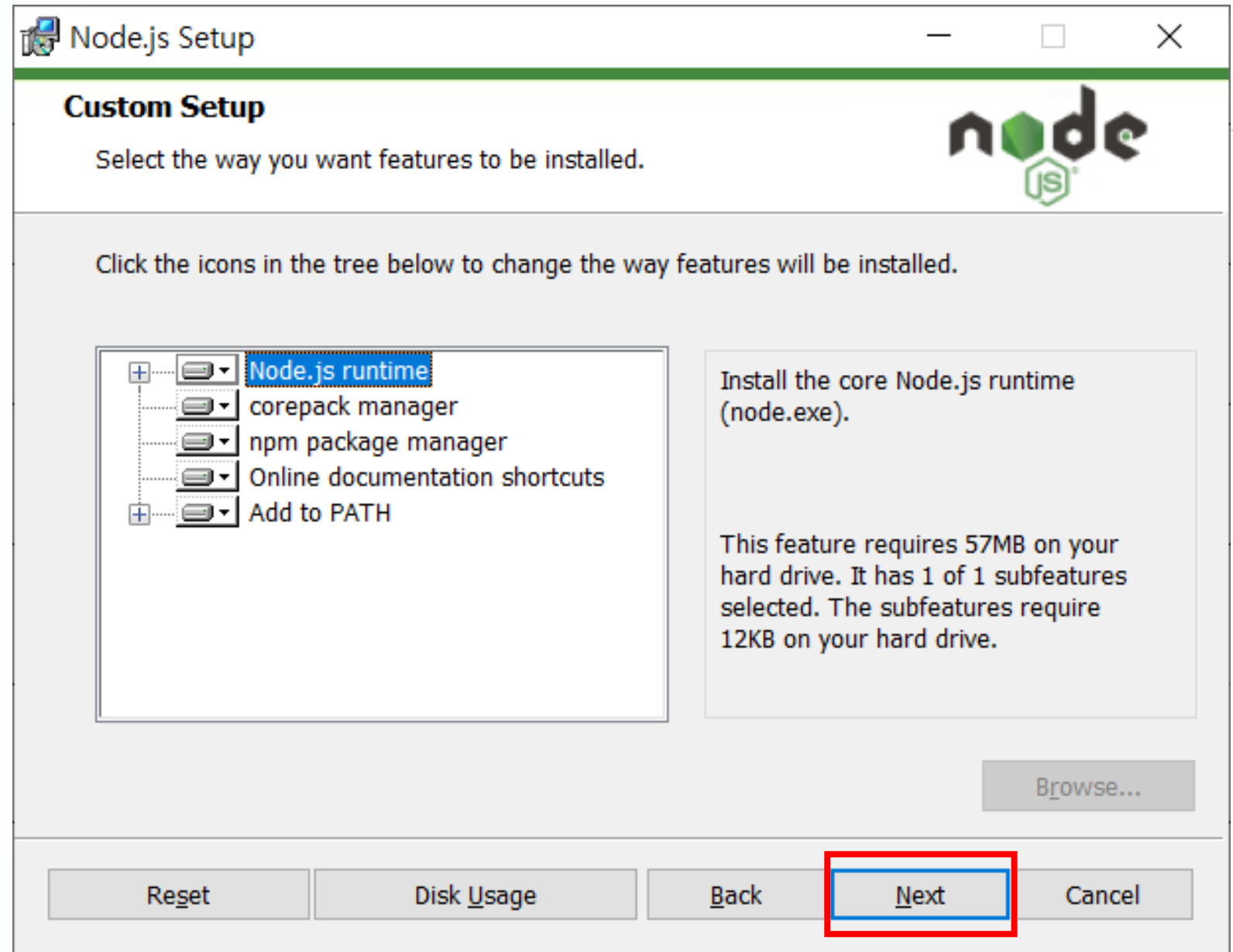
開発環境構成

①Node.jsインストール



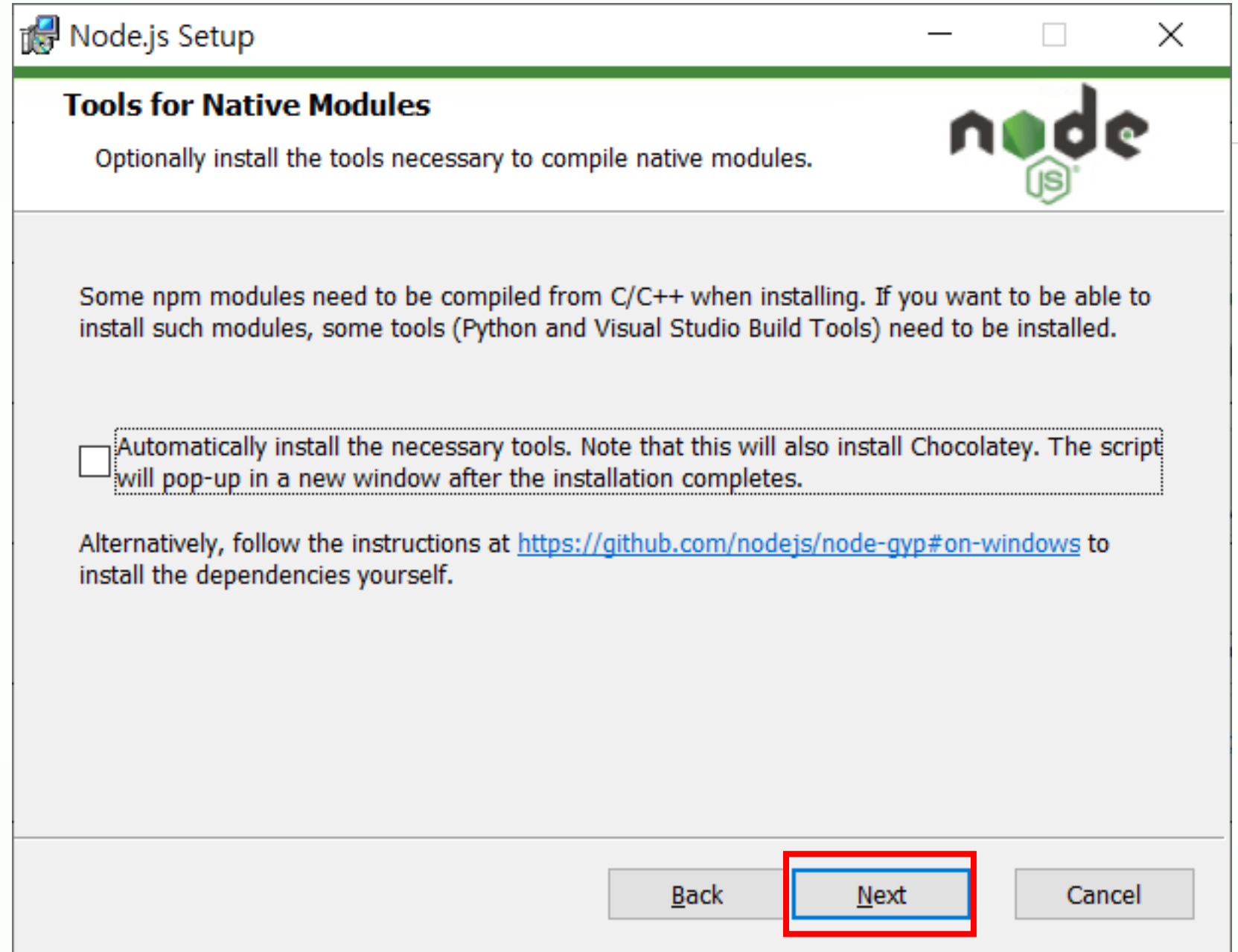
開発環境構成

①Node.jsインストール



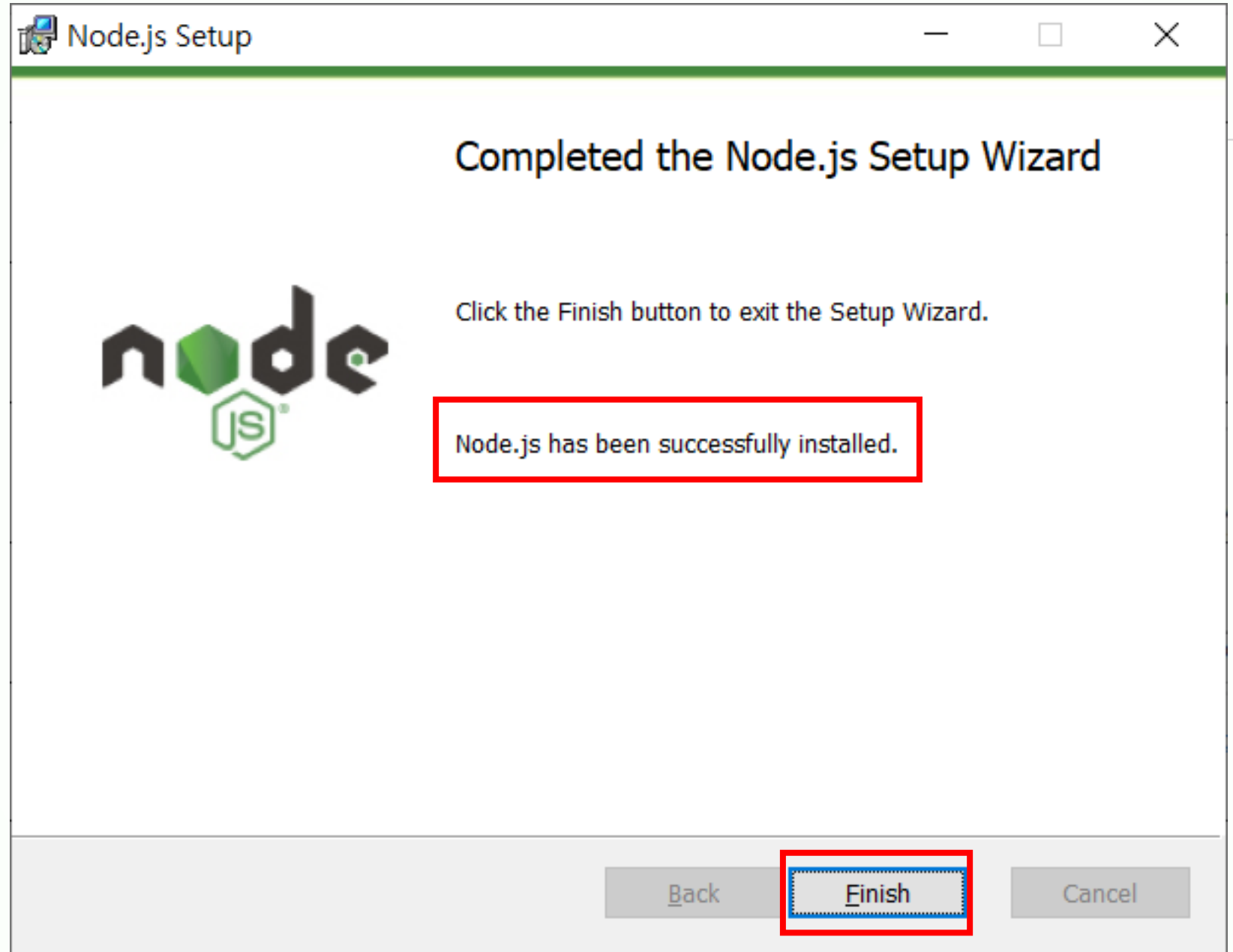
開発環境構成

①Node.jsインストール



開発環境構成

①Node.jsインストール



C:\WINDOWS\system32\cmd.exe

```
C:\Users\admin>node -v  
v16.14.2
```

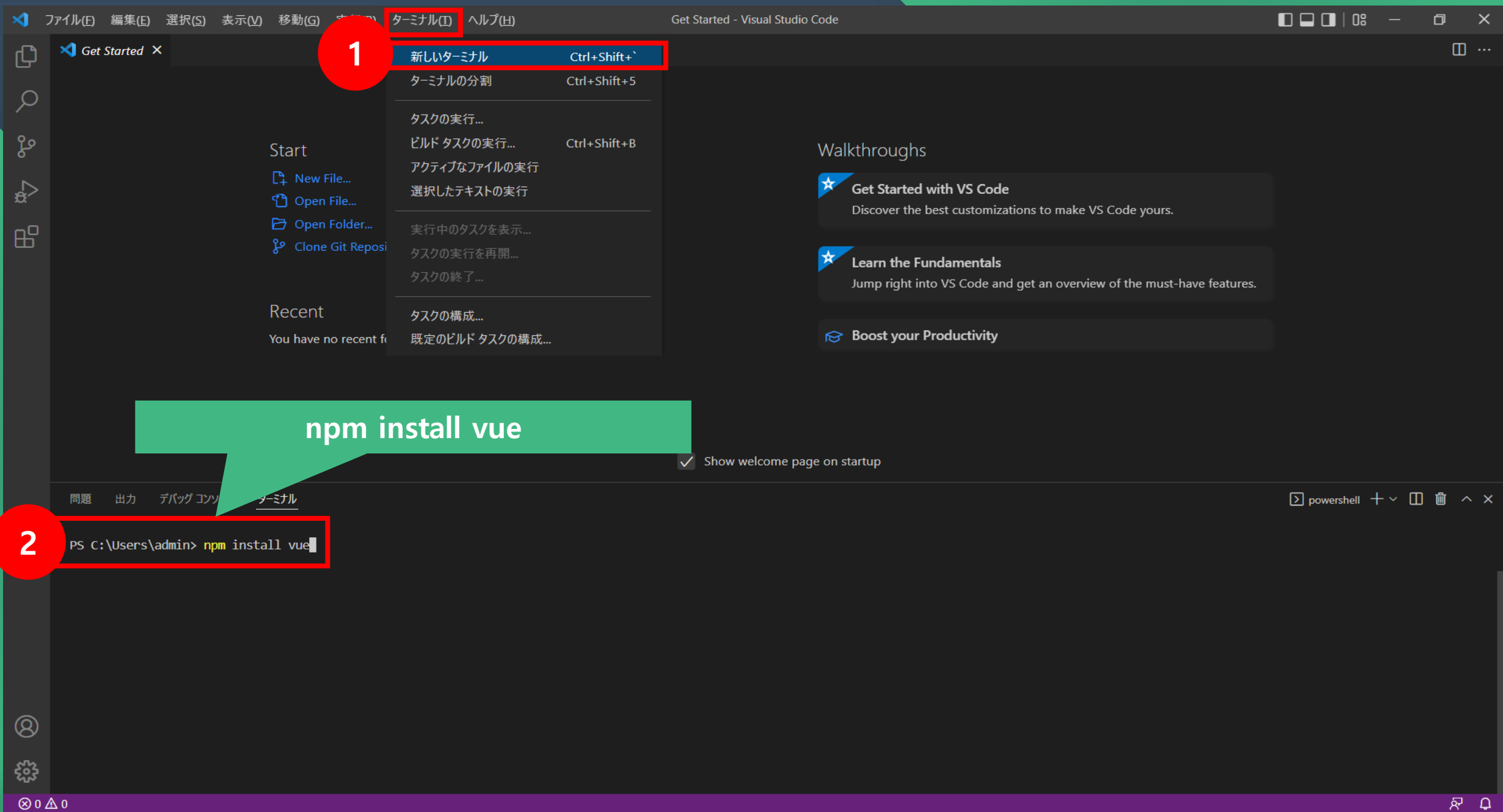
```
C:\Users\admin>
```

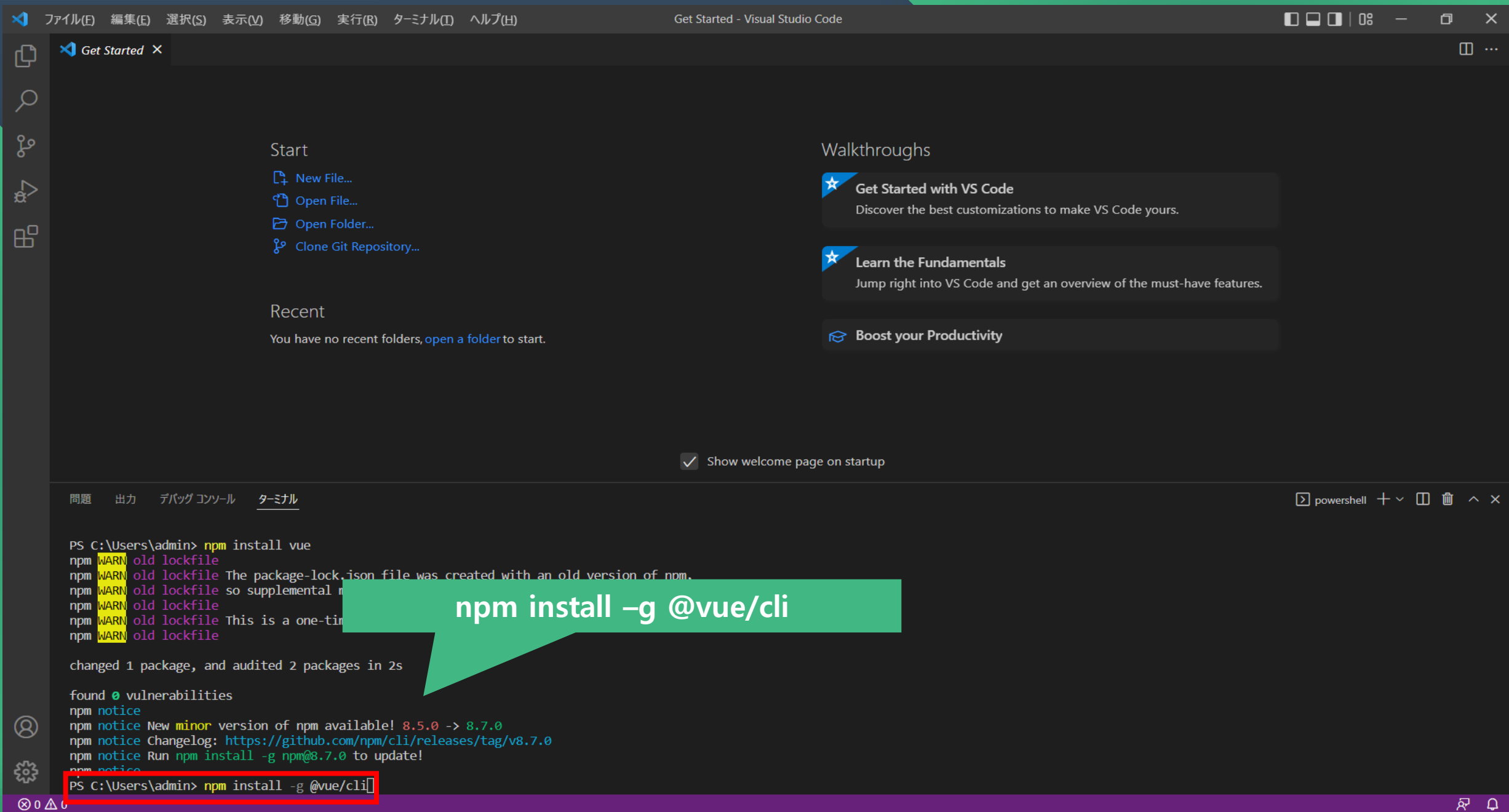
2. 開発環境 セットアップ

Vue.js インストール

Vue.js







C:\WINDOWS\system32\cmd.exe

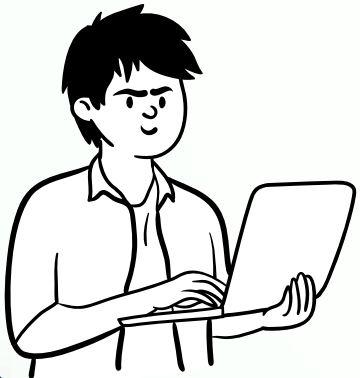
```
C:\Users\admin>vue -V  
@vue/cli 5.0.4
```

```
C:\Users\admin>vue --version  
@vue/cli 5.0.4
```

```
C:\Users\admin>
```


2. 開発環境セットアップ

付加機能インストール










Visual Studio Code


拡張機能

Marketplace で拡張機能を検索する

> インストール済み 2

▼ 推奨 7

-  **Docker** 14.3M ★ 4.5
Makes it easy to create, manage, an...
Microsoft [インストール](#)
-  **Vetur** 10.2M ★ 4.5
Vue tooling for VS Code
Pine Wu [インストール](#)
-  **npm** 5M ★ 3.5
npm support for VS Code
Microsoft [インストール](#)
-  **npm Intellisense** 4.1M ★ 4.5
Visual Studio Code plugin that auto...
Christian Kohler [インストール](#)
-  **Microsoft Edge Tool...** 590K ★ 4.5
Use the Microsoft Edge Tools from ...
Microsoft [インストール](#)
-  **SQLTools** 1.6M ★ 3.5
Database management done right. ...
Matheus Teixeira [インストール](#)
-  **GitLens — Git supe...** 14.2M ★ 4.5
Supercharge Git within VS Code — ...
GitKraken [インストール](#)



すべてのコマンドの表示 [Ctrl](#) + [Shift](#) + [P](#)

ファイルを開く [Ctrl](#) + [O](#)

フォルダーを開く [Ctrl](#) + [K](#) [Ctrl](#) + [O](#)

最近開いた項目 [Ctrl](#) + [R](#)

0 0 0

拡張機能: Vetur - Visual Studio Code

拡張機能: マーケットプレイス

1

vetur

2

Vetur v0.35.0
Pine Wu | 10,329,360 | ★★★★★ (144)
Vue tooling for VS Code

インストール

詳細 機能のコントリビューション 変更ログ

Vetur

passing v0.35.0 10.32M 4.33/5 (144)

Vue tooling for VS Code.

- vls: Vue Language Server
- vti: Vetur Terminal Interface
- Docs

VueConf 2017 [Slide](#) & [Video](#)

Sponsors

I quit my job to travel nomadically, to work on Open Source and to conduct independent study/research.

Vetur is my main focus in Open Source. Your help will alleviate my financial burden and allow me to spend more time working on Vetur. Thank you 🙏

<https://github.com/sponsors/octref>

Thanks to the following companies for supporting Vetur's development:

New Relic
CodeStream

Request and perform code reviews from inside your IDE. Review any code, even if it's a work-in-progress that hasn't been committed yet, and use jump-to-definition, your favorite keybindings, and other IDE tools. [Try it free.](#)

Track and prioritise tech debt and maintenance issues, straight from your IDE. Bookmark

カテゴリ
Programming Languages

リソース
マーケットプレイス
リポジトリ
ライセンス

マーケットプレイス情報

リリース日	2016. 11. 6. 11時 45分 23초
最終更新	2021. 10. 16. 15시 8분 25초
識別子	octref.vetur

0 0 0

Visual Studio Code interface showing the process of installing the HTML CSS Support extension.

Left Sidebar (Extensions: Marketplace):

- Search: `html css support`
- Results:
 - HTML CSS Support** (v1.11.0) by `ecmel` (9.5M, 3.5 stars) [インストール](#)
 - IntelliSense for CSS ... (v3.5) by `Zignd` (4.5M, 3.5 stars) [インストール](#)
 - JS-CSS-HTML Format... (v1.5) by `lonely` (3M, 1.5 stars) [インストール](#)
 - CSS Peek (v3.5) by `Pranay Prakash` (2.8M, 3.5 stars) [インストール](#)
 - HTML Snippets (v4) by `Mohamed Abusaid` (7.5M, 4 stars) [インストール](#)
 - Kite AutoComplete AI... (v3) by `Kite` (3.4M, 3 stars) [インストール](#)
 - Language Support ... (v3.5) by `Red Hat` (15.4M, 3.5 stars) [インストール](#)
 - HTML Preview (v3.5) by `Thomas Haakon Townsend` (1.8M, 3.5 stars) [インストール](#)
 - HTML Boilerplate (v5) by `sidthesloth` (1.4M, 5 stars) [インストール](#)
 - Live HTML Previewer (v2) by `Harshdeep Gupta` (855K, 2 stars) [インストール](#)
 - CSS Formatter (v4.5) by `Martin Aeschlimann` (1.1M, 4.5 stars) [インストール](#)
 - Spring Initializr Java... (v3.5) by `Microsoft` (1.4M, 3.5 stars) [インストール](#)

Main Panel (HTML CSS Support extension details):

- HTML CSS Support** (v1.11.0) by `ecmel` (9,583,343 downloads, 4.5 stars (85 reviews))
- [インストール](#) (highlighted with a red box and '2')
- [詳細](#) | [機能のコントリビューション](#) | [変更ログ](#)
- ### Visual Studio Code CSS Intellisense for HTML
- HTML `id` and `class` attribute completion for Visual Studio Code.
- Features**
 - HTML `id` and `class` attribute completion.
 - Supports linked and embedded style sheets.
 - Supports template inheritance.
 - Supports additional style sheets.
 - Supports other HTML like languages.
 - Validates CSS selectors on demand.
- Usage**

You can view a list of `id` and `class` attribute suggestions via `ctrl + space`.
- Linked and Embedded Style Sheets**

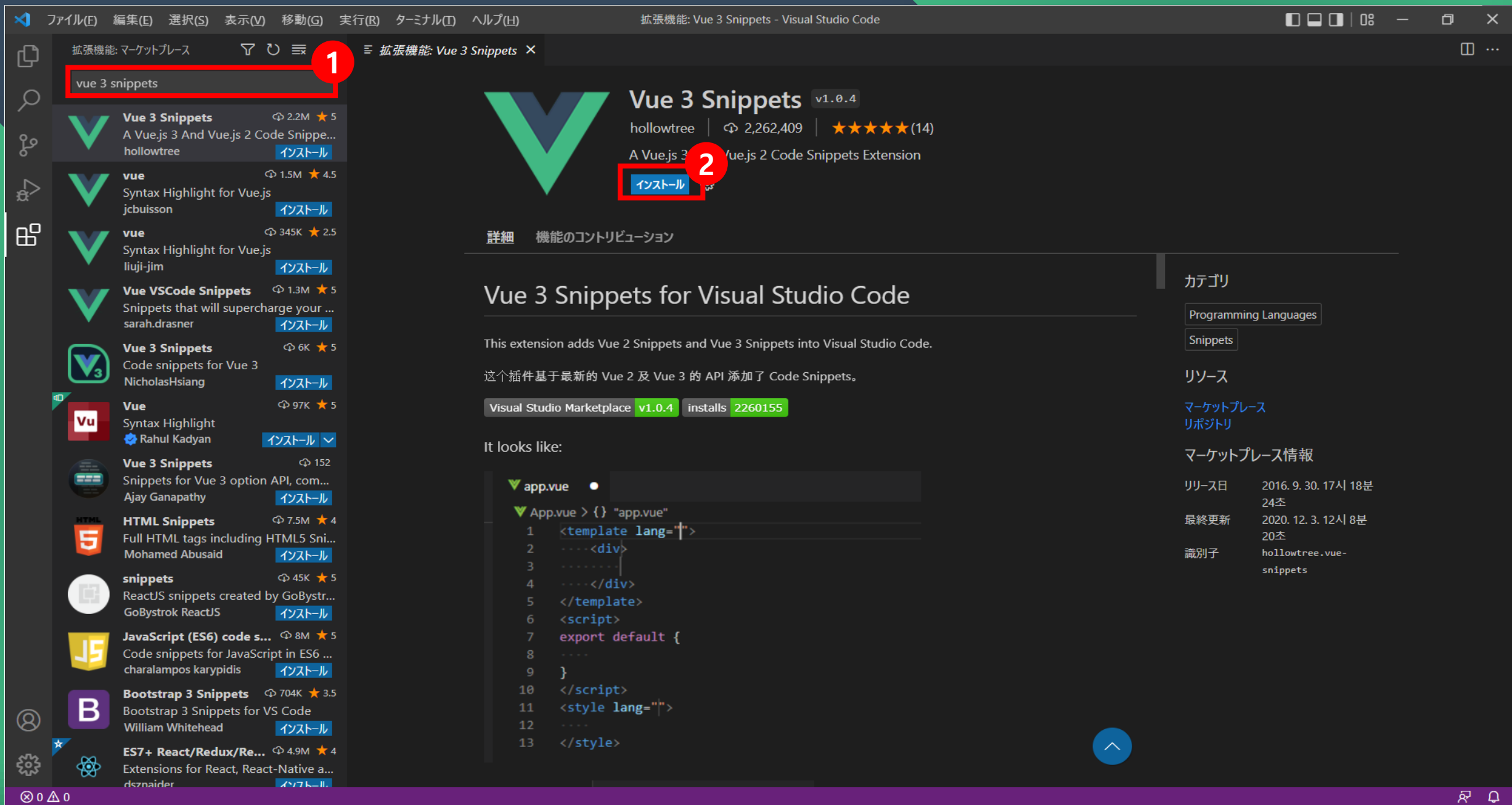
Linked [`<link rel="stylesheet">`] and embedded [`<style></style>`] style sheets are used in completion for `id` and `class` attributes. Links support local and remote files. Absolute local file paths are relative to the workspace folder while others are relative to HTML file:
- index.html**

```
<!DOCTYPE html>
```

Right Sidebar (Metadata):

- カテゴリ**
 - Programming Languages
- リソース**
 - [マーケットプレイス](#)
 - [リポジトリ](#)
 - [ライセンス](#)
- マーケットプレイス情報**

リリース日	2016. 7. 15. 6時 2分 33 秒
最終更新	2022. 2. 6. 20時 26分 34 秒
識別子	ecmel.vscode-html-css



拡張機能: マーケットプレイス

vue 3 snippets

1

Vue 3 Snippets 2.2M ★ 5
A Vue.js 3 And Vue.js 2 Code Snippets Extension
hollowtree インストール

vue 1.5M ★ 4.5
Syntax Highlight for Vue.js
jcbuisson インストール

vue 345K ★ 2.5
Syntax Highlight for Vue.js
liuji-jim インストール

Vue VSCode Snippets 1.3M ★ 5
Snippets that will supercharge your ...
sarah.drasner インストール

Vue 3 Snippets 6K ★ 5
Code snippets for Vue 3
NicholasHsiang インストール

Vue 97K ★ 5
Syntax Highlight
Rahul Kadyan インストール

Vue 3 Snippets 152
Snippets for Vue 3 option API, com...
Ajay Ganapathy インストール

HTML Snippets 7.5M ★ 4
Full HTML tags including HTML5 Sni...
Mohamed Abusaid インストール

snippets 45K ★ 5
ReactJS snippets created by GoBystr...
GoBystrok ReactJS インストール

JavaScript (ES6) code snippets 8M ★ 5
Code snippets for JavaScript in ES6 ...
charalampos karypidis インストール

Bootstrap 3 Snippets 704K ★ 3.5
Bootstrap 3 Snippets for VS Code
William Whitehead インストール

ES7+ React/Redux/Re... 4.9M ★ 4
Extensions for React, React-Native a...
dsznajder インストール

拡張機能: Vue 3 Snippets

Vue 3 Snippets v1.0.4
hollowtree | 2,262,409 | ★★★★★ (14)
A Vue.js 3 And Vue.js 2 Code Snippets Extension

2

インストール

詳細 機能のコントリビューション

Vue 3 Snippets for Visual Studio Code

This extension adds Vue 2 Snippets and Vue 3 Snippets into Visual Studio Code.

这个插件基于最新的 Vue 2 及 Vue 3 的 API 添加了 Code Snippets。

Visual Studio Marketplace v1.0.4 installs 2260155

It looks like:

```
app.vue
App.vue > {} "app.vue"
1 <template lang="html">
2   <div>
3     <div>
4   </div>
5 </template>
6 <script>
7   export default {
8     ...
9   }
10 </script>
11 <style lang="css">
12   ...
13 </style>
```

カテゴリ

Programming Languages

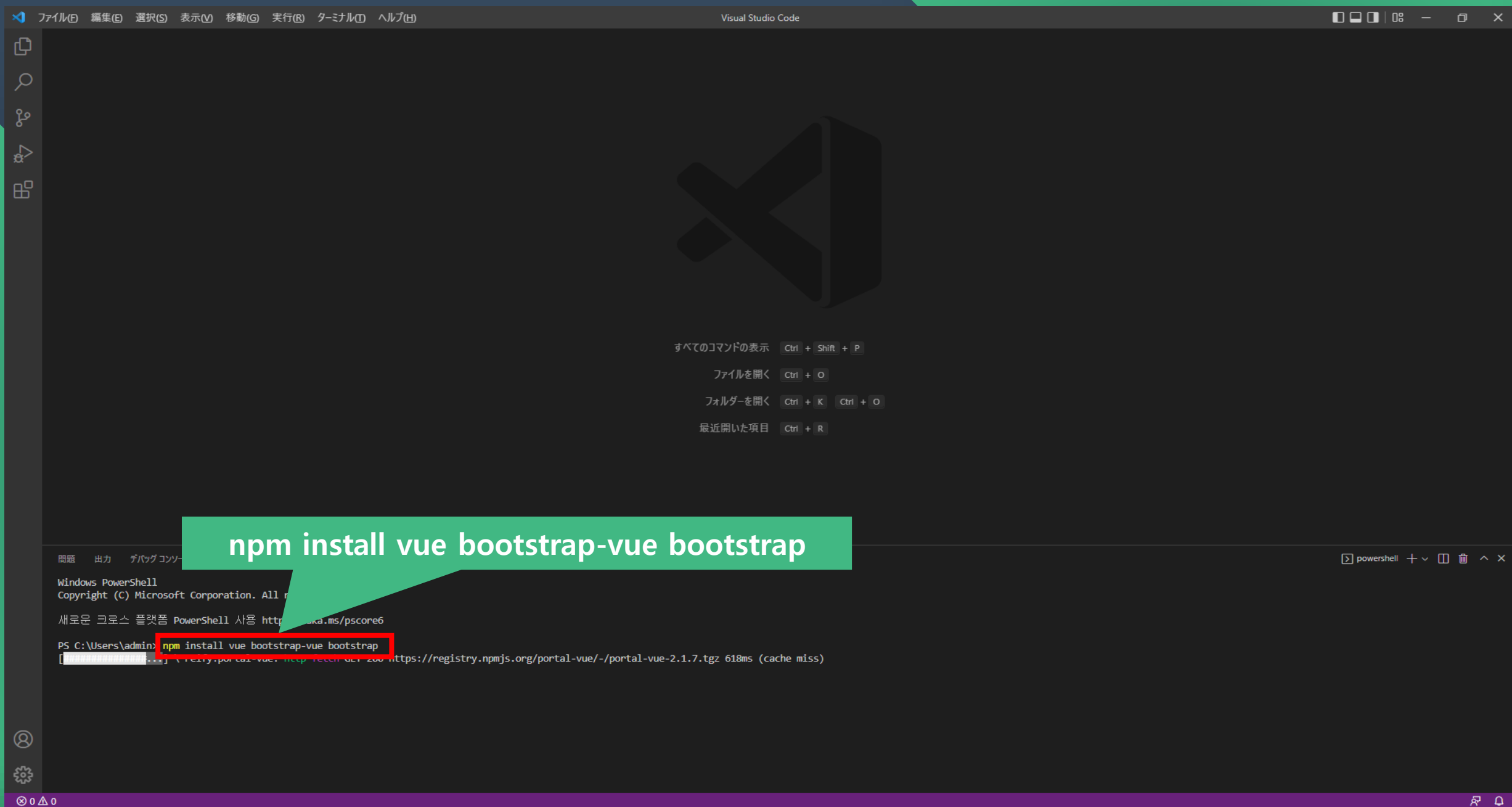
Snippets

リソース

マーケットプレイス
リポジトリ

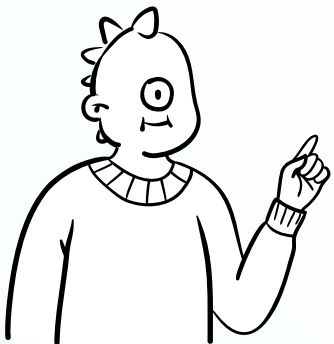
マーケットプレイス情報

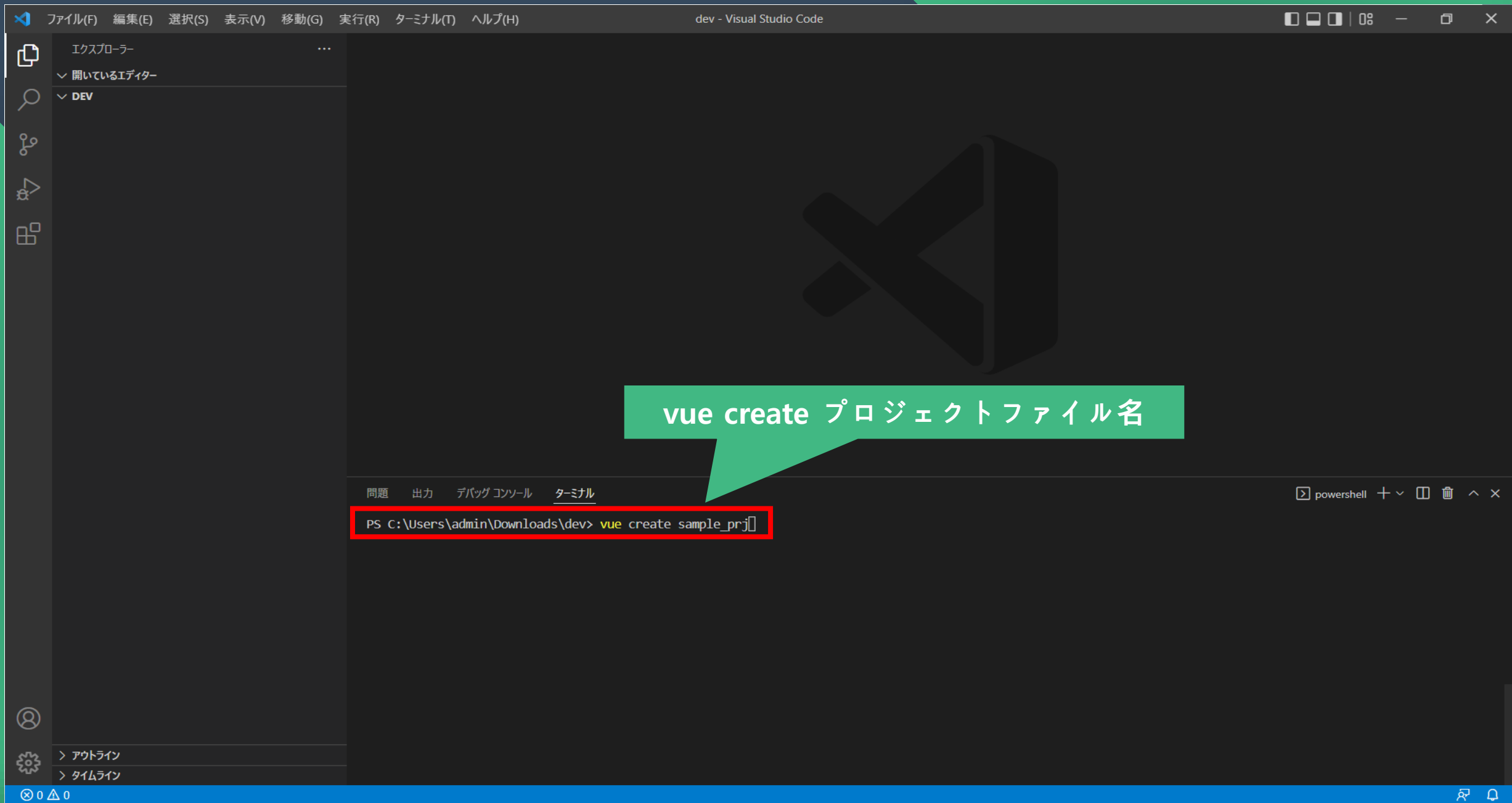
リリース日	2016. 9. 30. 17時 18分 24초
最終更新	2020. 12. 3. 12시 8분 20초
識別子	hollowtree.vue-snippets

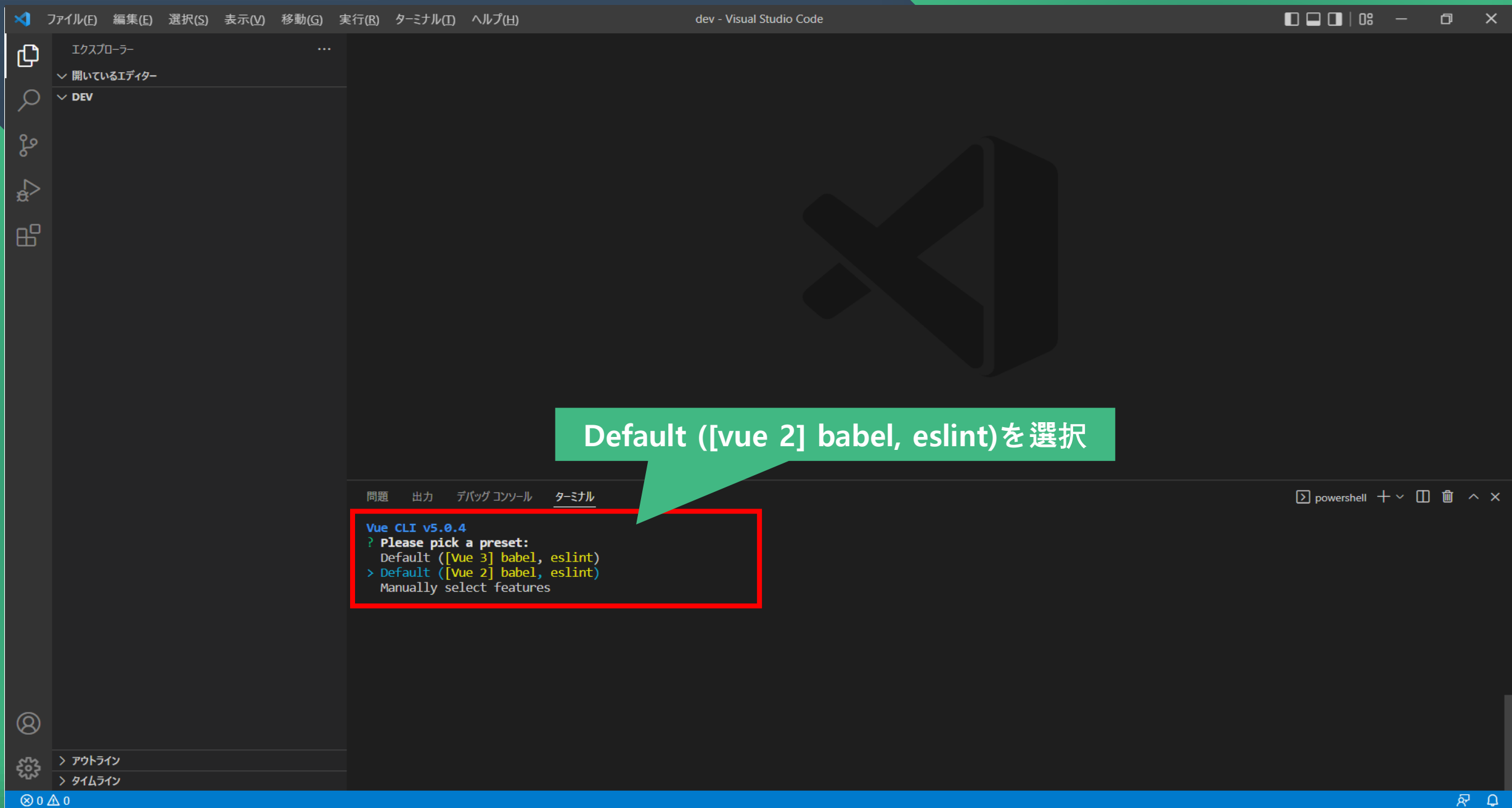


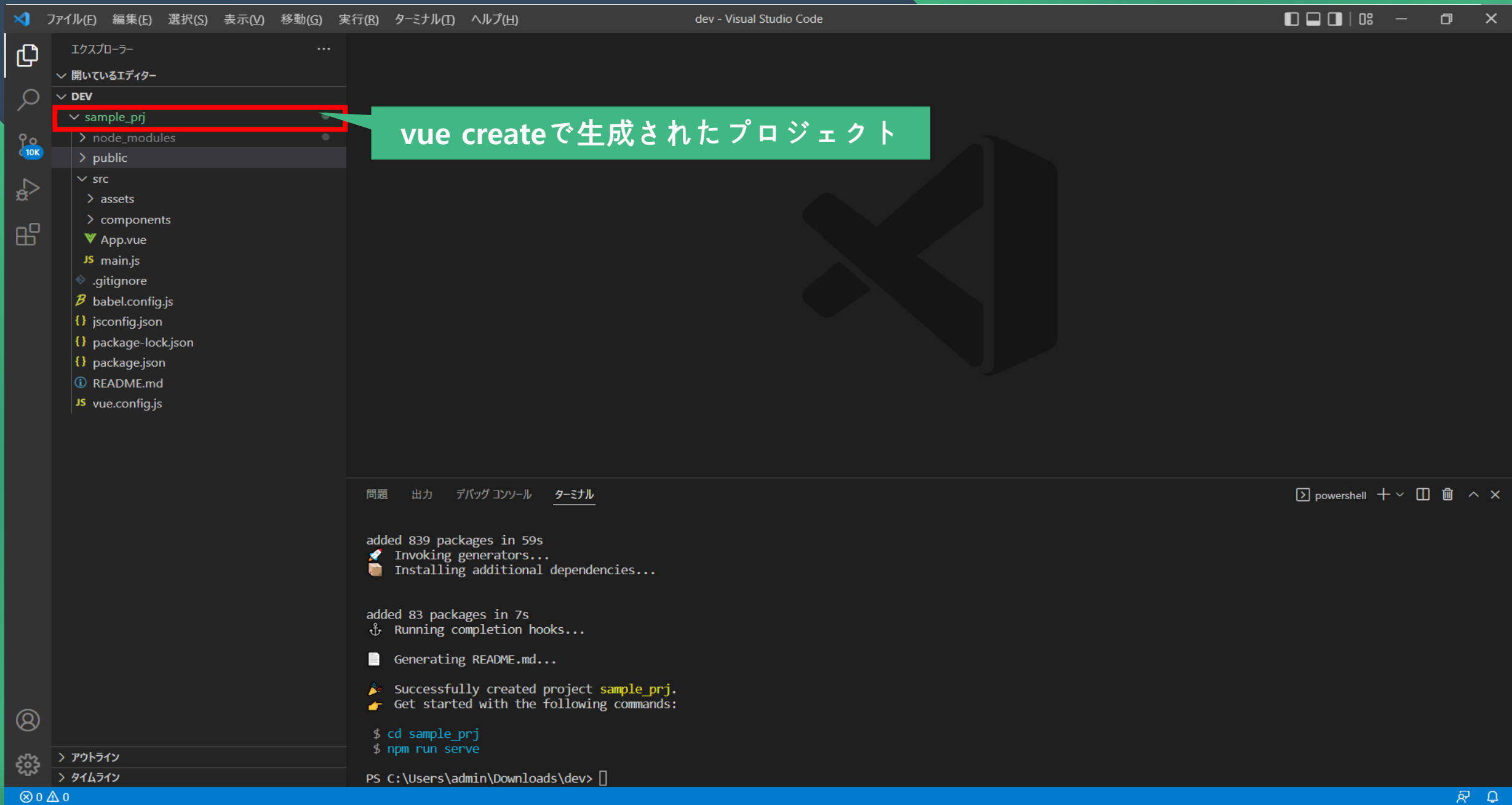
2. 開発環境セットアップ

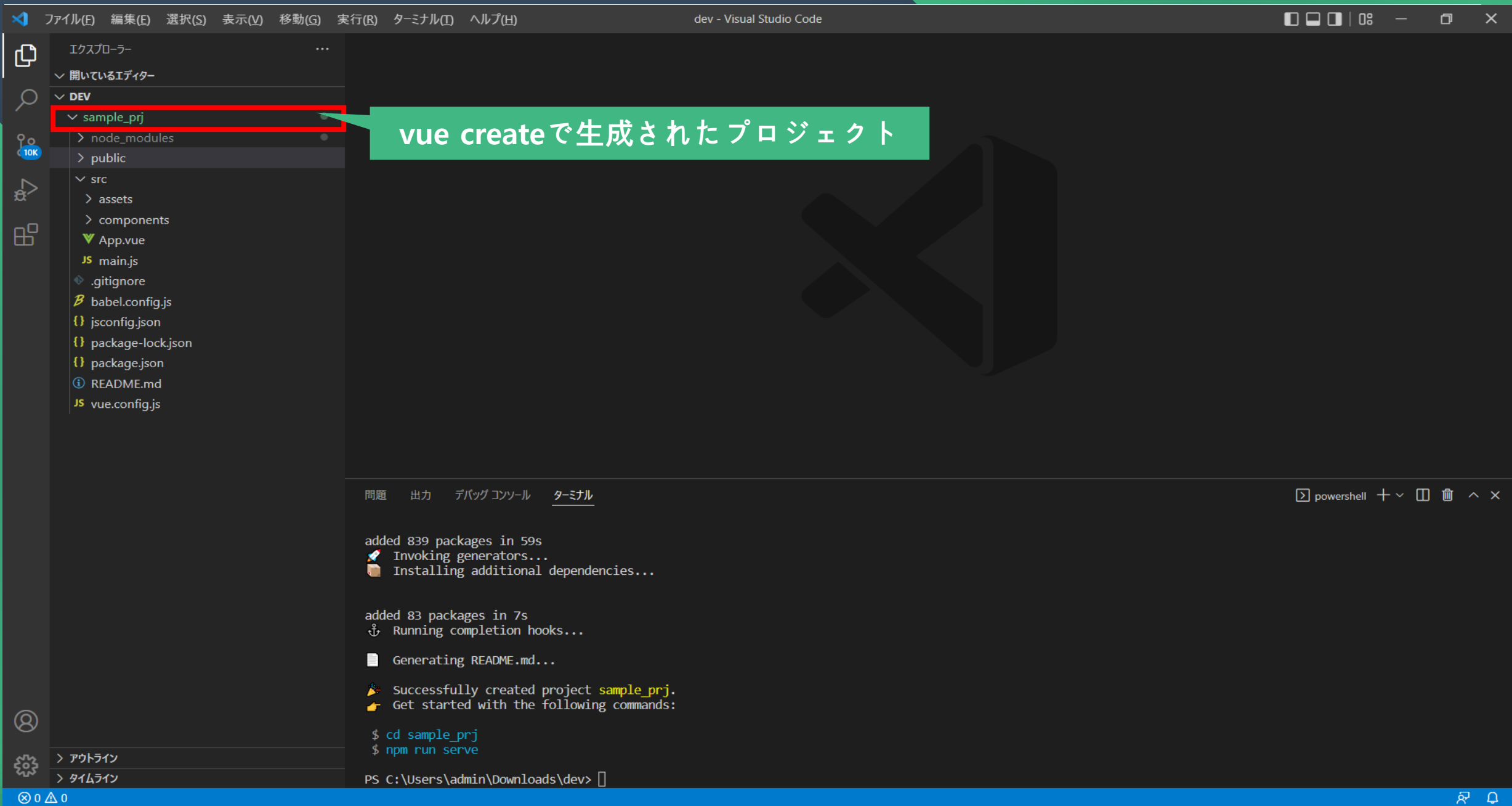
Vueプロジェクト生成











Visual Studio Code interface showing the Explorer sidebar on the left. The project structure is as follows:

- sample_prj
 - node_modules
 - public
 - src
 - assets
 - components
 - App.vue
 - main.js
 - .gitignore
 - babel.config.js
 - jsconfig.json
 - package-lock.json
 - package.json
 - README.md
 - vue.config.js

A green callout box points to the **sample_prj** folder in the Explorer sidebar, containing the text: **vue createで生成されたプロジェクト**

The terminal output at the bottom shows the successful creation of the project:

```
added 839 packages in 59s
Invoking generators...
Installing additional dependencies...

added 83 packages in 7s
Running completion hooks...

Generating README.md...

Successfully created project sample_prj.
Get started with the following commands:

$ cd sample_prj
$ npm run serve

PS C:\Users\admin\Downloads\dev>
```

Visual Studio Code - App.vue - sample_prj

プロジェクトで使われるツール

その他ファイル

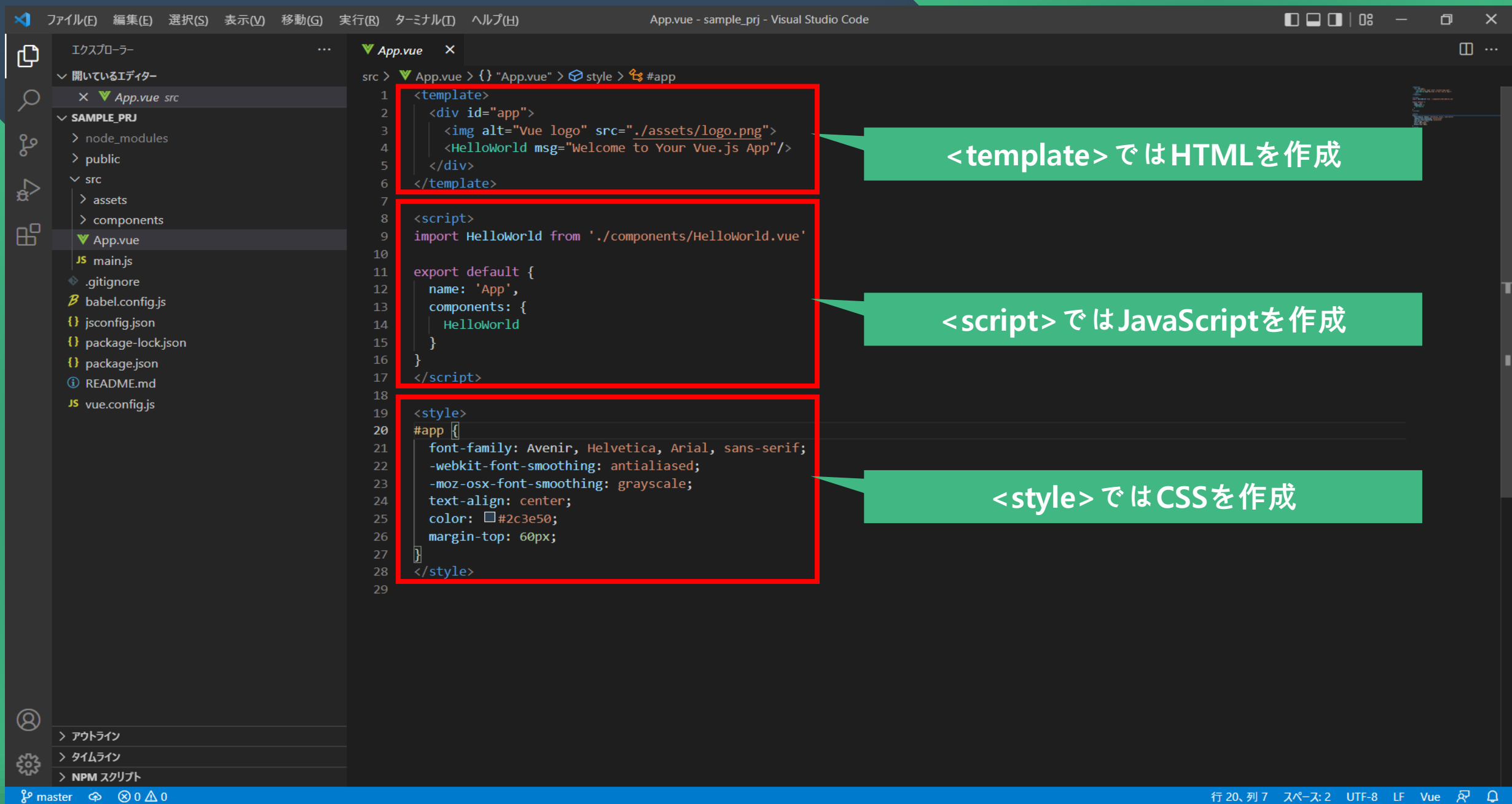
ソースファイル

App.vueで作成した内容はHTMLファイルに変換されウェブブラウザに表示される

App.vue

Index.html

行 20、列 7 スペース: 2 UTF-8 LF Vue



The image shows the Visual Studio Code editor with the `App.vue` file open. The file is divided into three sections, each highlighted with a red box and a corresponding green callout box:

- Template Section (Lines 1-6):** Contains HTML-like markup for the component's structure. The callout box states: `<template>` ではHTMLを作成 (Create HTML with `<template>`).
- Script Section (Lines 8-17):** Contains JavaScript code for the component's logic. The callout box states: `<script>` ではJavaScriptを作成 (Create JavaScript with `<script>`).
- Style Section (Lines 19-28):** Contains CSS code for the component's styling. The callout box states: `<style>` ではCSSを作成 (Create CSS with `<style>`).

The `App.vue` file content is as follows:

```
src > App.vue > {} "App.vue" > style > #app
1 <template>
2   <div id="app">
3     
4     <HelloWorld msg="Welcome to Your Vue.js App"/>
5   </div>
6 </template>
7
8 <script>
9   import HelloWorld from './components/HelloWorld.vue'
10
11   export default {
12     name: 'App',
13     components: {
14       HelloWorld
15     }
16   }
17 </script>
18
19 <style>
20 #app {
21   font-family: Avenir, Helvetica, Arial, sans-serif;
22   -webkit-font-smoothing: antialiased;
23   -moz-osx-font-smoothing: grayscale;
24   text-align: center;
25   color: #2c3e50;
26   margin-top: 60px;
27 }
28 </style>
29
```

At the bottom of the screen, the status bar shows: 行 20、列 7 スペース: 2 UTF-8 LF Vue

App.vue - sample_prj - Visual Studio Code

Explorer: App.vue

```
src > App.vue > {} "App.vue" > style
1 <template>
2   <div id="app">
3     
4     <HelloWorld msg="Welcome to Your Vue.js App"/>
5   </div>
6 </template>
7
8 <script>
9 import HelloWorld from '../components/HelloWorld.vue'
10
11 export default {
12   name: 'App',
13   components: {
14     HelloWorld
15   }
16 }
17 </script>
18
19 <style>
20 #app {
21   font-family: Avenir, Helvetica, Arial, sans-serif;
22   -webkit-font-smoothing: antialiased;
```

Terminal:

```
PS C:\Users\admin\Downloads\dev\sample_prj> npm run serve
> sample_prj@0.1.0 serve
> vue-cli-service serve

INFO Starting development server...

DONE Compiled successfully in 9125ms

App running at:
- Local: http://localhost:8080/
- Network: http://192.168.3.10:8080/

Note that the development build is not optimized.
To create a production build, run npm run build.
```

npm run serve を入力

「ctrl + click」または住所をブラウザに入力

行 28, 列 9 スペース: 2 UTF-8 LF Vue



Welcome to Your Vue.js App

For a guide and recipes on how to configure / customize this project,
check out the [vue-cli documentation](#).

Installed CLI Plugins

[babel](#) [eslint](#)

Essential Links

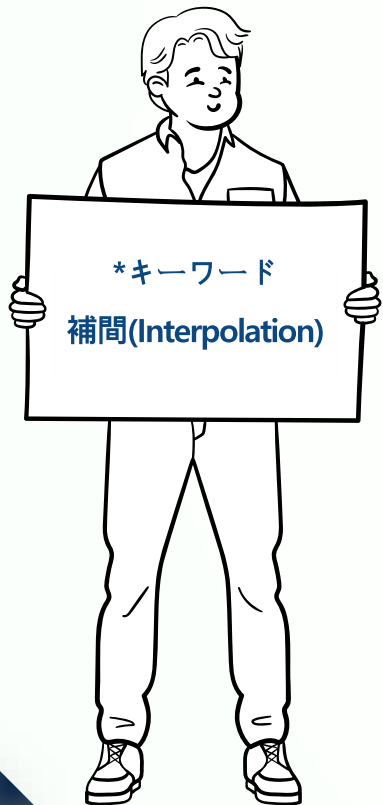
[Core Docs](#) [Forum](#) [Community Chat](#) [Twitter](#) [News](#)

Ecosystem

[vue-router](#) [vuex](#) [vue-devtools](#) [vue-loader](#) [awesome-vue](#)

3. Vue基本構文

Chapter1 【データバインディング】



2.Vue基本構文

Chapter1 【データバインディング】

既存のHTMLコーディング

```
<html>
  <body>
    <span id="myid">こんにちは！</span>
  </body>
</html>

<script>
  console.log(document.getElementById("myId"));
</script>
```

Vs

Vueを適用したHTMLコーディング

```
<template>
  <div id="app">
    <span>{{ msg }}</span>
  </div>
</template>

<script>
export default {
  name: "App",
  components: {},
  data() {
    return {
      msg: "こんにちは",
    };
  },
};
</script>
```

HTML

```
<template>
  <div id="app">
    <span>メッセージ: {{ msg }}</span>
    <span>メッセージ: 静的なテキストです。</span>
  </div>
</template>

<script>
export default {
  name: "App",
  components: {},
  data() {
    return {
      msg: "動的なテキストです。",
    };
  },
};
</script>
```

結果

メッセージ: 動的なテキストです。

メッセージ: 静的なテキストです。

Mustacheタグは、対応するオブジェクト(msg)のプロパティの値に置き換えられます。
また、msgのプロパティが変更される時、それに応じて更新されます。



- サンプルコード: <https://github.com/github-Ahn/PSSMiniPrj>
- プロジェクト名: sample_prj_chapter1

3. Vue基本構文

chapter2

【リストレンダリング】

*キーワード

v-for

配列(Array)



2.Vue基本構文

Chapter2 【リストレンダリング】

```
<table id="itemTable">
  <tr>
    <td>{{ item1 }}</td>
    <td>¥{{ price1 }}</td>
  </tr>
  <tr>
    <td>{{ item2 }}</td>
    <td>¥{{ price2 }}</td>
  </tr>
  <tr>
    <td>{{ item3 }}</td>
    <td>¥{{ price3 }}</td>
  </tr>
  <tr>
    <td>{{ item4 }}</td>
    <td>¥{{ price4 }}</td>
  </tr>
</table>
```

アイテムが増える度に
一個一個ハードコーディング?

領収書

〇〇スーパー〇〇店
TEL000-000-000
20xx年04月01日 (月) 13:27

ネギ	¥ 190
ジャガイモ	¥ 208
トマト	¥ 435
ニンジン	¥ 372

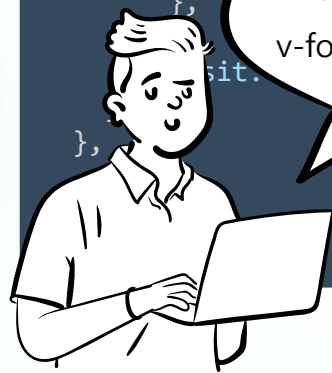
合計	¥ 1205
お預かり	¥ 1300
お釣り	¥ 95

2.Vue基本構文

Chapter2 【リストレンダリング】

```
<table id="itemTable">
  <tr v-for="(item, index) in shoppingList" :key="index">
    <td>{{ item.name }}</td>
    <td>¥{{ item.price }}</td>
  </tr>
</table>

export default {
  name: "App",
  components: {},
  data() {
    return {
      shoppingList: [
        {
          name: "ネギ",
          price: 190,
        },
        {
          name: "ジャガイモ",
          price: 208,
        },
        {
          name: "トマト",
          price: 435,
        },
        {
          name: "ニンジン",
          price: 372,
        },
      ],
    }
  },
}
```



アイテムをリストに作って
v-forを利用すれば、簡単！

領収書

〇〇スーパー〇〇店
TEL000-000-000
20xx年04月01日 (月) 13:27

ネギ	¥ 190
ジャガイモ	¥ 208
トマト	¥ 435
ニンジン	¥ 372

合計	¥ 1205
お預かり	¥ 1300
お釣り	¥ 95

2.Vue基本構文

Chapter2 【リストレンダリング】

V-forの使い方①

HTML

```
<ul id="example-1">
  <li v-for="(msg,index) in messages" :key="index">
    {{ msg }}
  </li>
</ul>
```

Diagram illustrating the V-for loop structure in HTML. A red circle with the number 1 points to the `messages` variable in the `in` clause. A red circle with the number 2 points to the `{{ msg }}` template expression.

JS

```
export default {
  name: "App",
  components: {},
  data() {
    return {
      messages: ["Foo", "Bar"],
    };
  },
};
```

Diagram illustrating the JavaScript data source for the V-for loop. A red arrow points from the `messages` array in the `data()` function to the `messages` variable in the HTML template.

v-forはリストの長さ（要素数）分
繰り返し、順次アイテムを
一つずつ順番に取り出します。

- Foo
- Bar

結果



2.Vue基本構文

Chapter2 【リストレンダリング】

V-forの使い方②

```
<ul id="example-2">
  <li v-for="(item, index) in items" :key="index">
    {{ parentMessage }} - {{ index }} - {{ item.message }}
  </li>
</ul>
```

HTML

```
export default {
  name: "App",
  components: {},
  data() {
    return {
      parentMessage: "Parent",
      items: [{ message: "Foo" }, { message: "Bar" }],
    };
  },
};
```

JS

- Parent-0-Foo
- Parent-1-Bar

結果

2.Vue基本構文

Chapter2 【リストレンダリング】

V-forの使い方③

```
<ul id="example-3">
  <li v-for="value in object" :key="value">
    {{ value }}
  </li>
</ul>
```

HTML

```
export default {
  name: "App",
  components: {},
  data() {
    return {
      object: {
        title: "How to do lists in Vue",
        author: "Jane Doe",
        publishedAt: "2016-04-10",
      },
    };
  },
};
```

JS

- How to do lists in vue
- Jane Doe
- 2016-04-10

結果

2.Vue基本構文

Chapter2 【リストレンダリング】

V-forの使い方④

```
<div v-for="(value, name, index) in object" :key="name"> HTML  
  {{ index }}. {{ name }}: {{ value }}  
</div>
```

```
export default { JS  
  name: "App",  
  components: {},  
  data() {  
    return {  
      object: {  
        title: "How to do lists in Vue",  
        author: "Jane Doe",  
        publishedAt: "2016-04-10",  
      },  
    };  
  },  
};
```

```
0.title: How to do lists in Vue  
1.author: Jane Doe  
2.publishedAt: 2016-04-10
```

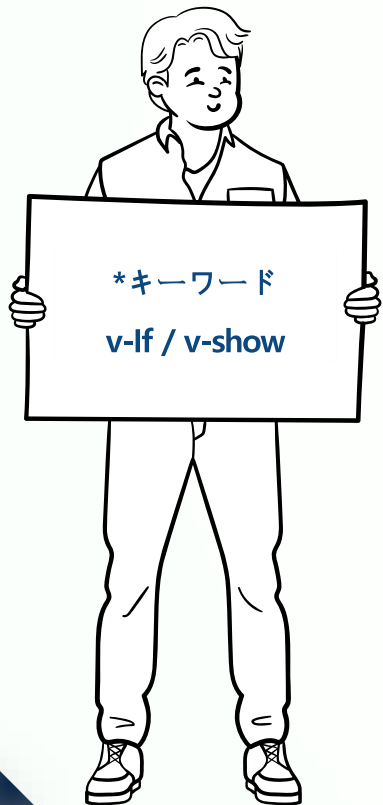
結果



- サンプルコード: <https://github.com/github-Ahn/PSSMiniPrj>
- プロジェクト名: sample_prj_chapter2

3. Vue基本構文

chapter3 【条件付きレンダリング】



2.Vue基本構文

Chapter3

【条件付きレンダリング】



2.Vue基本構文

Chapter3

【条件付きレンダリング】

```
<div>  
  <div v-if="type === 'A'">男の子です。 </div>  
  <div v-else-if="type === 'B'">女の子です。 </div>  
  <div v-else> 男でも女でもありません </div>  
</div>
```

HTML

```
export default {  
  name: "App",  
  components: {},  
  data() {  
    return {  
      type: "A",  
    };  
  },  
};
```

JS

結果

男の子です。

2.Vue基本構文

Chapter3

【条件付きレンダリング】

HTML

```
<h1 v-show="isEvening">こんばんは！</h1>
```

JS

```
export default {  
  name: "App",  
  components: {},  
  data() {  
    return {  
      isEvening : true,  
    };  
  },  
};
```

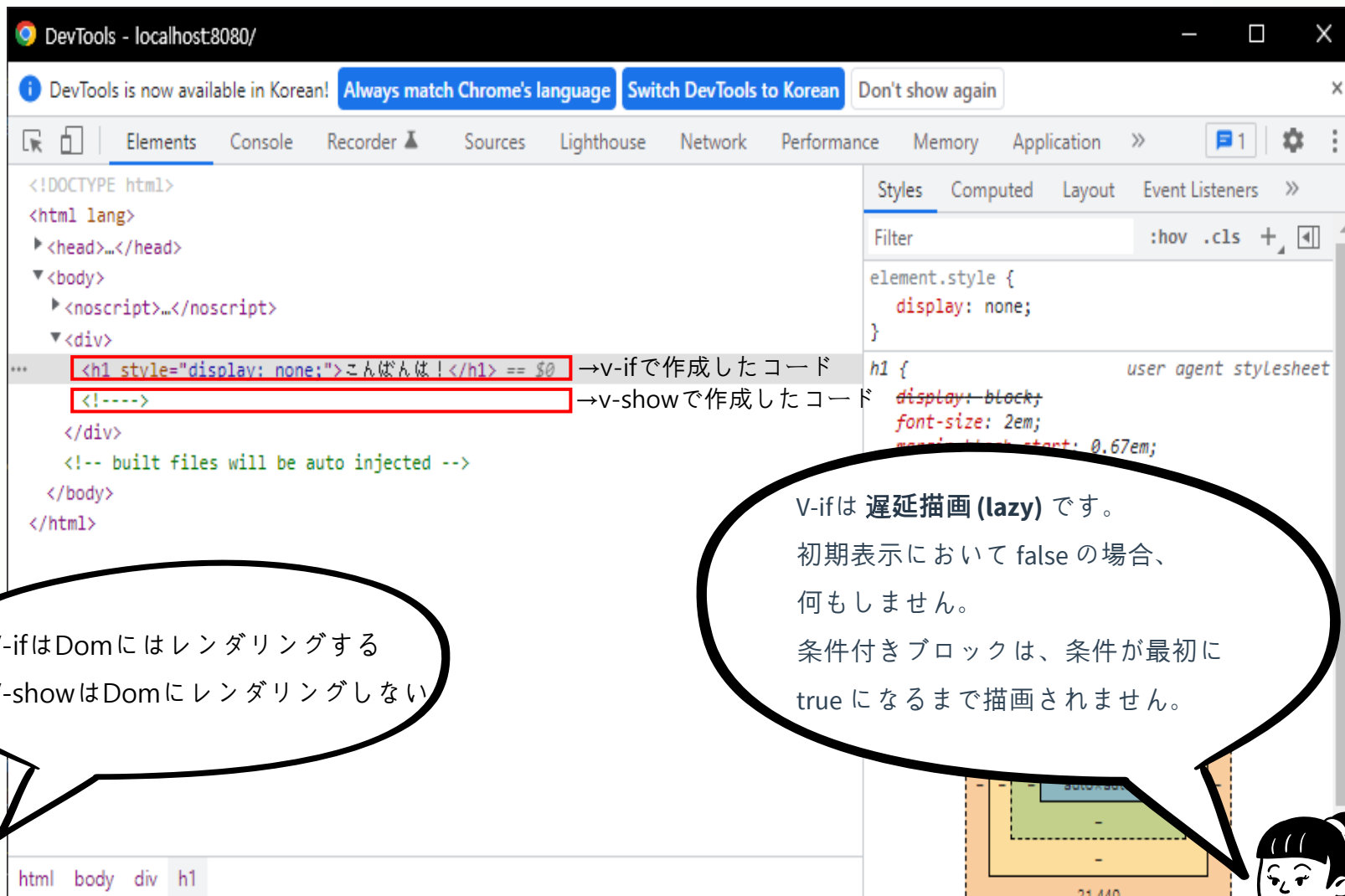
結果

こんばんは！

2.Vue基本構文

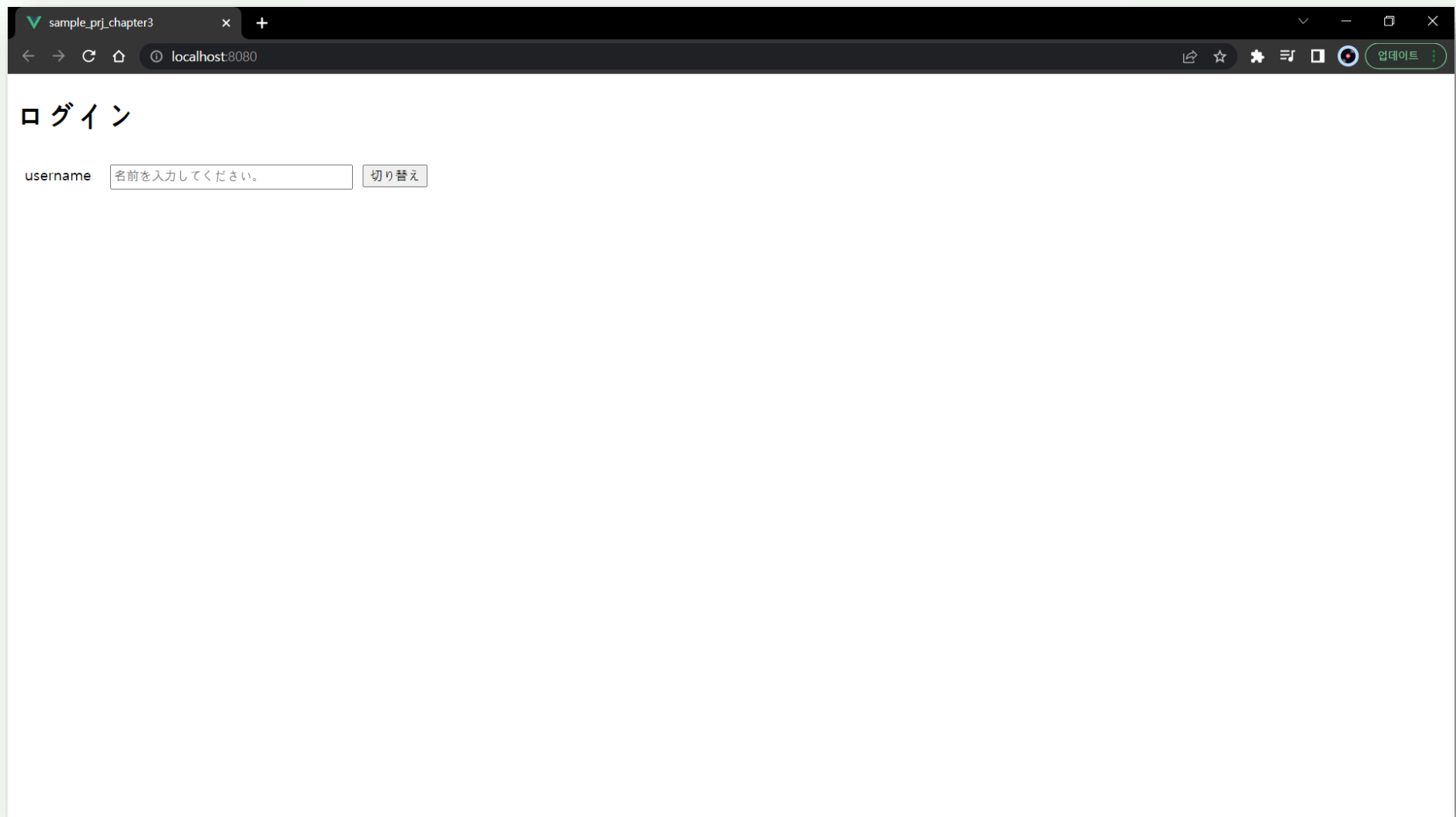
Chapter3

【条件付きレンダリング】



V-ifはDomにはレンダリングする
V-showはDomにレンダリングしない

V-ifは遅延描画 (lazy) です。
初期表示において false の場合、
何もありません。
条件付きブロックは、条件が最初に
true になるまで描画されません。

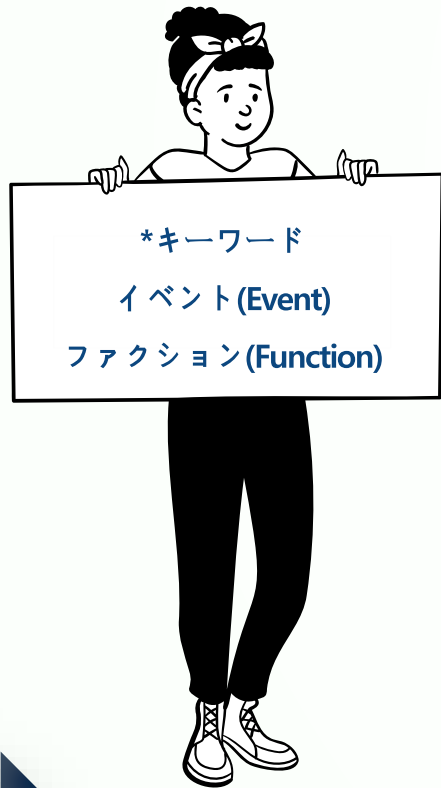


- サンプルコード: <https://github.com/github-Ahn/PSSMiniPrj>
- プロジェクト名: sample_prj_chapter3

3. Vue基本構文

chapter4

【イベントハンドリング】



2.Vue基本構文

Chapter4 【イベントハンドリング】

```
<div id="example-1">  
  <button v-on:click="counter += 1">Add 1</button>  
  <p>The button above has been clicked {{ counter }} times.</p>  
</div>
```

HTML

```
export default {  
  name: "App",  
  components: {},  
  data() {  
    return {  
      counter: 0,  
    };  
  },  
};
```

JS

Add 1

The button above has been clicked 0 times.

結果

2.Vue基本構文

Chapter4 【イベントハンドリング】

```
<div id="example-2">
  <!-- `greet` は、あらかじめ定義したメソッドの名前 -->
  <button v-on:click="greet">Greet</button>
</div>
```

HTML

```
export default {
  name: "App",
  components: {},
  data() {
    return {
      name: "Vue.js",
    };
  },
  // `methods` オブジェクトの下にメソッドを定義する
  methods: {
    greet: function (event) {
      // メソッド内の `this` は、Vue インスタンスを参照します
      alert("Hello " + this.name + "!");
      // `event` は、ネイティブ DOM イベントです
      if (event) {
        alert(event.target.tagName);
      }
    },
  },
};
```

JS

2.Vue基本構文

Chapter4

【イベントハンドリング】

```
<div id="example-3">
  <button v-on:click="say('hi')">Say hi</button>
  <button v-on:click="say('what')">Say what</button>
</div>
```

HTML

```
export default {
  name: "App",
  components: {},
  data() {
    return {};
  },
  // `methods` オブジェクトの下にメソッドを定義する
  methods: {
    say: function (message) {
      alert(message);
    },
  },
};
```

JS

2.Vue基本構文

Chapter4 【イベントハンドリング】

HTML

```
<!-- クリックイベントの伝搬が止まります -->
<a v-on:click.stop="doThis"></a>

<!-- submit イベントによってページがリロードされません -->
<form v-on:submit.prevent="onSubmit"></form>

<!-- 修飾子は繋げることができます -->
<a v-on:click.stop.prevent="doThat"></a>

<!-- 値を指定せず、修飾子だけ利用することもできます -->
<form v-on:submit.prevent></form>

<!-- イベントリスナーを追加するときにキャプチャモードで使います -->
<!-- 言い換えれば、内部要素を対象とするイベントは、その要素によって処理
される前にここで処理されます -->
<div v-on:click.capture="doThis">...</div>

<!-- event.target が要素自身のときだけ、ハンドラが呼び出されます -->
<!-- 言い換えると子要素のときは呼び出されません -->
<div v-on:click.self="doThat">...</div>
```

Tip

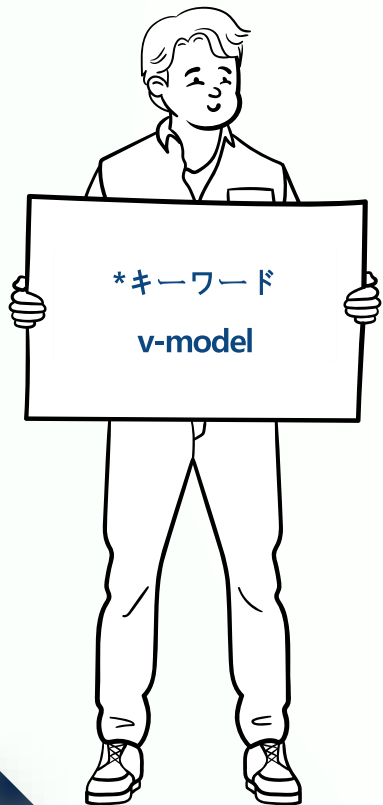
修飾子を使用するとき、関連するコードが同じ順序で生成されるため注意！

`v-on:click.prevent.self`を使用すると**全ての**クリックイベントを防ぐことはできますが、
`v-on:click.self.prevent`は要素自身におけるクリックイベントを防ぐだけです。

3. Vue基本構文

chapter5

【フォーム入力バインディング】



2.Vue基本構文

Chapter5

【フォーム入力バインディング】

amazon.co.jp

アカウントを作成

名前

フリガナ

携帯電話番号またはメールアドレス

パスワード

最低6文字必要です

i パスワードは最低6文字ありますか？
もう一度パスワードを入力してください

次に進む

Amazonの[利用規約](#)と[プライバシー規約](#)に同意いただける場合はアカウントを作成してください。

すでにアカウントをお持ちですか？ [サインイン](#)・
法人用の購買ですか？ [無料のビジネスアカウントを作成](#)

V-model:nameKanji

V-model:nameKana

V-model:address

フォーム入力したデータを簡単に扱う方法は？

V-modelを利用してフォームに簡単につなぎ合わせることができます。



The screenshot displays a web application with the following sections:

- input**: A single-line text input containing the text "テストテキストです!". Below it, a message reads "Message is: テストテキストです!".
- inputMultiline**: A multi-line text area containing the text "Multiline message is:" followed by a list of numbers: 1, 2, 3, 4.
- CheckBox**: A checkbox that is checked, with the label "true" next to it.
- CheckBoxMulti**: Three checkboxes labeled "Jack", "John", and "Mike". The "John" checkbox is checked. Below them, the text "Checked names: ['John']" is displayed.
- Radio**: Two radio buttons labeled "One" and "Two". The "Two" radio button is selected. Below them, the text "Picked: Two" is displayed.
- Select**: A dropdown menu with the letter "C" selected. Below it, the text "Selected: C" is displayed.

- サンプルコード: <https://github.com/github-Ahn/PSSMiniPrj>
- プロジェクト名: sample_prj_chapter5

4. ミニプロジェクト

テーマ：Vueで作る掲示板