



# Vue.js 프로그래밍

Song, Young Ohk

# 순서

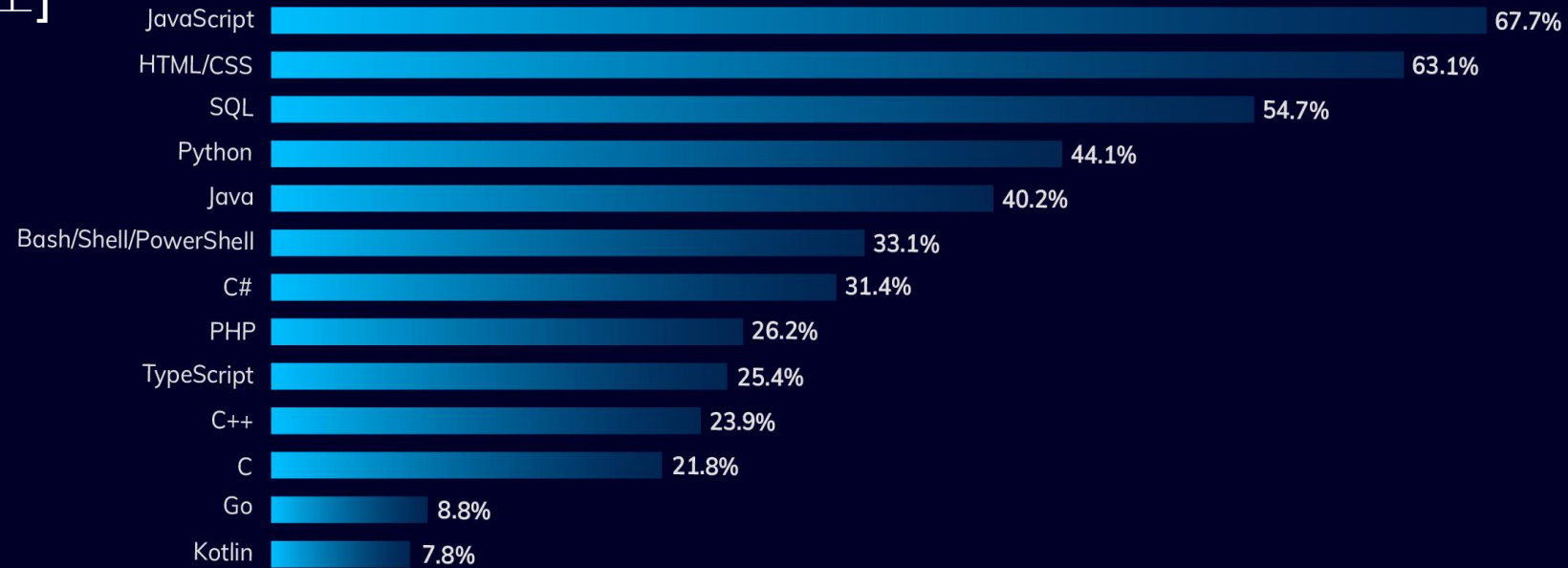
## ▶ 웹 개발 언어 이해

- ▶ 웹 개발을 위한 언어들
- ▶ HTML, CSS, JavaScript의 활용
- ▶ Front-End(Client Side) vs Back-End(Server Side) 개발
- ▶ SPA vs MPA 이해
- ▶ Front End 개발 익혀야 할 단계들

## ▶ Vue

- Vue 개발 환경
- Vue 프로젝트 생성
  - Vue.js 2와 Vue.js 3의 차이
- Vue Vite 프로젝트 생성
- 단일 페이지와 멀티 페이지 구조 이해

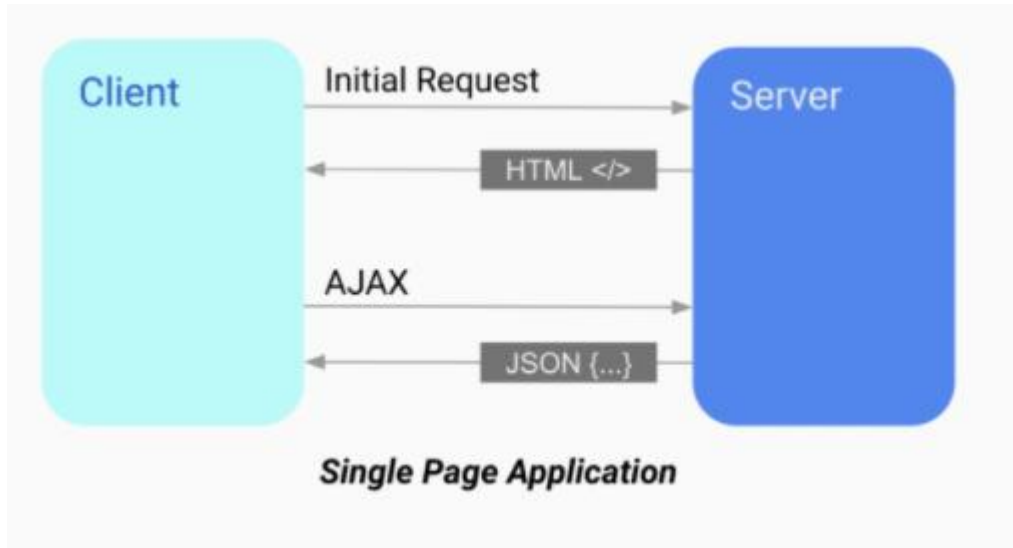
## [언어의 관심도]



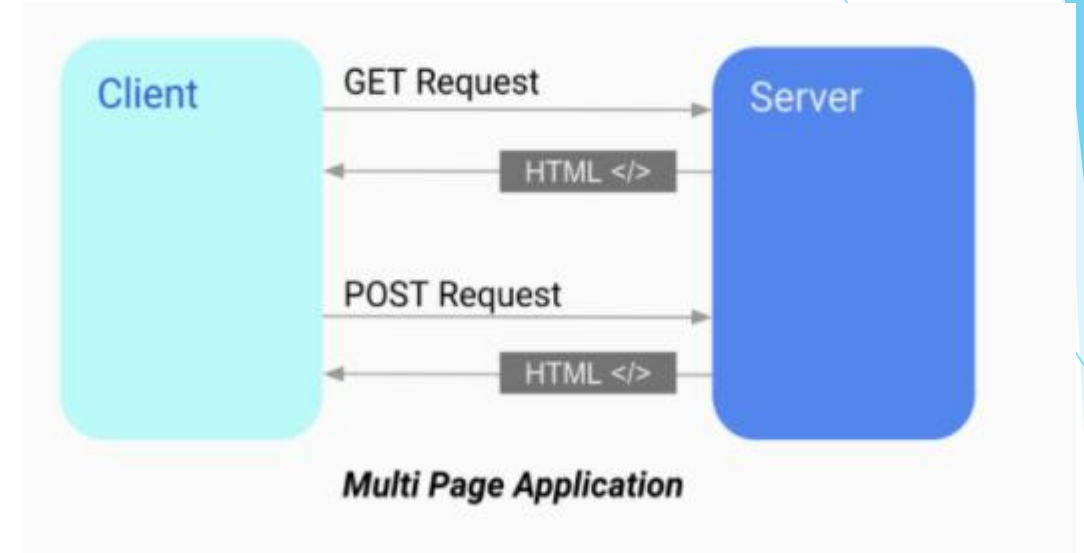
- 웹 개발용 가장 기본 언어
  - JavaScript + HTML + CSS을 잘 활용할 수 있는지
  - Server side에 필요한 언어
  - DataBase 사용
  - ...

# SPA vs MPA

## ▶ 어떻게 다를까!



1. 한 개(Single)의 Page로 구성된 Application이다.
2. CSR(Client Side Rendering) 방식으로 렌더링한다.
3. 단 한 번만 리소스(HTML, CSS, JavaScript)를 로딩하고 그 후에는 데이터를 받아올 때만 서버와 통신한다.
4. Vue, React, Angular



1. 완성된 형태의 HTML 파일을 서버로부터 전달받음 - 검색엔진이 페이지를 크롤링하기에 적합
2. 첫 로딩이 짧다.
  - 서버에서 이미 렌더링해 가져오기 때문이다.
  - 클라이언트가 JS 파일을 모두 다운로드하고 적용한 후에 각각의 기능이 동작 한다.

# Windows 기반 Vue 개발 환경

## (1) Vue 개발 환경 구축에 필요한 기본 SW

### ▶ Node.js와 NPM 설치(NPM 또는 yarn 사용)

▶ C:> start <https://nodejs.org>

▶ Yarn 설치(npm 대신 사용 가능한 도구)

▶ `npm install -g yarn`

### ▶ Visual Studio Code 설치

▶ C:> start <https://code.visualstudio.com>

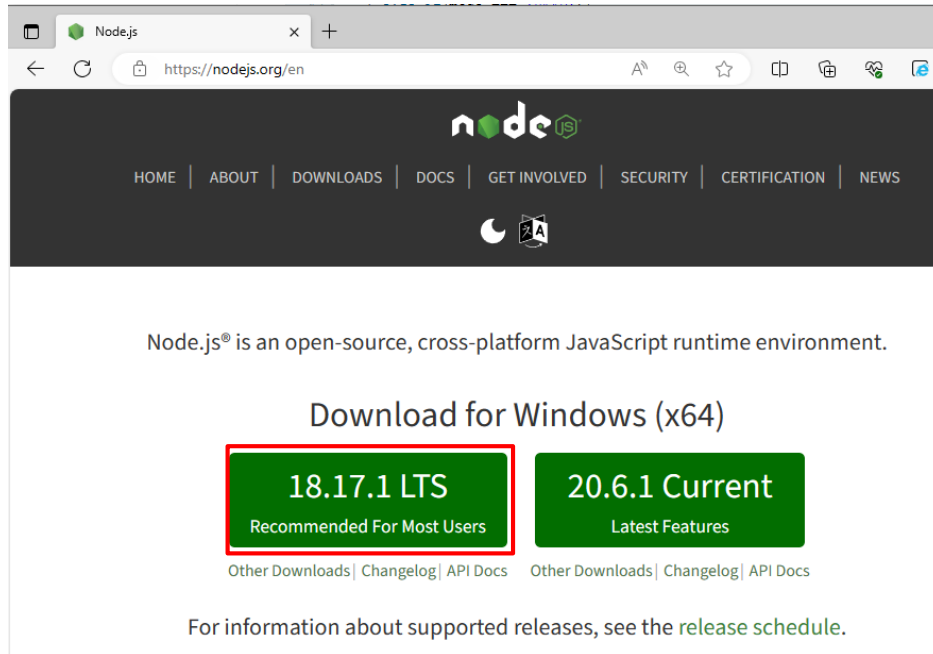
참고 사이트 : <https://ko.legacy.reactjs.org/>

> 노드 버전 확인 방법 : `node -v`

> npm 버전 확인 방법 : `npm -v`

# Windows 기반 Vue 개발 환경

## ▶ Node.js 및 Visual studio code 설치



Node.js® is an open-source, cross-platform JavaScript runtime environment.

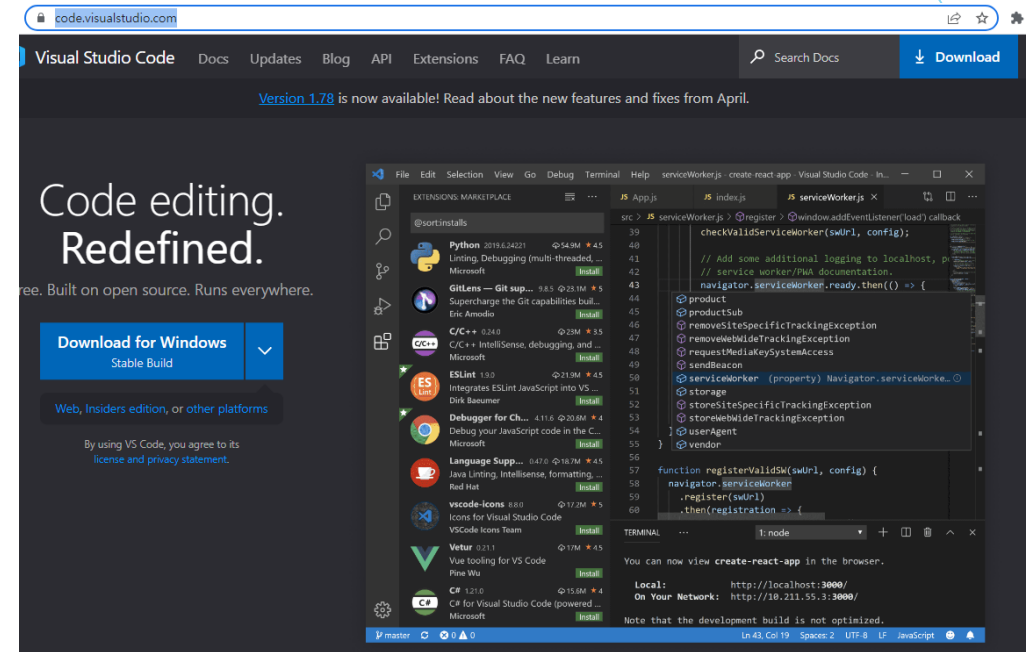
Download for Windows (x64)

**18.17.1 LTS**  
Recommended For Most Users

**20.6.1 Current**  
Latest Features

[Other Downloads](#) | [Changelog](#) | [API Docs](#)   [Other Downloads](#) | [Changelog](#) | [API Docs](#)

For information about supported releases, see the [release schedule](#).



또

Download Node.js®

**20.11.0 LTS**  
Recommended For Most Users

**21.6.1 Current**  
Latest Features

[Other Downloads](#) | [Changelog](#) | [API Docs](#)   [Other Downloads](#) | [Changelog](#) | [API Docs](#)

For information about supported releases, see the [release schedule](#).

# Vue CLI 이용 Vue2 Project 생성 및 실행

## 1 Vue CLI를 이용하여 Vue2 버전으로 프로젝트 생성

]# vue create vue-project-name

```
Vue CLI v5.0.8
? Please pick a preset:
  Default ([Vue 3] babel, eslint)
> Default ([Vue 2] babel, eslint)
  Manually select features
```

```
yarn install v1.22.21
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
[4/4] Building fresh packages...

success Saved lockfile.
Done in 3.97s.
🔗 Running completion hooks...

📄 Generating README.md...

🎉 Successfully created project vueproject2.
👉 Get started with the following commands:

$ cd vueproject2
$ yarn serve
```

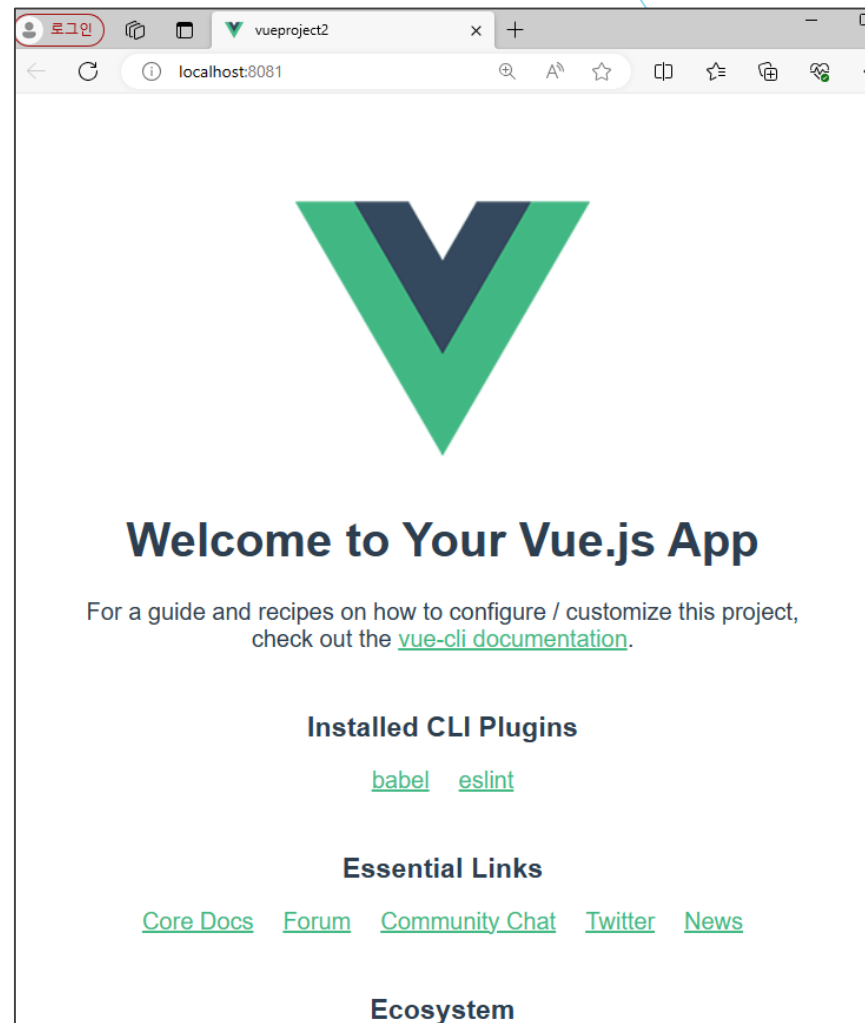
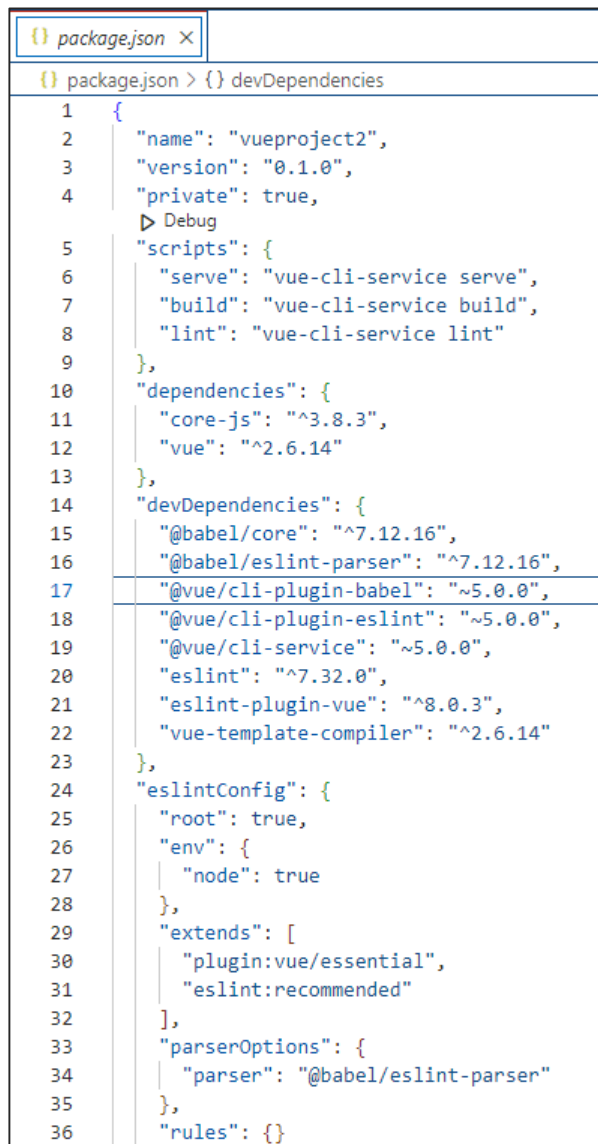
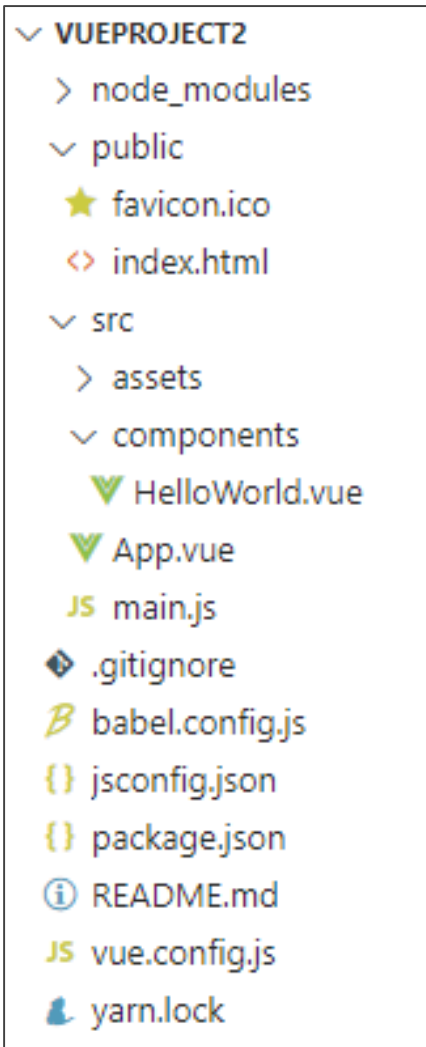
```
DONE Compiled successfully in 1975ms
```

App running at:

- Local: <http://localhost:8081/>
- Network: <http://192.168.219.108:8081/>

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

# Vue2 실행 및 프로젝트 구조





# Vue2 파일 구조 이해

The diagram illustrates the file structure and relationships in a Vue2 project. It shows three files: `index.html`, `main.js`, and `App.vue`.

**index.html** (public > <> index.html > ...):

```
1 <!DOCTYPE html>
2 <html lang="">
3   <head>
4     <meta charset="utf-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width,initial-scale=1.0">
7     <link rel="icon" href="<%= BASE_URL %>favicon.ico">
8     <title><%= htmlWebpackPlugin.options.title %></title>
9   </head>
10  <body>
11    <noscript>
12      <strong>We're sorry but <%= htmlWebpackPlugin.options.title %> doesn't w
13    </noscript>
14    <div id="app"></div>
15    <!-- built files will be auto injected -->
16  </body>
17 </html>
```

**main.js** (src > JS main.js):

```
1 import { createApp } from 'vue'
2 import App from './App.vue'
3
4 createApp(App).mount('#app')
5
6
7 render: h => h(App),
8 }).$mount('#app')
```

**App.vue** (src > App.vue):

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

Red circles and arrows indicate the relationships between the files:

- A red circle around the `id="app"` attribute in `index.html` and a red circle around the `mount('#app')` call in `main.js` are connected by a red arrow, indicating that the `main.js` file mounts the application to the `#app` element in `index.html`.
- A red circle around the `App` argument in `createApp(App)` in `main.js` and a red circle around the `App.vue` file name in the editor tab are connected by a red arrow, indicating that `main.js` imports and uses the `App` component defined in `App.vue`.
- A red circle around the `import HelloWorld` statement in `App.vue` and a red circle around the `components: { HelloWorld }` block in the same file are connected by a red arrow, indicating that the `HelloWorld` component is imported and registered in the `App` component.

# Vue CLI 이용 Vue3 Project 생성 및 실행

## 2. Vue CLI를 이용하여 Vue3 버전으로 프로젝트 생성

]# vue create vue-project-name

```
Vue CLI v5.0.8
? Please pick a preset: (Use arrow keys)
> Default ([Vue 3] babel, eslint)
  Default ([Vue 2] babel, eslint)
  Manually select features
```

```
yarn install v1.22.21
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
[4/4] Building fresh packages...

success Saved lockfile.
Done in 6.91s.
🔗 Running completion hooks...

📄 Generating README.md...

🎉 Successfully created project vueproject
👉 Get started with the following commands:

$ cd vueproject
$ yarn serve
```

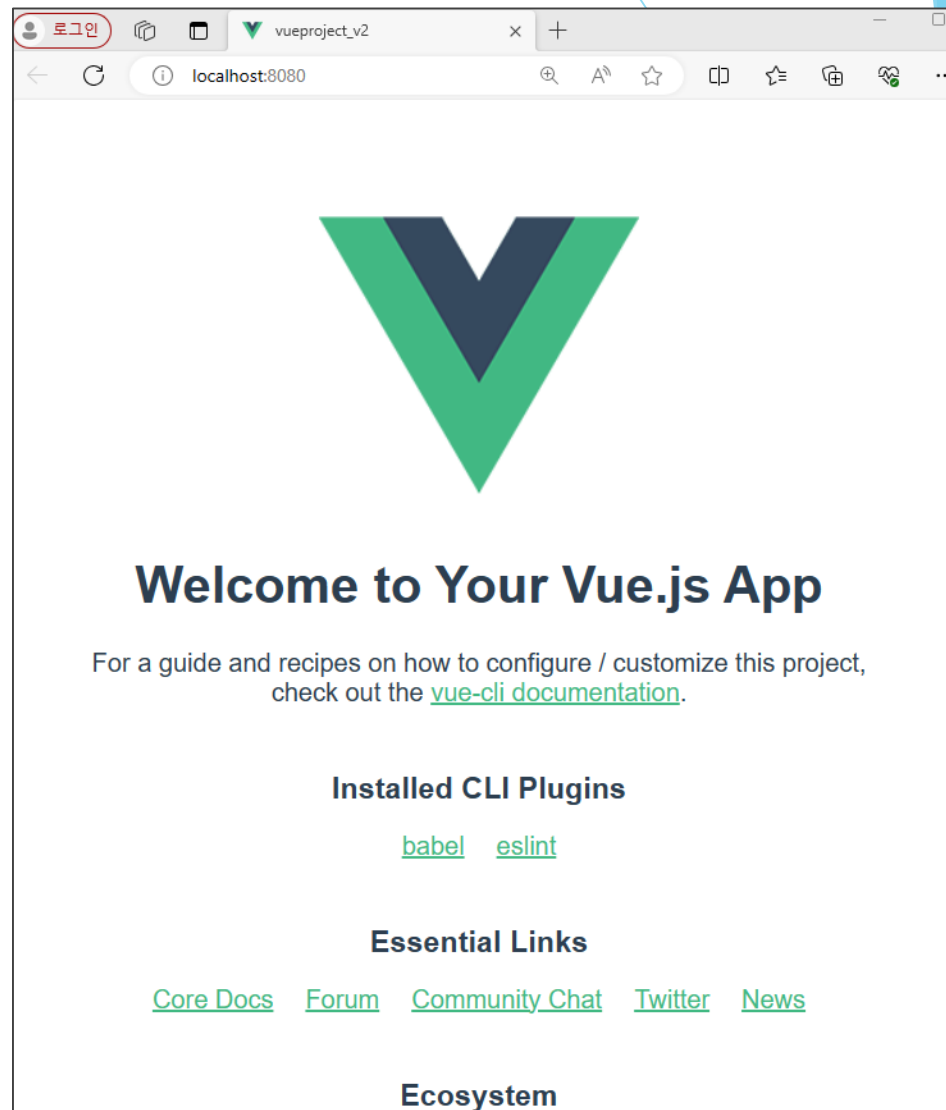
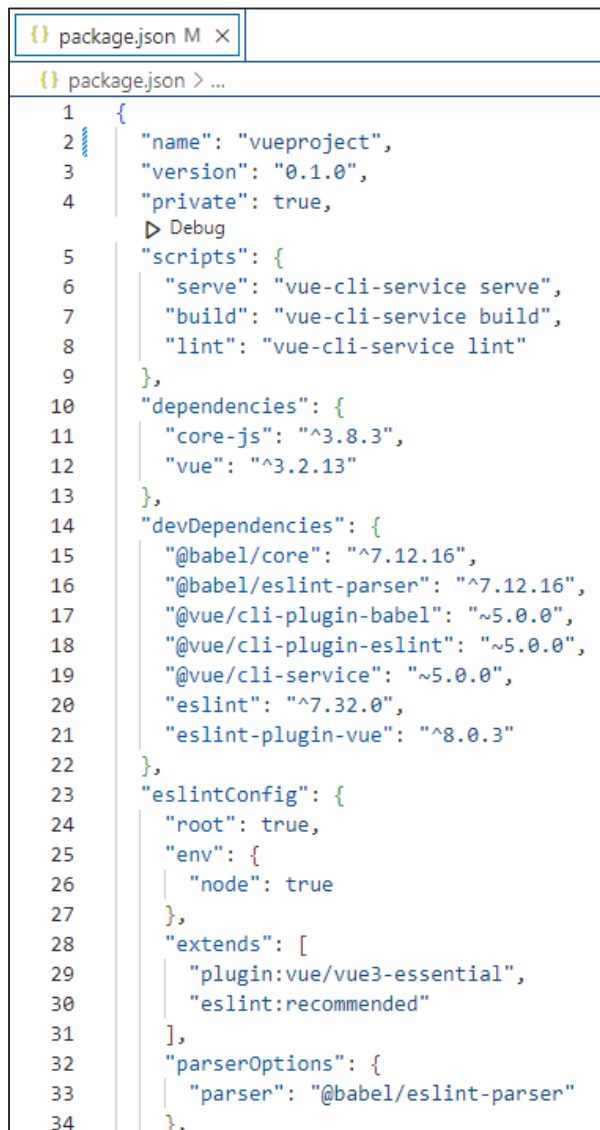
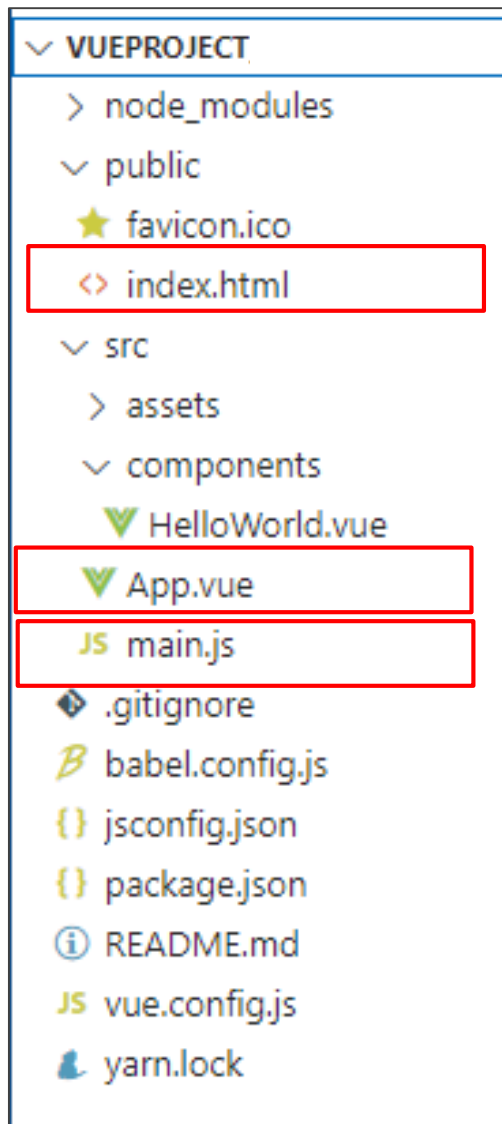
```
DONE Compiled successfully in 1908ms
```

App running at:

- Local: <http://localhost:8080/>
- Network: <http://192.168.219.108:8080/>

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

# Vue3 실행 및 프로젝트 구조



# Vue3 파일 구조 이해

The diagram illustrates the file structure and component relationships in a Vue3 project. It shows three files: `index.html`, `main.js`, and `App.vue`.

**index.html** (public > <> index.html > ...):

```
1 <!DOCTYPE html>
2 <html lang="">
3   <head>
4     <meta charset="utf-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width,initial-scale=1.0">
7     <link rel="icon" href="<%= BASE_URL %>favicon.ico">
8     <title><%= htmlWebpackPlugin.options.title %></title>
9   </head>
10  <body>
11    <noscript>
12      <strong>We're sorry but <%= htmlWebpackPlugin.options.title %> doesn't w
13    </noscript>
14    <div id="app"></div>
15    <!-- built files will be auto injected -->
16  </body>
17 </html>
```

**main.js** (src > JS main.js):

```
1 import { createApp } from 'vue'
2 import App from './App.vue'
3
4 createApp(App).mount('#app')
```

**App.vue** (src > App.vue):

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

Red circles highlight the `id="app"` attribute in `index.html`, the `App` component in `main.js`, and the `App.vue` file in the file explorer. Red arrows indicate the relationships: from `id="app"` to `App` in `main.js`, and from `App` in `main.js` to `App.vue`.

# Vue Vite 이용 Project 생성 및 구조

## 3 Vue Vite를 이용하여 프로젝트 생성

]# vue create vite@latest

```
E:\vueproject>npm create vite@latest
Need to install the following packages:
create-vite@5.2.1
Ok to proceed? (y) y
✓ Project name: ... vueviteproject
? Select a framework: » - Use arrow-keys. Return to submit.
  Vanilla
>  Vue
  React
  Preact
  Lit
  Svelte
  Solid
  Qwik
  Others
```

```
E:\vueproject>npm create vite@latest
Need to install the following packages:
create-vite@5.2.1
Ok to proceed? (y) y
✓ Project name: ... vueviteproject
✓ Select a framework: » Vue
? Select a variant: » - Use arrow-keys. Return to submit.
  TypeScript
>  JavaScript
  Customize with create-vue ↗
  Nuxt ↗
```

```
E:\vueproject>npm create vite@latest
Need to install the following packages:
create-vite@5.2.1
Ok to proceed? (y) y
✓ Project name: ... vueviteproject
✓ Select a framework: » Vue
✓ Select a variant: » JavaScript

Scaffolding project in E:\vueproject\vueviteproject...

Done. Now run:

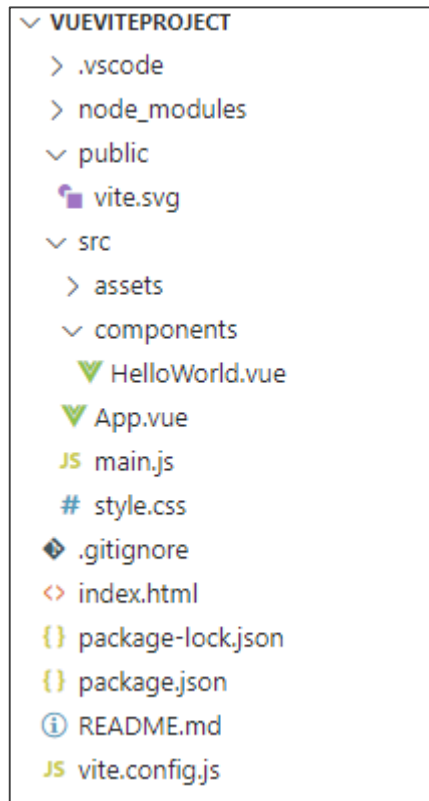
  cd vueviteproject
  npm install
  npm run dev

E:\vueproject>cd vueviteproject

E:\vueproject\vueviteproject>npm install
[ ] | reify:@rollup/rollup-win32-x64-msvc: http fe
```

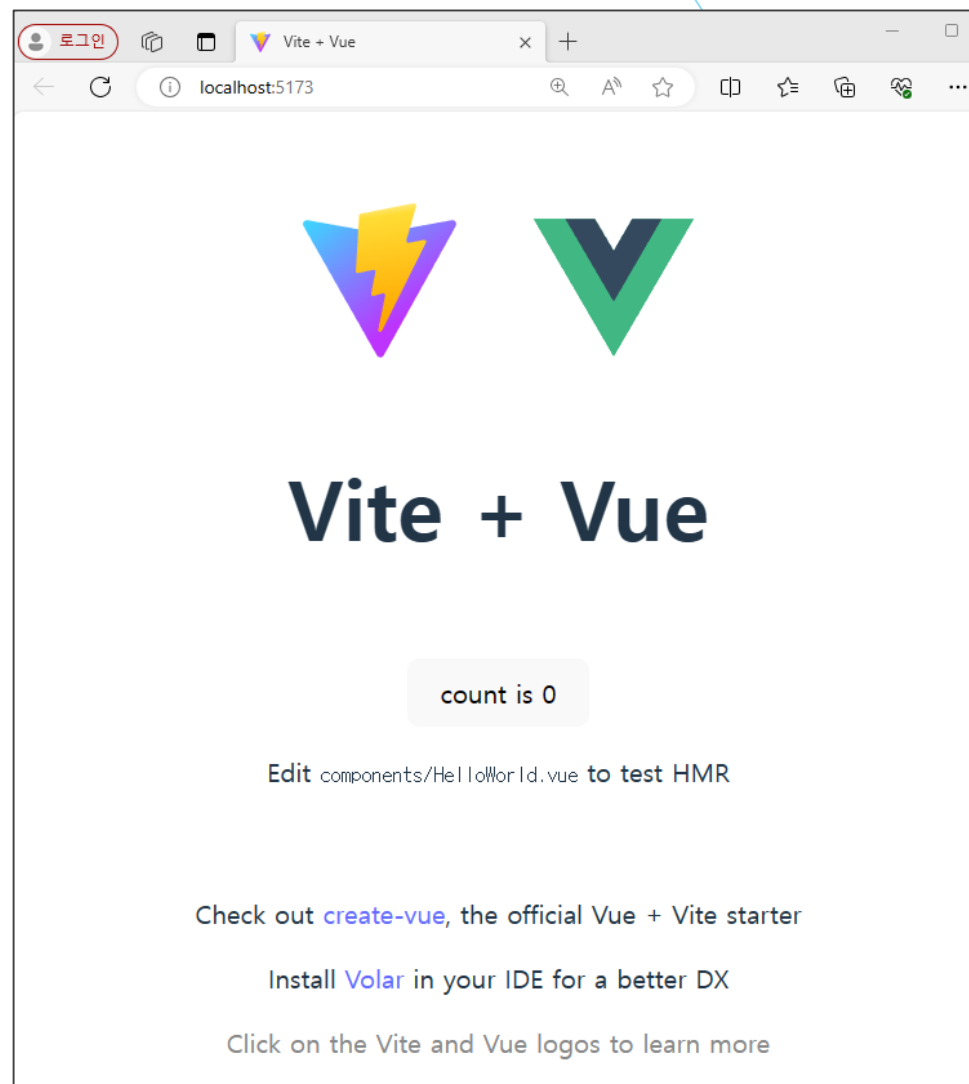
# Vue Vite 실행 및 프로젝트 구조

]# npm run dev



```
VITE v5.1.5 ready in 202 ms
→ Local:   http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help
```

```
{ } package.json X
{ } package.json > ...
1  {
2    "name": "vueviteproject",
3    "private": true,
4    "version": "0.0.0",
5    "type": "module",
6    "scripts": {
7      "dev": "vite",
8      "build": "vite build",
9      "preview": "vite preview"
10   },
11   "dependencies": {
12     "vue": "^3.4.19"
13   },
14   "devDependencies": {
15     "@vitejs/plugin-vue": "^5.0.4",
16     "vite": "^5.1.4"
17   }
18 }
```



# Vue Vite 파일 구조 이해

```
<> index.html X
<> index.html > ...
1 <!doctype html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8" />
5   <link rel="icon" type="image/svg+xml" href="/vite.svg" />
6   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7   <title>Vite + Vue</title>
8 </head>
9 <body>
10   <div id="app"></div>
11   <script type="module" src="/src/main.js"></script>
12 </body>
13 </html>
```

```
JS main.js X
src > JS main.js
1 import { createApp } from 'vue'
2 import './style.css'
3 import App from './App.vue'
4
5 createApp(App).mount('#app')
```

```
App.vue X
src > App.vue
1 <script setup>
2 import HelloWorld from './components/HelloWorld.vue'
3 </script>
4
5 <template>
6   <div>
7     <a href="https://vitejs.dev" target="_blank">
8       
9     </a>
10    <a href="https://vuejs.org/" target="_blank">
11      
12    </a>
13  </div>
14  <HelloWorld msg="Vite + Vue" />
15 </template>
16
17 <style scoped>
18 .logo {
19   height: 6em;
20   padding: 1.5em;
21   will-change: filter;
22   transition: filter 300ms;
23 }
24 .logo:hover {
25   filter: drop-shadow(0 0 2em #646cffaa);
26 }
27 .logo.vue:hover {
28   filter: drop-shadow(0 0 2em #42b883aa);
29 }
30 </style>
```

# Vue-cli 와 Vue Vite Node 버전

---

## ▶ Vue-cli 특징

- ▶ Vue-cli는 Node 6이상에서 이용
- ▶ Vue-cli는 Vue2, Vue3 모두 생성 가능

## ▶ Vue Vite 특징

- ▶ Vue vite는 Node 12 이상 필수
- ▶ Vite는 Vue3 만 생성



# Vue-CLI와 Vite의 비교

## ▶ Vue-CLI의 장단점

### ▶ 장점

- ▶ 다양한 기능을 제공하며 **Vue.js** 애플리케이션의 빌드, 테스트 및 배포를 위한 다양한 구성을 제공한다.
- ▶ 프로젝트의 개발을 위한 설정을 쉽게 커스텀 할 수 있는 설정 파일을 제공한다.
- ▶ 대규모 운영 애플리케이션에 적합하다.
- ▶ 공식 **Vue.js** 팀에서 유지 보수하므로 업데이트와 일관된 개발 경험을 보장한다.
- ▶ **Vue2**, **Vue3** 모두 지원한다.

### ▶ 단점

- ▶ **Vue-CLI**는 간단한 프로젝트를 구성할 때 설정이 복잡하다 느낄 수 있다.
- ▶ 프로젝트의 규모가 커질수록 빌드나 서버 실행의 속도가 **Vite**보다 느려진다.

# Vue-CLI와 Vite의 비교

## ▶ Vite의 장단점

### ▶ 장점

- ▶ ESM(ES Module)을 기반으로 한 빠른 개발 서버이다.
- ▶ 설정이 간소화 되어있어 직관적인 구성 설정을 제공한다.
- ▶ Vue3에 최적화되어 있으며, Vue2가 사용 종료될 시점이 다가오므로 상당한 이점으로 작용하다.
- ▶ 속도와 간결성 때문에 중소 규모 프로젝트를 빌드하는 데 탁월하다.
- ▶ Vue 뿐만 아니라 Vanilla, React, Preact, Lit, Svelte, Solid, Qwik 등의 다른 프레임워크와도 사용할 수 있다.

### ▶ 단점

- ▶ Vue-CLI만큼 많은 최적화 및 프로덕션 기능을 제공하지 않을 수 있다.
- ▶ Vue-CLI에 비해 플러그인이나 커뮤니티 등이 부족할 수 있다.
- ▶ 좀 더 전문화하기 위해서는 Webpack을 직접 이해하고 구성해야 할 수 있어 Vite의 모든 기능을 끌어내지 못할 수 있다.