

# Configure for Lab

**Bok, Jong Soon**  
**javaexpert@nate.com**  
**<https://github.com/swacademy/Vue.js>**

# Contents

- Web Browser
  - Google Chrome
  - Mozilla Firefox
- Editor or IDE
  - VSCode
- Node.js
- Git
- Vue.js devtools
- Vue-CLI

# Google Chrome Browser

- Google Chrome Browser
  - <https://www.google.com/chrome/>
- Mozilla Firefox
  - <https://www.mozilla.org/ko/>



# Editor or IDE

- VSCode

- <https://code.visualstudio.com/>

- Atom

- <https://atom.io/>

- SublimeText3

- <https://www.sublimetext.com/3>






# VSCode

- <https://code.visualstudio.com/Download>

## Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.



↓ Windows

Windows 7, 8, 10

User Installer64 bit32 bit

System Installer64 bit32 bit

.zip64 bit32 bit

↓ .deb

Debian, Ubuntu

.deb64 bit

.rpm64 bit

.tar.gz64 bit

↓ .rpm

Red Hat, Fedora, SUSE

.deb64 bit

.rpm64 bit

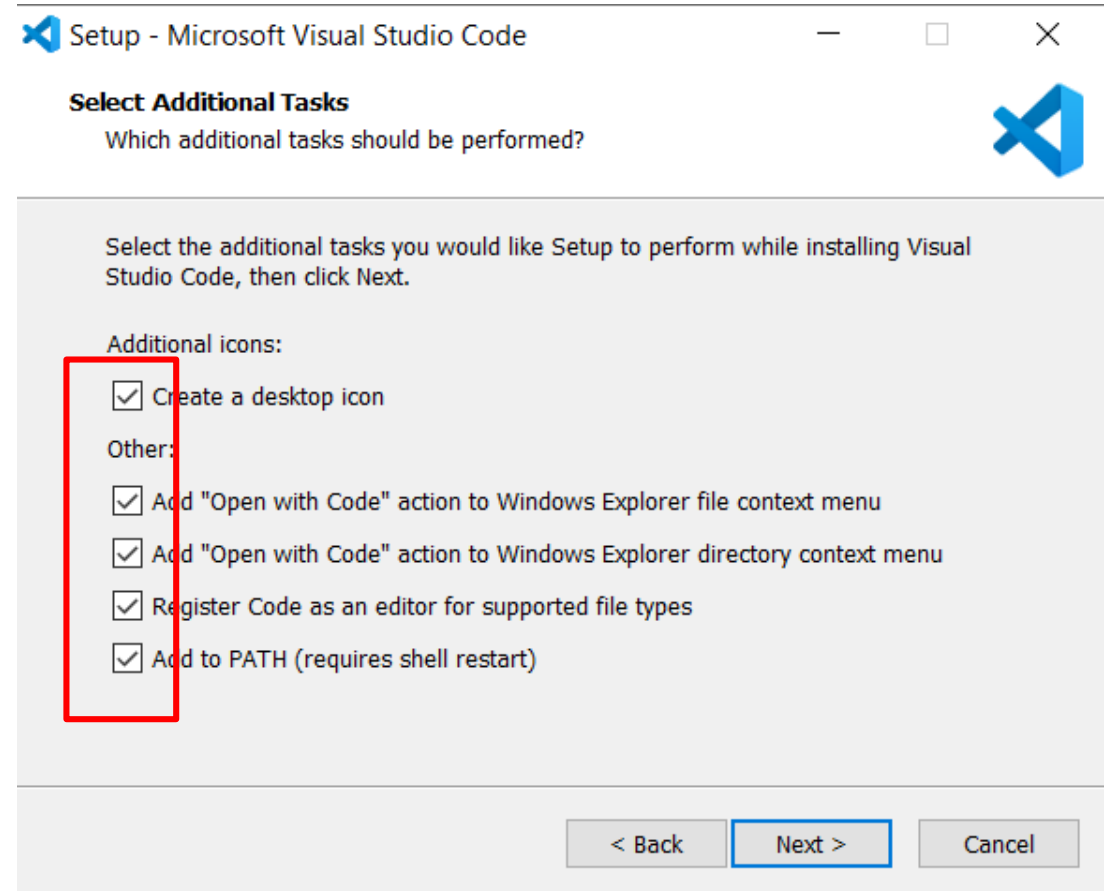
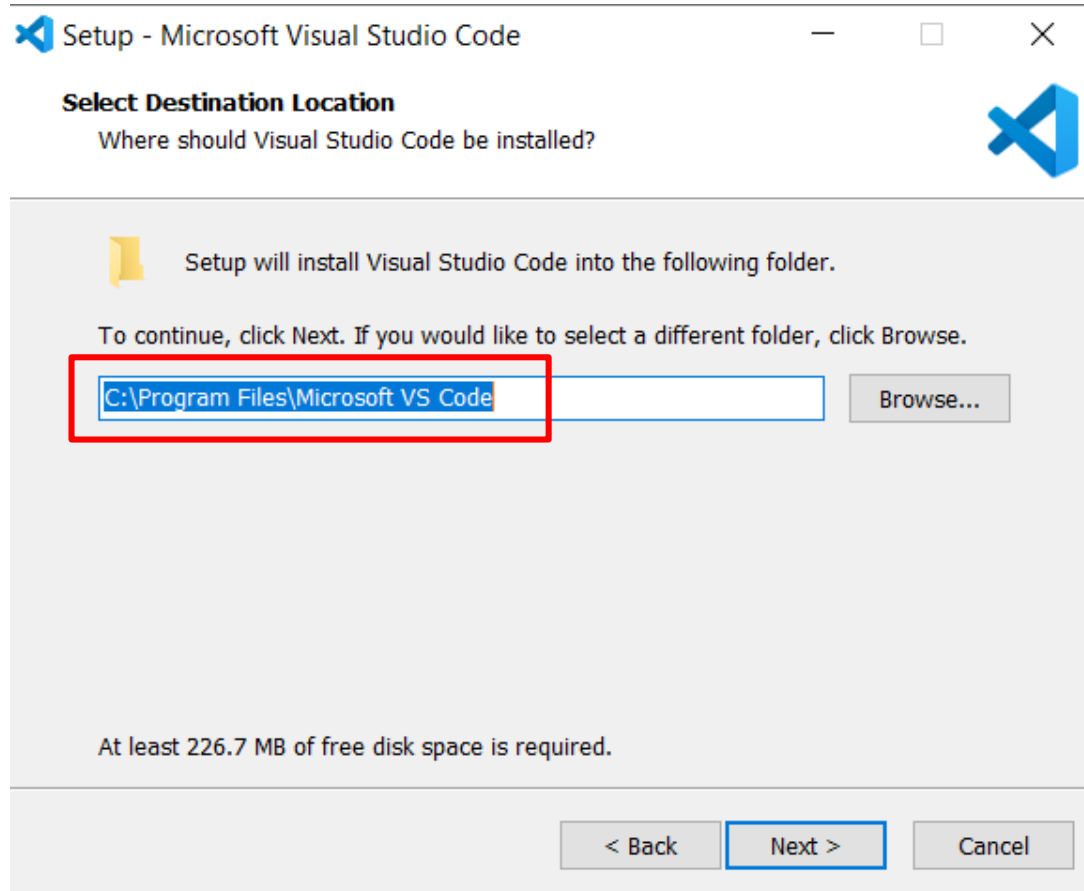
.tar.gz64 bit

↓ Mac

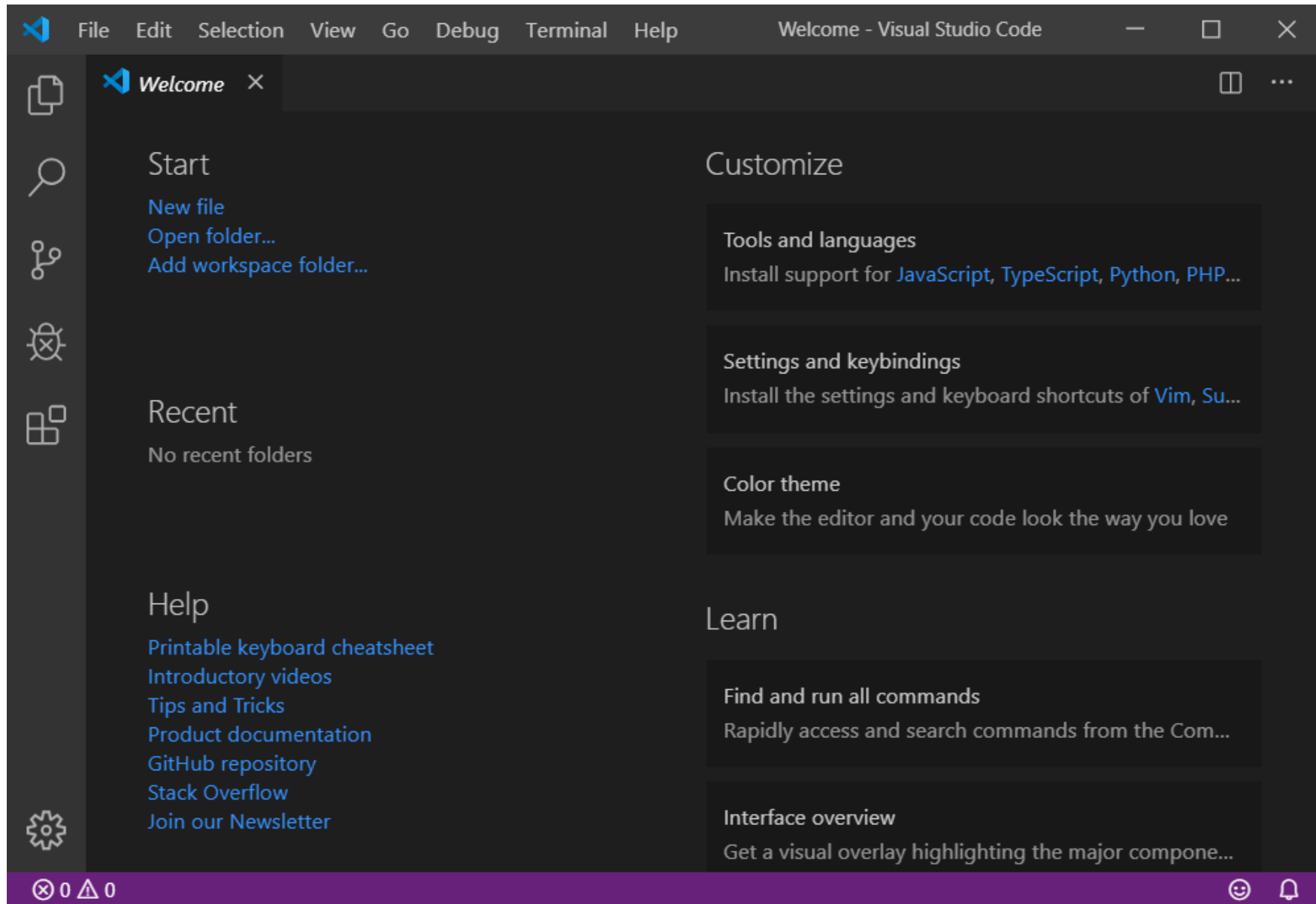
macOS 10.10+

Snap Store

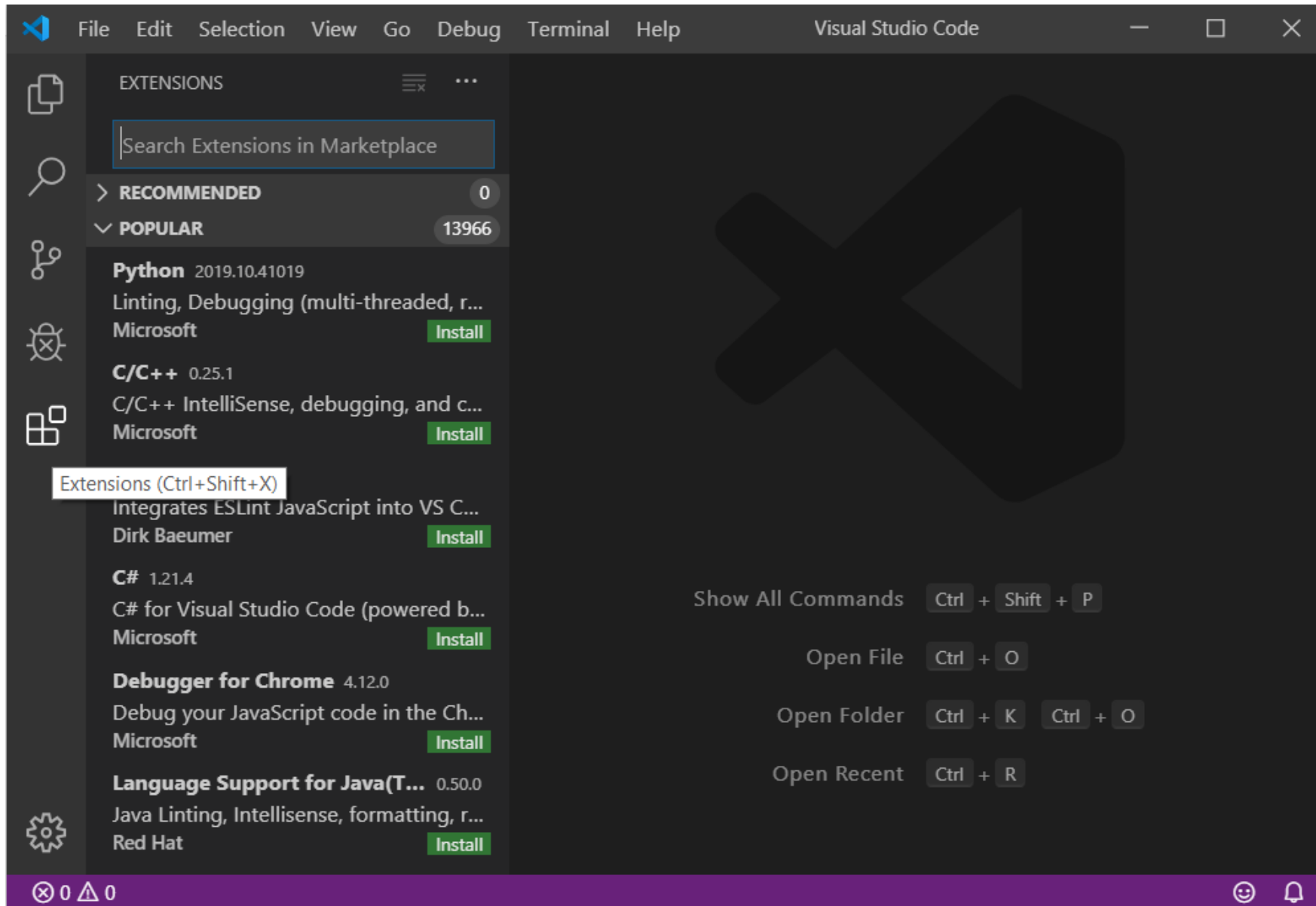
# VSCode (Cont.)



# VSCode (Cont.)



# VSCode Extensions





# VSCode Extensions (Cont.)

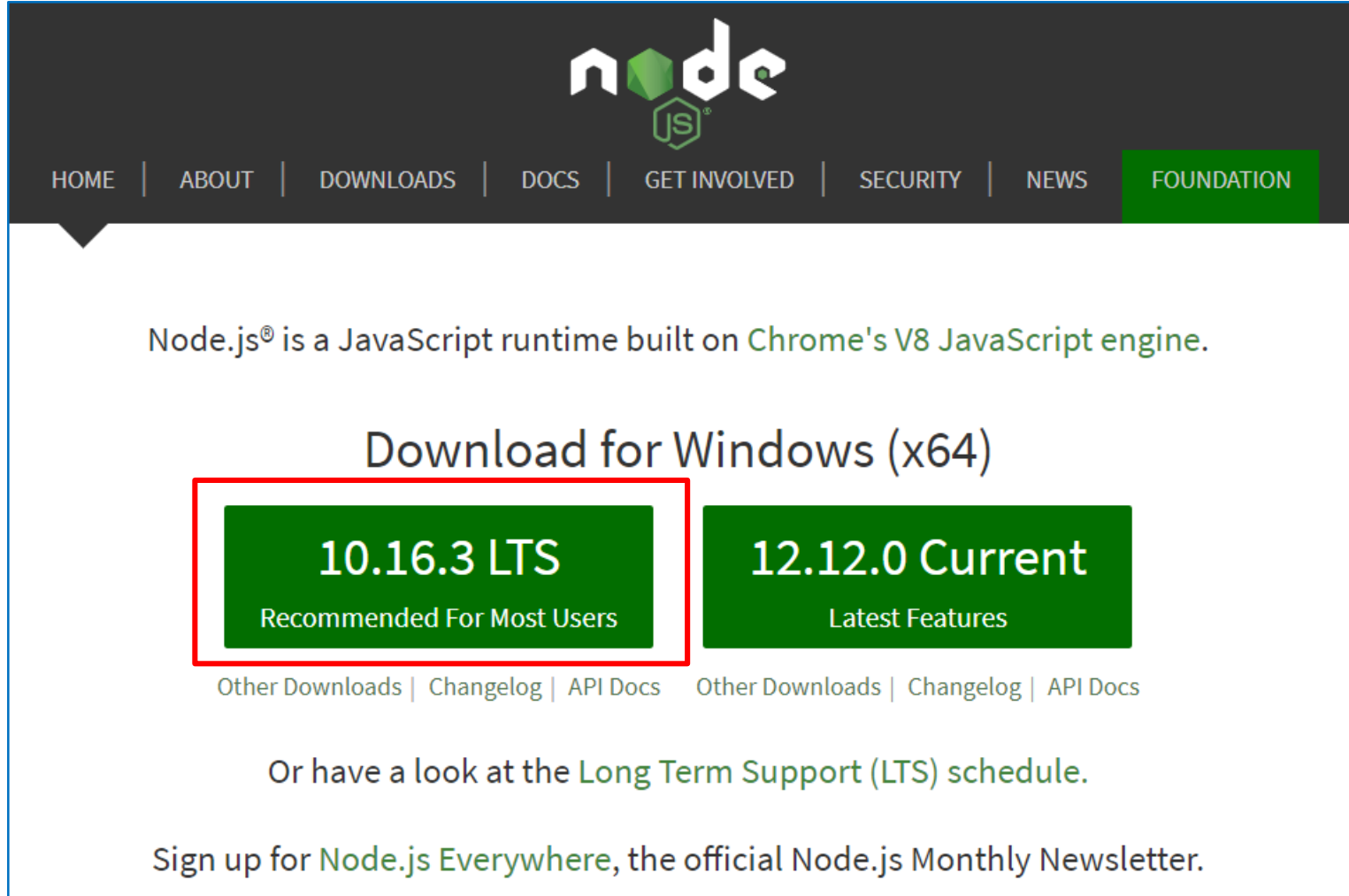
- Open in Browser
  - To open the current file in default browser.
- Htmltagwrap
  - Wraps selected code with HTML Tags
- Vetur
  - Vue.js code에 대한 문법 강조, 코드 자동완성, Debugging, Linting
- Vue Peek
  - Code 상에 작성되어 있는 Vue Component를 통해 해당 Component 파일을 열고 그 파일로 이동할 수 있는 기능
- Vue 2 Snippets
  - Vue Component 작성시 수기로 작성해야할 부분을 VSCode의 Plugin에서 자동완성

# VSCode Extensions (Cont.)

- Night Owl
  - VSCode color theme.
- Material Icon Theme
  - VSCode icon theme
- Live Server
  - Launch a development local Server with live reload feature for static & dynamic pages.
- ESLint
  - JavaScript code style, 문법 check 기능 제공
- Prettier
  - Code formatter
- Auto Close Tag
- Atom Keymap

# Node.js

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



The screenshot shows the Node.js website homepage. At the top is a dark navigation bar with the Node.js logo and links for HOME, ABOUT, DOWNLOADS, DOCS, GET INVOLVED, SECURITY, NEWS, and FOUNDATION. The FOUNDATION link is highlighted in green. Below the navigation bar, a text block states: "Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine." This is followed by a heading "Download for Windows (x64)". Under this heading are two green buttons: "10.16.3 LTS" with the subtext "Recommended For Most Users" (this button is highlighted with a red border) and "12.12.0 Current" with the subtext "Latest Features". Below these buttons are two identical links: "Other Downloads | Changelog | API Docs". Further down is a text block: "Or have a look at the Long Term Support (LTS) schedule." At the bottom is a text block: "Sign up for Node.js Everywhere, the official Node.js Monthly Newsletter."

node

HOME | ABOUT | DOWNLOADS | DOCS | GET INVOLVED | SECURITY | NEWS | FOUNDATION

Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

Download for Windows (x64)

**10.16.3 LTS**  
Recommended For Most Users

**12.12.0 Current**  
Latest Features

Other Downloads | Changelog | API Docs    Other Downloads | Changelog | API Docs

Or have a look at the Long Term Support (LTS) schedule.

Sign up for Node.js Everywhere, the official Node.js Monthly Newsletter.

## Node.js (Cont.)


```
C:\Users\bluee>echo %PATH%  
C:\Program Files\Java\jdk1.8.0_221\bin;C:\Windows\system32;C:\Windows;C:  
\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Wi  
ndows\System32\OpenSSH\;C:\Program Files\apache-maven-3.6.2\bin;C:\Progr  
am Files\apache-tomcat-9.0.26\bin;;C:\Program Files\Microsoft VS Code\bi  
n;C:\Program Files\nodejs\;C:\Users\bluee\AppData\Local\Microsoft\Window  
sApps;C:\Program Files\Bandizip\;C:\Users\bluee\AppData\Roaming\npm
```

```
C:\Users\bluee>node --version  
v10.16.3
```

```
C:\Users\bluee>npm --version  
6.9.0
```


# Git


■ <https://git-scm.com/>

 **git** --fast-version-control

Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.


Git is **easy to learn** and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient **staging areas**, and **multiple workflows**.






### About

The advantages of Git compared to other source control systems.




### Documentation

Command reference pages, Pro Git book content, videos and other material.




### Downloads

GUI clients and binary releases for all major platforms.



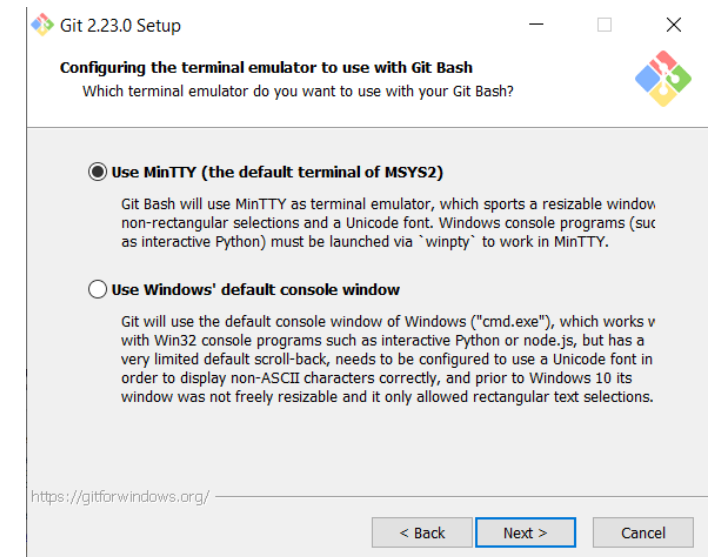
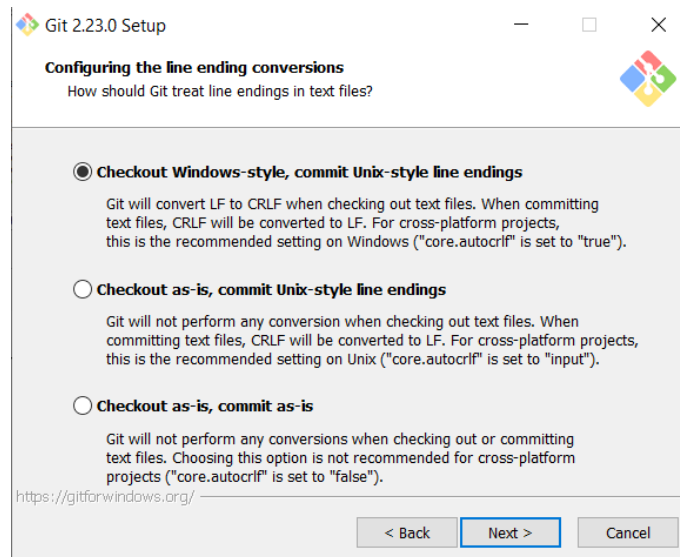
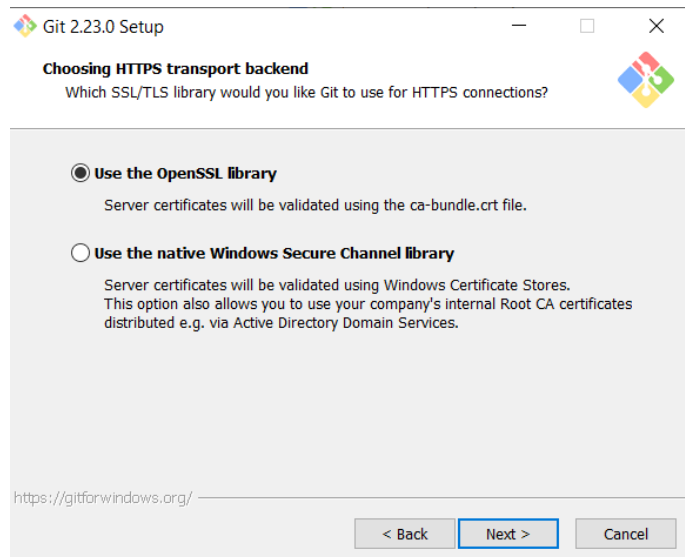
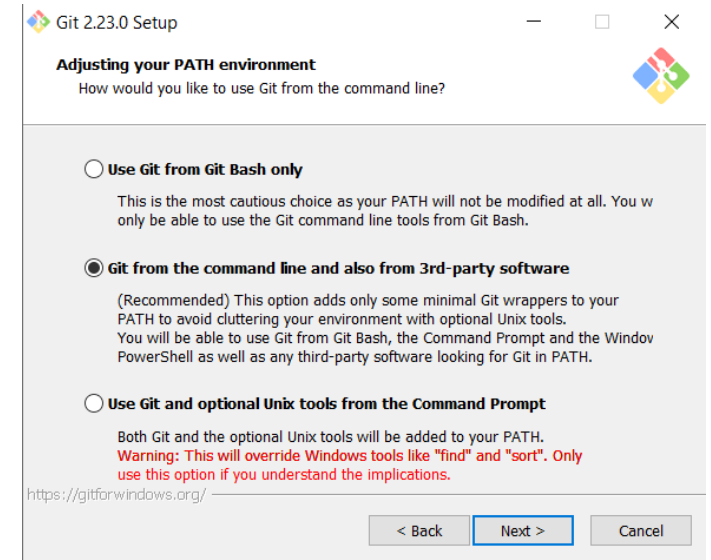
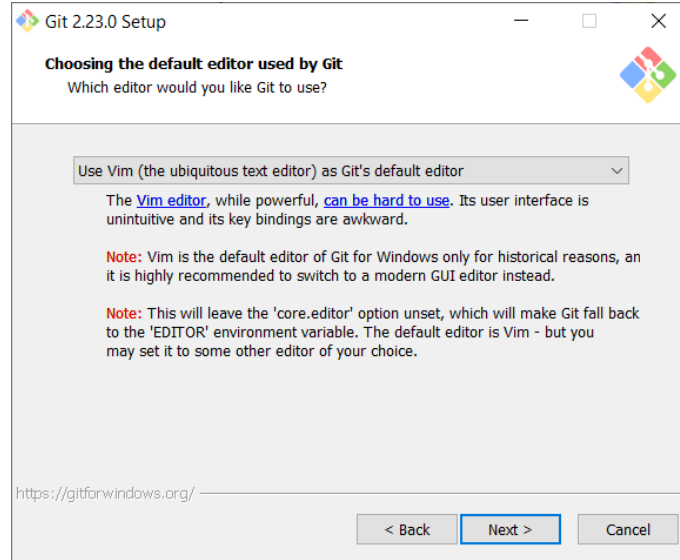
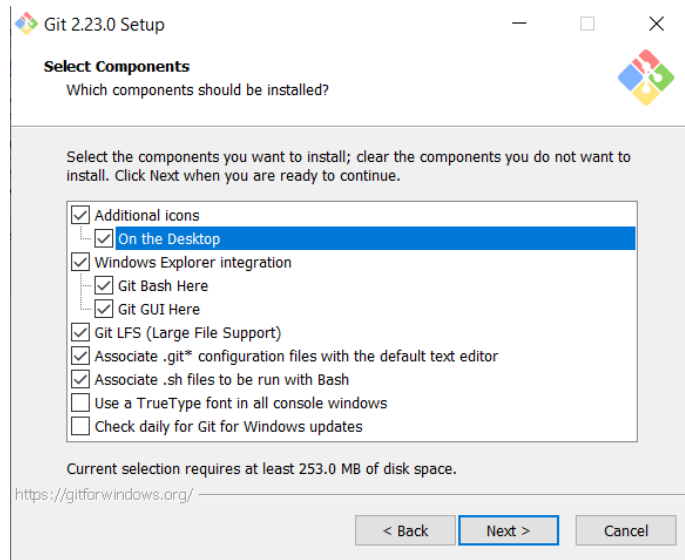
### Community

Get involved! Bug reporting, mailing list, chat, development and more.

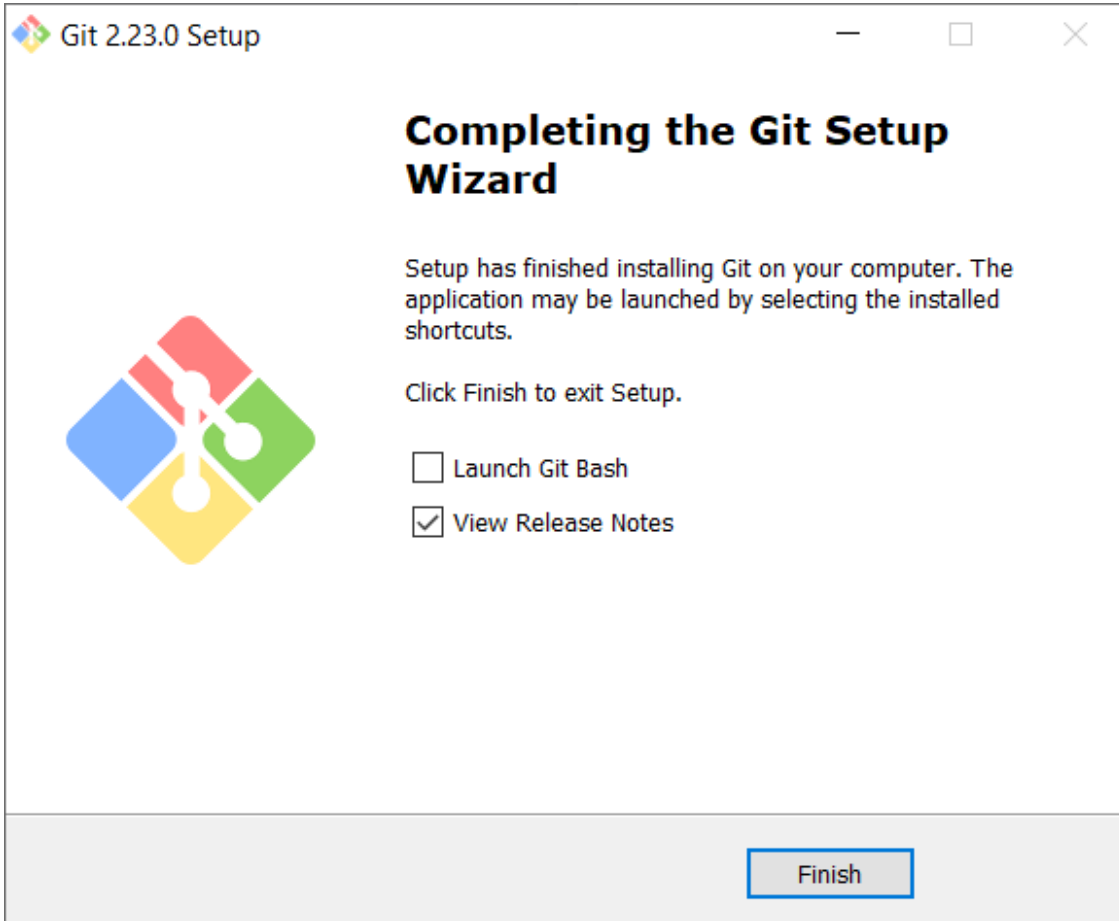


Latest source Release  
**2.23.0**  
Release Notes (2019-08-16)  
Download 2.23.0 for Windows

# Git (Cont.)



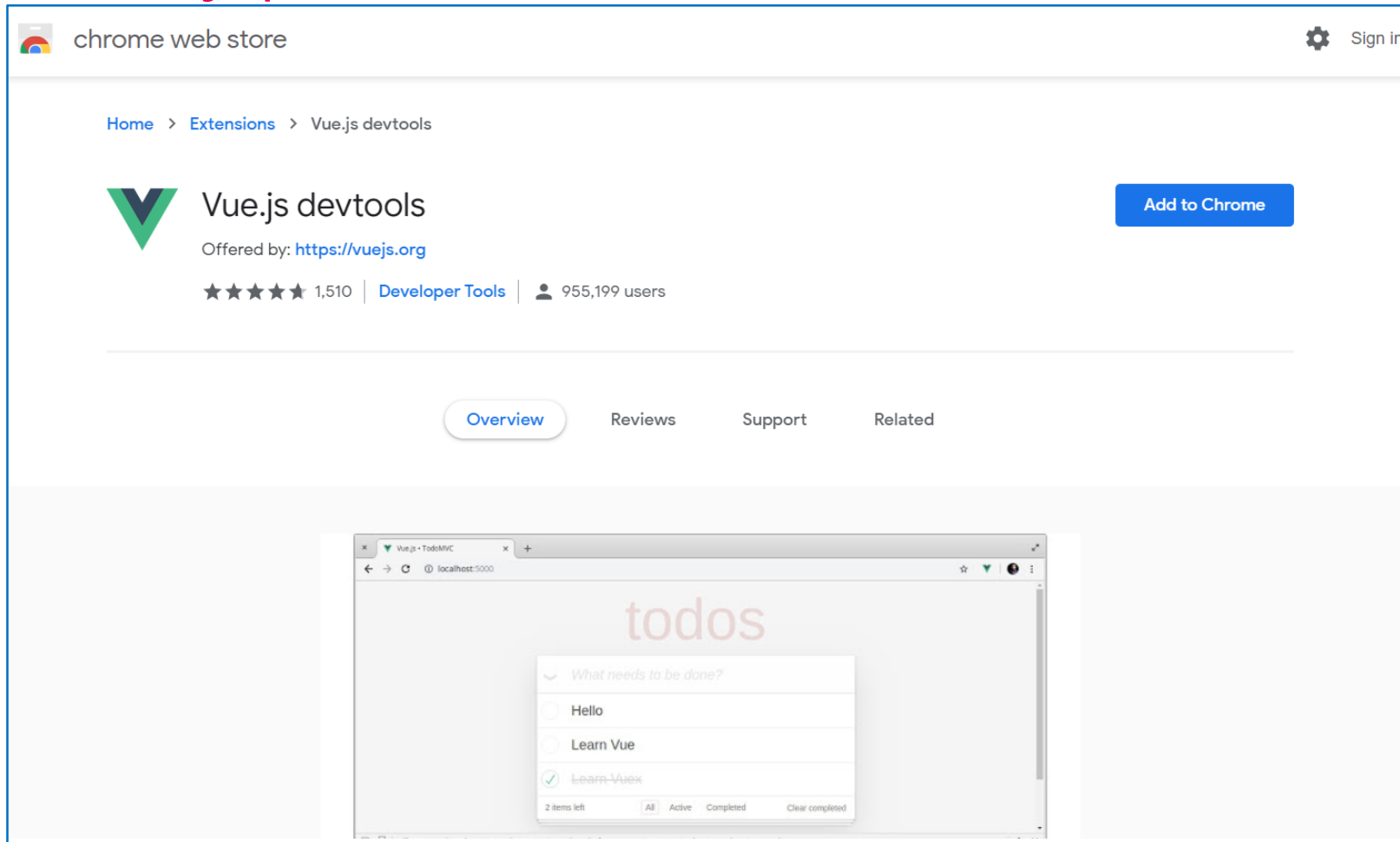
# Git (Cont.)



```
C:\Users\bluee>git --version  
git version 2.23.0.windows.1
```

# Vue.js devtools

- <https://chrome.google.com/webstore/detail/vuejs-devtools/nhdogjmejiglipccpnnnanhbledajbpd>





# Hello, Vue.js – First Way

## ■ 독립 실행 Version

- Direct **<script>** Include
- Simply download and include with a script tag.
- Vue will be registered as a global variable.
- Vuejs.org에서 Download
- Development Version
  - <https://vuejs.org/js/vue.js>
  - With full warnings and debug mode
  - 341,462 Bytes
- Production Version
  - <https://vuejs.org/js/vue.min.js>
  - Warnings stripped, 33.30KB min+gzip
  - 93,675 Bytes

# Hello, Vue.js – First Way (Cont.)

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7   <title>Hello Vue.js</title>
8 </head>
9 <body>
10   <div id="app">
11     {{ message }}
12   </div>
13   <script src="vue.min.js"></script>
14   <script>
15     new Vue({
16       el : '#app',
17       data : {
18         message : 'Hello, Vue.js!'
19       }
20     });
21   </script>
22 </body>
23 </html>
```

# Hello, Vue.js – Second Way

## ■ 독립 실행 CDN Version

- Direct `<script>` Include
- Simply download and include with a script tag.
- Vue will be registered as a global variable.
- CDN에서 가져오기
  - `<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>`
  - `<script src="https://cdn.jsdelivr.net/npm/vue@2.6.0"></script>`
  - `<script type="module">`  
`import Vue from 'https://cdn.jsdelivr.net/npm/vue@2.6.10/dist/vue.esm.browser.js'`  
`</script>`
- You can browse the source of the NPM package at [cdn.jsdelivr.net/npm/vue](https://cdn.jsdelivr.net/npm/vue).

# Hello, Vue.js – Second Way (Cont.)

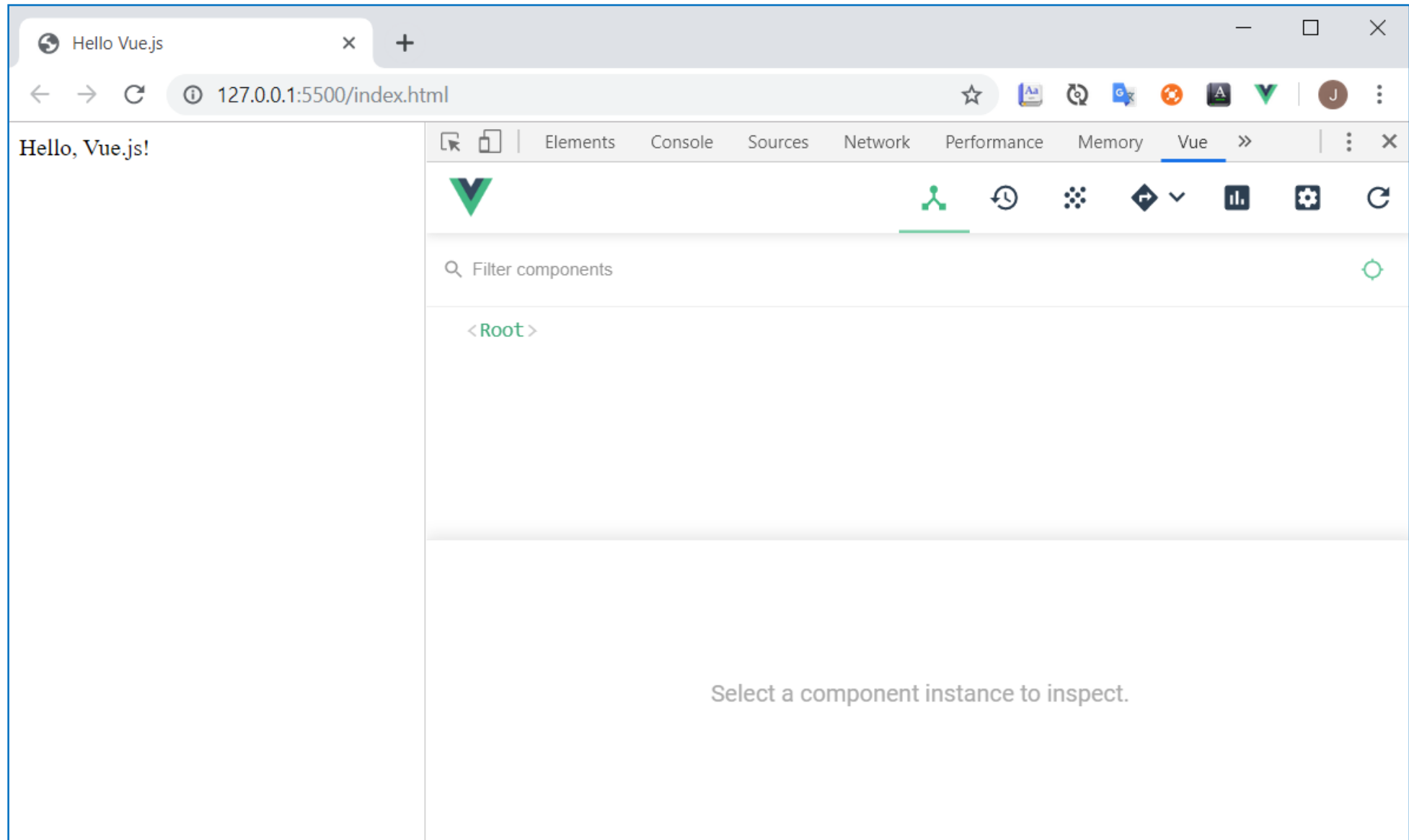
## ■ 독립 실행 CDN Version

- Vue is also available on [unpkg](#) and [cdnjs](#).
- cdnjs takes some time to sync so the latest release may not be available yet.
- Make sure to :
  - Read about [the different builds of Vue](#)
  - Use the production version.
  - Replacing vue.js with vue.min.js.

# Hello, Vue.js – Second Way (Cont.)

```
index.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <meta http-equiv="X-UA-Compatible" content="ie=edge">
7      <title>Hello Vue.js</title>
8  </head>
9  <body>
10     <div id="app">
11         {{ message }}
12     </div>
13     <script src="https://cdn.jsdelivr.net/npm/vue@2.6.0"></script>
14     <script>
15         new Vue({
16             el : '#app',
17             data : {
18                 message : 'Hello, Vue.js!'
19             }
20         });
21     </script>
22 </body>
23 </html>
```

# Hello Vue.js – Second Way (Cont.)



# Hello, Vue.js – Third Way

## ■ NPM 사용하기

- NPM is the recommended installation method when building large scale applications with Vue.
- It pairs nicely with module bundlers such as [Webpack](#) or [Browserify](#).
- Vue also provides accompanying tools for authoring Single File Components.

```
# latest stable
```

```
$ npm install vue
```

# Hello, Vue.js – Third Way (Cont.)

```
C:\VueHome>npm install vue
npm WARN saveError ENOENT: no such file or directory, open 'C:\VueHome\package.json'
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN enoent ENOENT: no such file or directory, open 'C:\VueHome\package.json'
npm WARN VueHome No description
npm WARN VueHome No repository field.
npm WARN VueHome No README data
npm WARN VueHome No license field.

+ vue@2.6.10
added 1 package from 1 contributor and audited 1 package in 1.849s
found 0 vulnerabilities
```



# Vue.js 개발 환경 선택하기

- 배포용 Version과 개발용 Version
- Npm과 Vue-CLI
- Build Tools

# Vue.js 개발 환경 선택하기 (Cont.)

## 배포용 Version과 개발용 Version

- Web Page에 Vue.js Library File을 포함하는 방법
- 배포용 Version
  - 개발자가 알아볼 수 없도록 난독화와 압축을 한 파일
  - 크기가 적고 파일을 편집기로 열었을 때 알아볼 수 없음.
  - Vue.js Project 개발을 완료하고 Web Application을 공개/배포 때 사용
  - vue.min.js
- 개발용 Version
  - Framework 내부에서 발생하는 경고 및 Debug Mode 모두 포함
  - 난독화하지 않은 File이므로 Vue.js Project 개발하면서 Vue.js 내부 동작을 살펴보기에 적합
  - vue.js

# Vue.js 개발 환경 선택하기 (Cont.)

## 배포용 Version과 개발용 Version

- Web page에 Vue.js Library file을 포함할 때는 CDN 사용 권장.
- 대표적인 CDN
  - <https://unpkg.com>
  - <https://www.jsdelivr.com>
  - <https://cdnjs.com>
- CDN에서 제공하는 Library File은 Version별로 구분해서 제공됨.
- NPM에 올라온 최신 Version을 사용할 때는 unpkg.com의 CDN 권장.

# Vue.js 개발 환경 선택하기 (Cont.)

## 배포용 Version과 개발용 Version

### ■ unpkg.com

- 개발용 : <https://unpkg.com/vue@<version>/dist/vue.js>
- 배포용 : <https://unpkg.com/vue@<version>/dist/vue.min.js>
- 개발 Version에 한정해 <https://unpkg.com/vue>로 적용 가능

### ■ jsDelivr.com

- 개발용 : <https://cdn.jsdelivr.net/npm/vue@<version>/dist/vue.js>
- 배포용 : <https://cdn.jsdelivr.net/npm/vue@<version>/dist/vue.min.js>

### ■ cdnjs.com

- 개발용 : <https://cdnjs.cloudflare.com/ajax/libs/vue/<version>/vue.js>
- 배포용 : <https://cdnjs.cloudflare.com/ajax/libs/vue/<version>/vue.min.js>

### ■ Vue.js Project 개발시 개발 편의성을 위해 개발용 Version 권장

# Vue.js 개발 환경 선택하기 (Cont.)

## NPM과 Vue-CLI 설치

### ■ NPM 설치

- Vue.js 개발하다보면 ES6를 사용하거나 Vue.js 프로그램을 쉽게 배포하기 위한 Bundling 할 때
- Npm을 사용해서 Vue.js를 설치하면 webpack, Browserify와 같은 Module Bundler를 사용할 수 있다.
- 대규모 Web Application을 개발하기에 좋다.
- Npm 명령어로 Vue.js를 설치하는 것은 Node.js Project에 적용할 때만 사용해야 한다.

```
$ mkdir node_project
```

```
$ cd node_project
```

```
$ npm init
```

```
$ npm install vue
```

# Vue.js 개발 환경 선택하기 (Cont.)

## NPM과 Vue-CLI 설치

### ■ Vue-CLI 설치

- Vue.js를 사용할 때 Vue 전용 CLI 도구인 Vue-CLI를 활용할 수 있다.
- Vue-CLI는 Node.js에 기반을 둔 Vue.js 프로그램의 빠른 시작과 구현을 위해 Build 설정이 포함된 Project를 생성한다.
- Vue-CLI 덕분에 Vue.js가 적용된 Node.js Project를 쉽게 만들 수 있다.
- Vue-CLI는 Project를 쉽게 Build할 수 있도록 몇 개의 Build Template을 제공한다.

# Vue.js 개발 환경 선택하기 (Cont.)

## Build Tools

- Vue.js를 Web Page나 Application에 쉽게 포함하는 방법은 script tag를 사용하는 것이다.
- JavaScript는 기본적으로 Module 化 기능을 지원하지 않으므로 Module化하려면 Build 도구를 사용해야 한다.
- UMD Build
  - Universal Module Definition
  - Browser와 Node.js에서 Vue.js File을 쉽게 사용할 수 있도록 참조해야 할 때 UMD 기반으로 작성한 Module File을 사용할 수 있다.
  - Vue.js는 2.x에서 vue.min.js와 vue.js외에 Runtime 배포용(vue.runtime.min.js), Runtime 개발용(vue.runtime.js) File을 따로 제공한다.

# Vue-CLI

- Evan You가 공식적으로 관리하는 Command line Interface 기반의 Scaffolding 도구.
- Vue.js App을 개발시 Project의 Directory 구조를 어떻게 잡을 것인지 Test, Build, Dependency 등을 어떻게 설정할 것인지에 대한 고민을 하지 않도록 Project의 기본 Template을 설정해 주는 Tool.
- Node.js 6.x 이상, npm version 3 이상의 환경 필요.
- Installation

**\$ npm install -g vue-cli**

```
C:\Users\bluee>npm install -g vue-cli_
```

**Scaffolding** : 원래 '공사장 등에서 사용하는 임시 가설물'이라는 뜻. Application을 개발할 때 처음부터 개발하는 것이 아니라 기본적인 Interface와 Tool을 제공해 준다. 개발자는 이 Tool을 활용해서 전체 App을 개발할 수 있다. - 원형섭, <Vue.js 쿼 스타트>, 2017.9, 루비페이퍼, p11.



## Vue-CLI (Cont.)

- 설치 후 설치 유무 및 Version 정보 확인

\$ **vue -V** 혹은 **vue --version**

```
C:\Users\bluee>vue --version  
2.9.6
```

- Vue-CLI로 만들 수 있는 Project Template을 알아보자.

\$ **vue list**

## Vue-CLI (Cont.)

```
C:\Users\bluee>vue list
```

Available official templates:

- ❓ **browserify** - A full-featured Browserify + vueify setup with hot-reload, linting & unit testing.
- ❓ **browserify-simple** - A simple Browserify + vueify setup for quick prototyping.
- ❓ **pwa** - PWA template for vue-cli based on the webpack template
- ❓ **simple** - The simplest possible Vue setup in a single HTML file
- ❓ **webpack** - A full-featured Webpack + vue-loader setup with hot reload, linting, testing & css extraction.
- ❓ **webpack-simple** - A simple Webpack + vue-loader setup for quick prototyping.

# Vue-CLI (Cont.)

## ■ simple

- 가장 간단단 Project Template 유형.
- HTML file 하나만 생성.
- 내부에는 Vue.js를 CDN(Content Delivery Network)방식으로 참조하고 있음.
- 가장 단순한 방법으로 기본적인 기능을 Test 해 볼 때 사용 가능.

## ■ browserify

- browserify + vueify 조합으로 hot-reload, linting, 단위 Test를 지원하는 Project Template을 구성.
- 대규모 Application을 개발할 때 주로 사용

## ■ browserify-simple

- 빠른 Prototyping을 위한 browserify + vueify가 설정된 간단한 Template 제공

# Vue-CLI (Cont.)

## ■ webpack

- Webpack + vue-loader 조합의 기능에 hot-reload, linting, 단위 Test 기능을 제공하는 Project Template.
- 대규모 Application 개발에 적절.

## ■ webpack-simple

- 빠른 Prototyping을 위한 Webpack + vue-loader가 설정된 간단한 Template 제공

## ■ pwa

- PWA(Progressive Web App) 개발을 위한 Template 제공.
- Webpack과 유사하지만 몇 가지 Library가 추가되어 있음.

## Vue-CLI (Cont.)

- Vue-CLI로 Project 기본 틀을 만드는 방법은 아래와 같다.

**\$ vue init [Template이름] [Project 이름]**

- @vue/cli-init module 설치하는 명령어

**\$ npm install @vue/cli-init -g**

```
C:\Users\bluee>npm install @vue/cli-init -g
npm WARN deprecated vue-cli@2.9.6: This package has been deprecated in favour of
@vue/cli
npm WARN deprecated coffee-script@1.12.7: CoffeeScript on NPM has moved to "coff
eescript" (no hyphen)
+ @vue/cli-init@3.12.0
added 257 packages from 208 contributors in 12.779s
```

## Vue-CLI (Cont.)

- hellovuejs Project

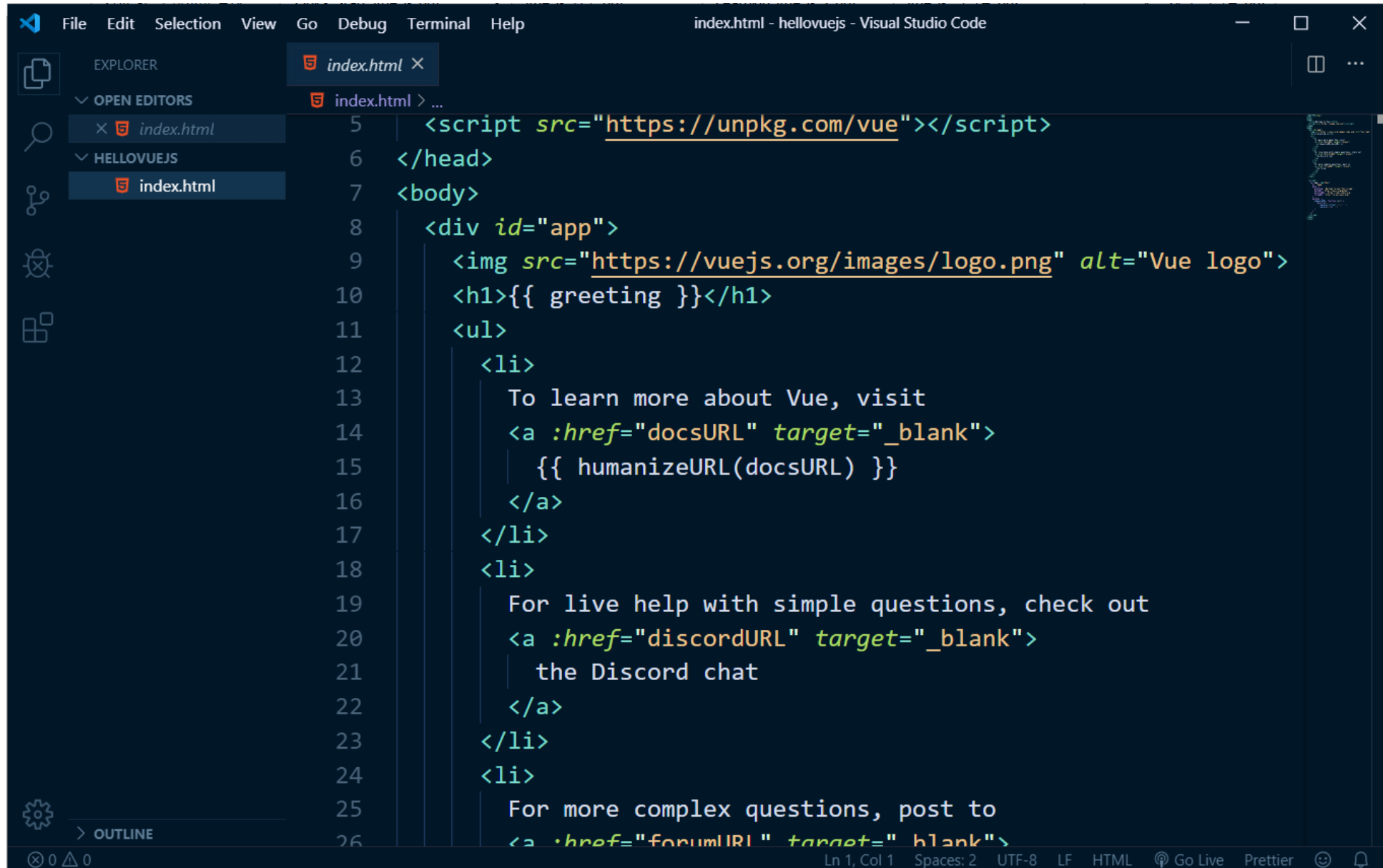
`$ vue init simple hellovuejs`

```
C:\VueHome>vue init simple hellovuejs
```

```
? name hellovuejs
```

```
vue-cli · Generated "hellovuejs".
```

# Vue-CLI (Cont.)



```
index.html - hellovuejs - Visual Studio Code

EXPLORER
  index.html x
  index.html > ...
  index.html
  HELLOVUEJS
    index.html

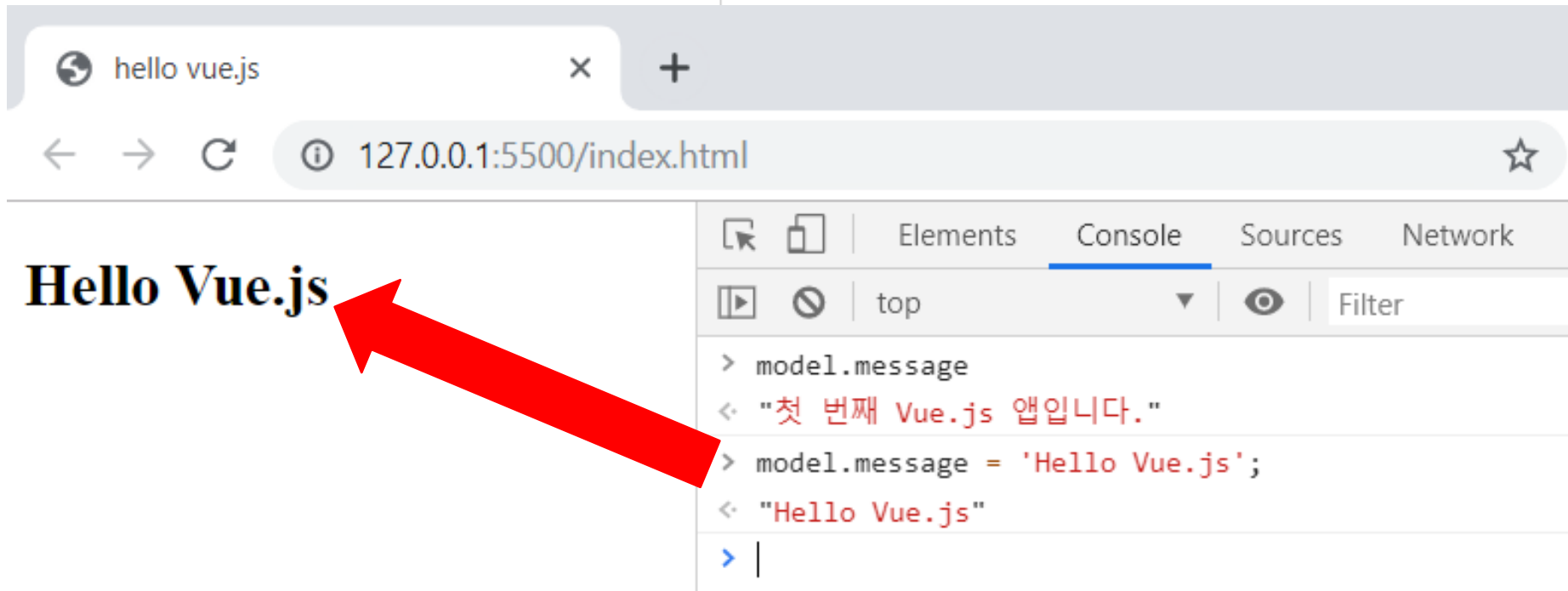
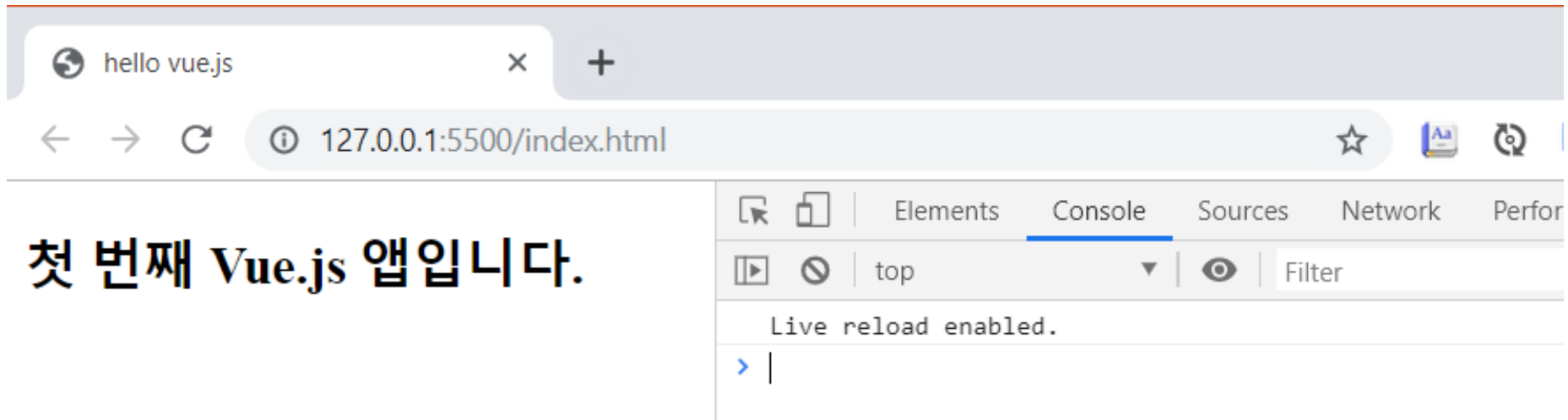
5 <script src="https://unpkg.com/vue"></script>
6 </head>
7 <body>
8   <div id="app">
9     
10    <h1>{{ greeting }}</h1>
11    <ul>
12      <li>
13        To learn more about Vue, visit
14        <a :href="docsURL" target="_blank">
15          {{ humanizeURL(docsURL) }}
16        </a>
17      </li>
18      <li>
19        For live help with simple questions, check out
20        <a :href="discordURL" target="_blank">
21          the Discord chat
22        </a>
23      </li>
24      <li>
25        For more complex questions, post to
26        <a :href="forumURL" target="_blank">
```

# Vue-CLI (Cont.)

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <meta charset="utf-8">
5    <title>hello vue.js</title>
6    <script src="https://unpkg.com/vue/dist/vue.min.js"></script>
7  </head>
8  <body>
9    <div id="simple">
10     <h2>{{message}}</h2>
11   </div>
12   <script>
13     var model = {
14       message : '첫 번째 Vue.js 앱입니다.'
15     };
16
17     var simple = new Vue({
18       el : '#simple',
19       data : model
20     });
21   </script>
22 </body>
23 </html>
```



# Vue-CLI (Cont.)



# Vue-CLI (Cont.)

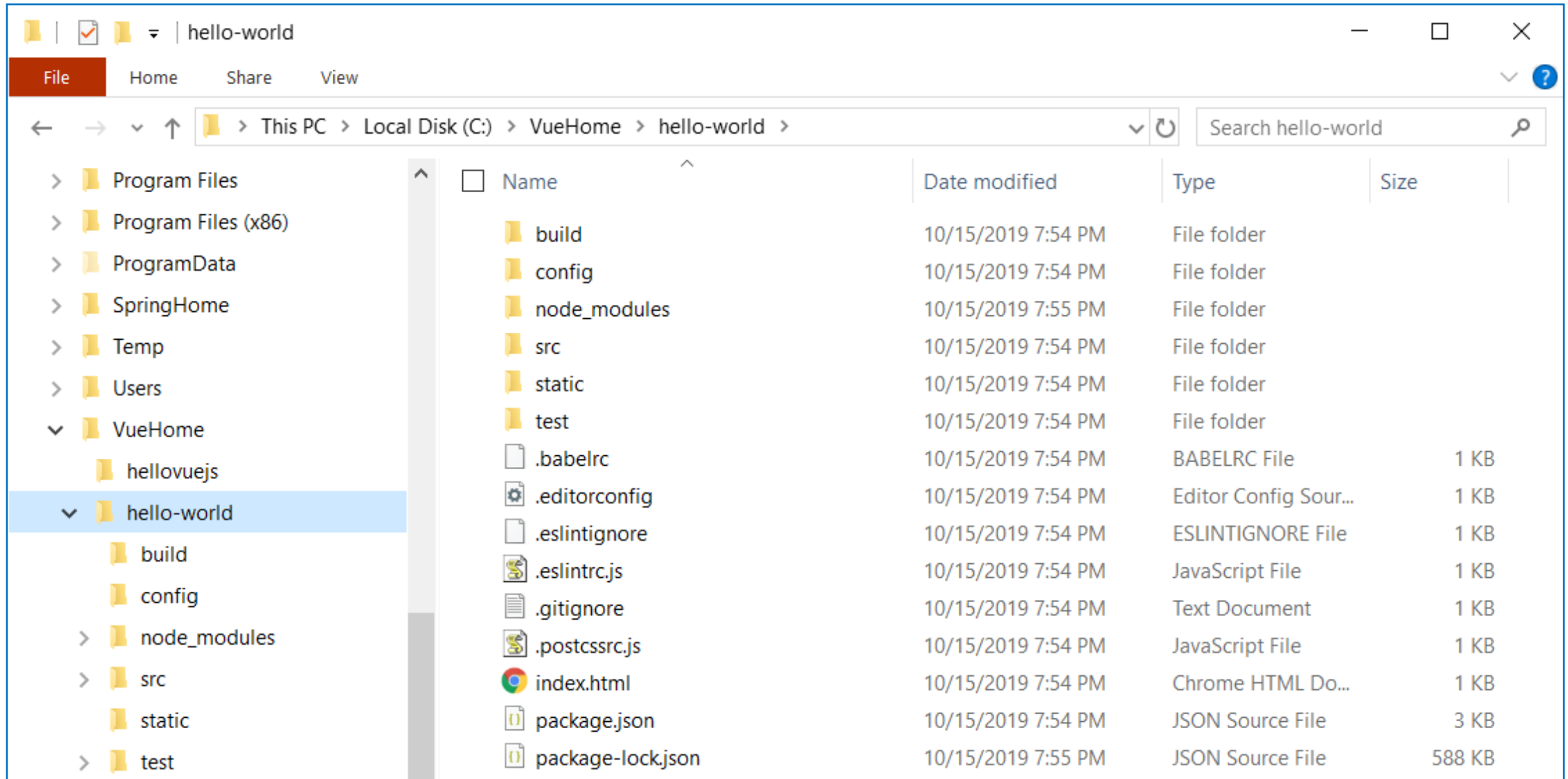
`$ vue init webpack hello-world`

```
C:\VueHome>vue init webpack hello-world
```

```
? Project name hello-world
? Project description A Vue.js project
? Author instructor
? Vue build standalone
? Install vue-router? Yes
? Use ESLint to lint your code? Yes
? Pick an ESLint preset Standard
? Set up unit tests Yes
? Pick a test runner jest
? Setup e2e tests with Nightwatch? Yes
? Should we run `npm install` for you after the project has been created? (recommended) npm
```

```
vue-cli · Generated "hello-world".
```

# Vue-CLI (Cont.)



# Vue-CLI (Cont.)

```
C:\VueHome>cd hello-world
```

```
C:\VueHome\hello-world>npm run dev
```

```
> hello-world@1.0.0 dev C:\VueHome\hello-world
```

```
> webpack-dev-server --inline --progress --config build/webpack.dev.conf.js
```

```
13% building modules 28/36 modules 8 active ...o-world\src\components\HelloWorld.vue{ parser: "babylon" } is deprecated; we now treat it as { parser: "babel" }.  
95% emitting
```

```
DONE Compiled successfully in 3968ms
```

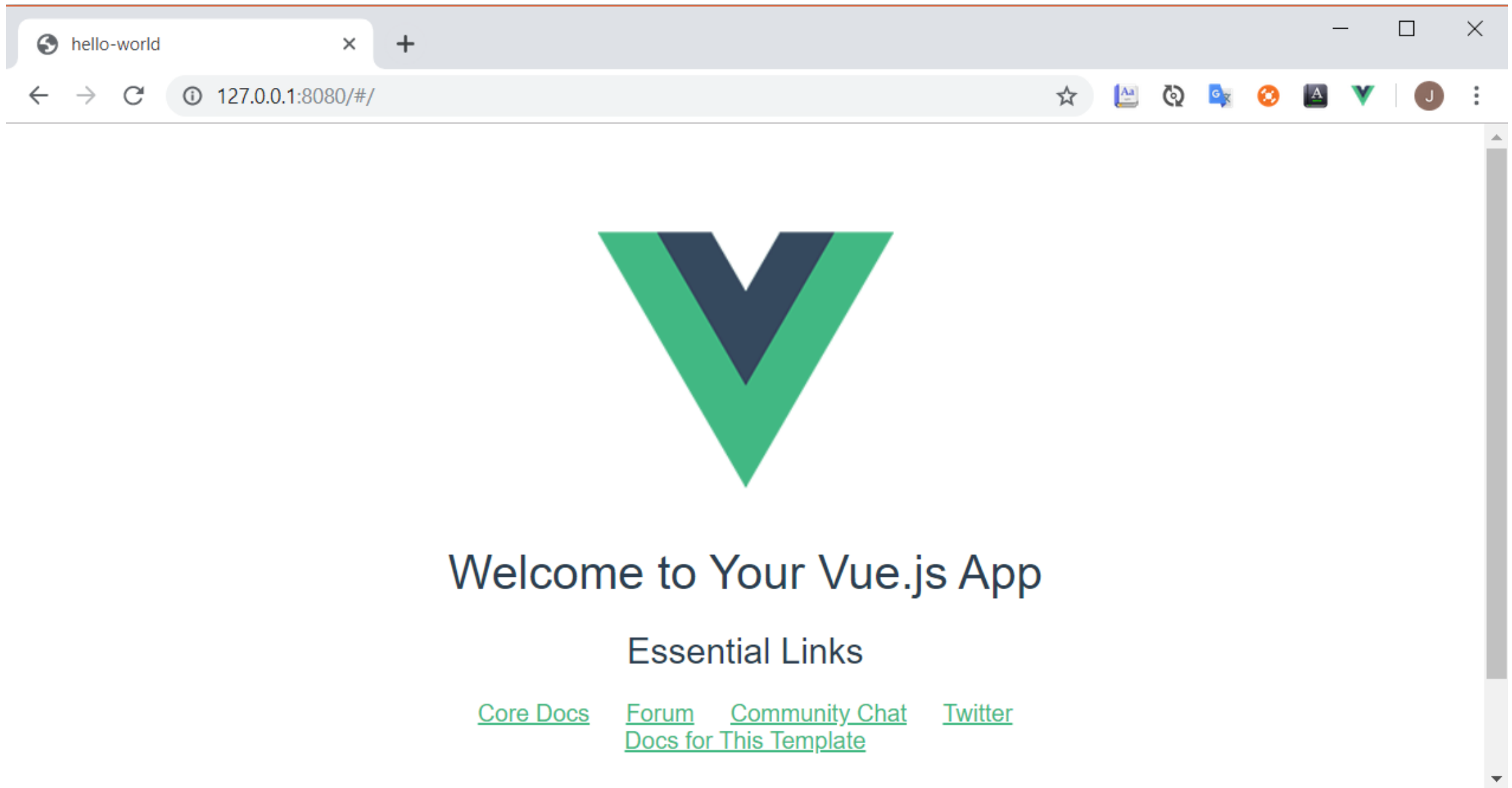
```
8:02:03 PM
```

```
I Your application is running here: http://localhost:8080
```

# Vue-CLI (Cont.)



# Vue-CLI (Cont.)

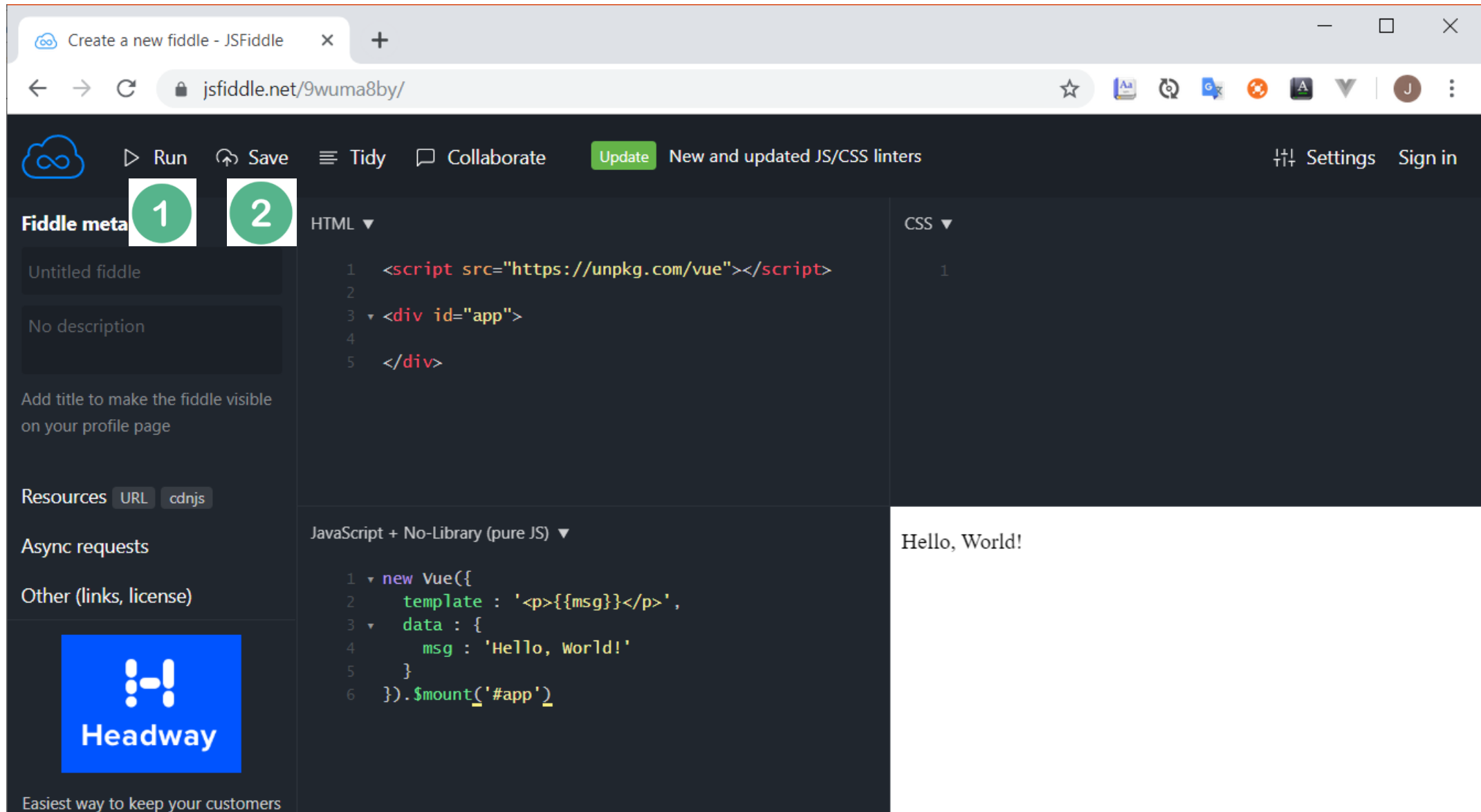


# JSFiddle을 사용한 실습 방법

The screenshot shows the JSFiddle website interface. At the top, there's a navigation bar with the JSFiddle logo and a 'Close' button. Below this, the 'Start with a boilerplate:' section offers various options like jQuery, Vue, React, React + JSX, Preact, TypeScript, CoffeeScript, SCSS, CSS Grid (responsive), Bootstrap, and PostCSS (marked as 'New'). A link to 'Show boilerplate bar less often?' is also present. To the right, the 'Links:' section includes links to the Roadmap, Bug tracker, About, Docs, and Service status. Further right, the 'JSFiddle is for:' section lists its uses: Demos for docs, Bug reporting, Presenting code answers, Live code collaboration, and Code snippets hosting. The main editor area is divided into three panels: 'Fiddle meta' on the left, 'HTML' in the center, and 'CSS' on the right. The 'Fiddle meta' panel includes fields for 'Untitled fiddle', 'No description', and a note to 'Add title to make the fiddle visible on your profile page'. It also has tabs for 'Resources' (URL, cdnjs) and sections for 'Async requests' and 'Other (links, license)'. The 'HTML' panel shows a single line of code. The 'CSS' panel shows a single line of code. At the bottom, there's a 'JavaScript + No-Library (pure JS)' section with a 'Result' panel.

JSFiddle interface showing boilerplate selection (jQuery, Vue, React, etc.), links, and the main editor area with HTML, CSS, and JavaScript panels.

# JSFiddle을 사용한 실습 방법 (Cont.)



The screenshot displays the JSFiddle web application interface. The browser's address bar shows the URL `jsfiddle.net/9wuma8by/`. The interface includes a top navigation bar with options like 'Run', 'Save', 'Tidy', 'Collaborate', 'Update', and 'Settings'. On the left, the 'Fiddle meta' panel shows 'Untitled fiddle' and 'No description'. The main editor area is divided into three sections: 'HTML', 'CSS', and 'JavaScript + No-Library (pure JS)'. The 'HTML' section contains the following code:

```
1 <script src="https://unpkg.com/vue"></script>
2
3 <div id="app">
4
5 </div>
```

The 'CSS' section is empty. The 'JavaScript' section contains the following code:

```
1 new Vue({
2   template : '<p>{{msg}}</p>',
3   data : {
4     msg : 'Hello, World!'
5   }
6 }).$mount('#app')
```

The output area on the right displays 'Hello, World!'. A 'Headway' logo is visible in the bottom left corner of the interface.