

# Vue.js2 vs Vue.js3

# Vue.js 3

## • Vue.js 3에서 달라진 내용

1. 성능 개선  
→ 가상 DOM 최적화로 인한 성능 개선
2. Typescript 지원
3. **Composition API**  
→ 각각의 기능을 함수로 묶어 유지보수성 향상
4. 트리쉐이킹 강화  
→ 필요한 코드만 import하도록 하여 성능 대폭 향상됨
5. 기타 (텔레포트, 프래그먼트, 서스펜스, 리액티비티API 등...)

100

## 2버전

## Options API

[illegible]

## 3버전

## Composition API

# Vue.js 3

- **마이그레이션이 필요할 경우**

만약 실무 들어갔을 때 Vue.js 2버전에서 3버전으로 마이그레이션 작업을 할 경우 마이그레이션 가이드를 한번 정독 하고 진행하면 됨

<https://v3-migration.vuejs.org/breaking-changes/v-model.html>

# Vue.js 3

## • 문법 차이 - 1

CDN방식으로 사용 시  
js파일에 type module 추가 필요

### Vue.js 2

```
var vm = new Vue({
  el: '#method-handler',
  data: {
    name: 'Vue.js'
  },
  methods: {
    greet: function(event) {
      alert("Hello, " + this.name);
      console.log(event);
      if(event) {
        alert(event.target.tagName);
      }
    }
  }
})
```

### Vue.js 3

```
<script type="module">
  import { createApp } from 'vue'

  createApp({
    data() {
      return {
        name: 'Vue.js'
      }
    },
    methods: {
      greet: function(event) {
        alert("Hello, " + this.name);
        console.log(event);
        if(event) {
          alert(event.target.tagName);
        }
      }
    }
  }).mount('#method-handler')
```

# Vue.js 3

## • 문법 차이 - 2

### Vue.js 2

```
var vm = new Vue({
  el: '#method-handler',
  data: {
    name: 'Vue.js'
  },
  methods: {
    greet: function(event) {
      alert("Hello, " + this.name);
      console.log(event);
      if(event) {
        alert(event.target.tagName);
      }
    }
  }
})
```

Vue 생성자 대신  
createApp 함수 사용

### Vue.js 3

```
<script type="module">
import { createApp } from 'vue'

createApp({
  data() {
    return {
      name: 'Vue.js'
    }
  },
  methods: {
    greet: function(event) {
      alert("Hello, " + this.name);
      console.log(event);
      if(event) {
        alert(event.target.tagName);
      }
    }
  }
}).mount('#method-handler')
```

# Vue.js 3

## • 문법 차이 - 3

### Vue.js 2

```
var vm = new Vue({  
  el: '#method-handler',  
  data: {  
    name: 'Vue.js'  
  },  
  methods: {  
    greet: function(event) {  
      alert("Hello, " + this.name);  
      console.log(event);  
      if(event) {  
        alert(event.target.tagName);  
      }  
    }  
  }  
})
```

el 대신 mount 사용

### Vue.js 3

```
<script type="module">  
  import { createApp } from 'vue'  
  
  createApp({  
    data() {  
      return {  
        name: 'Vue.js'  
      }  
    },  
    methods: {  
      greet: function(event) {  
        alert("Hello, " + this.name);  
        console.log(event);  
        if(event) {  
          alert(event.target.tagName);  
        }  
      }  
    }  
  }).mount('#method-handler')
```

# Vue.js 3

## • 문법 차이 - 4

### Vue.js 2

```
var vm = new Vue({  
  el: '#method-handler',  
  data: {  
    name: 'Vue.js'  
  },  
  methods: {  
    greet: function(event) {  
      alert("Hello, " + this.name);  
      console.log(event);  
      if(event) {  
        alert(event.target.tagName);  
      }  
    }  
  }  
})
```

data가 함수 방식으로 변경

### Vue.js 3

```
<script type="module">  
  import { createApp } from 'vue'  
  
  createApp({  
    data() {  
      return {  
        name: 'Vue.js'  
      }  
    },  
    methods: {  
      greet: function(event) {  
        alert("Hello, " + this.name);  
        console.log(event);  
        if(event) {  
          alert(event.target.tagName);  
        }  
      }  
    }  
  }).mount('#method-handler')
```



# Vue.js 3

- 리스트 렌더링을 Vue.js3으로 변경 해보기

```
createApp({  
  data() {  
    return {  
      items: [  
        { id:"1", message : '아이템1'},  
        { id:"2", message : '아이템2'},  
        { id:"3", message : '아이템3'},  
        { id:"4", message : '아이템4'},  
        { id:"5", message : '아이템5'}  
      ]  
    }  
  }  
}).mount('#list-example')
```

```
<ul id="list-example">  
  <li v-for="item in items" :key="item.id">  
    {{ item.message }}  
  </li>  
</ul>
```



Vue.js3

# Vue.js 3 기초

## • 코드 구성

1. template : HTML 코드가 작성되는 공간
2. script : JavaScript, 뷰 인스턴스가 작성되는 공간
3. style : css 코드가 작성되는 공간

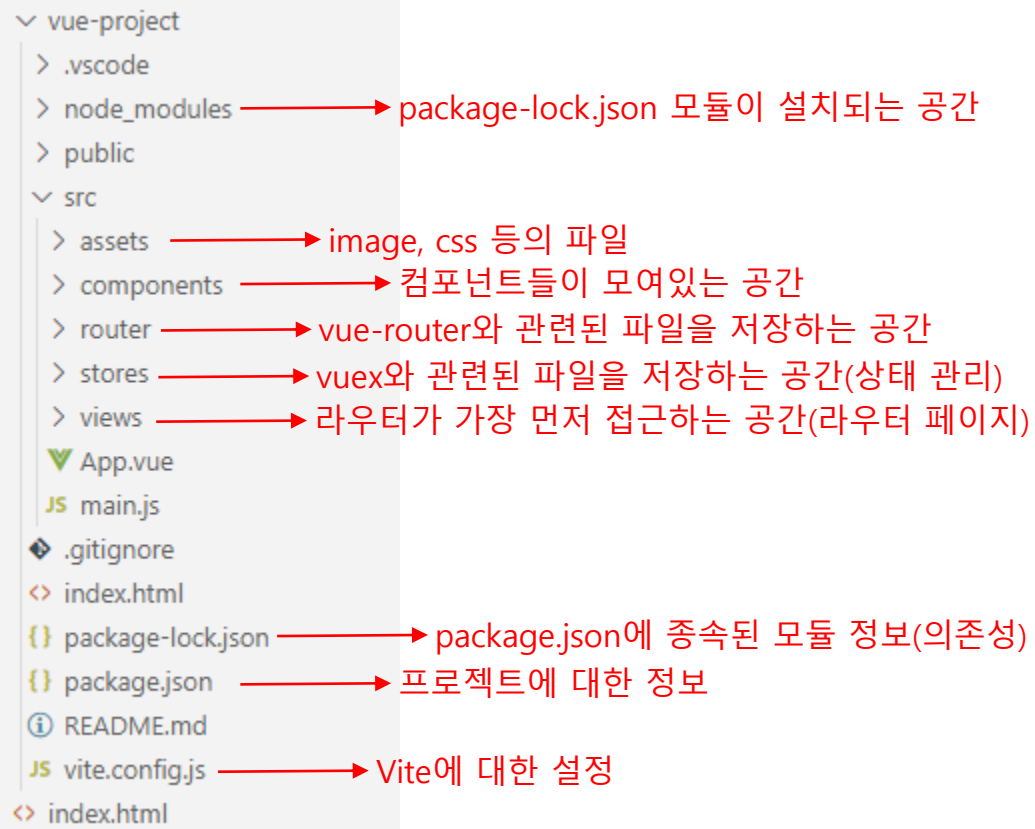
```
<template>  
  <!-- HTML 코드 -->  
</template>
```

```
<script>  
  // JavaScript - 뷰 인스턴스  
</script>
```

```
<style>  
  /* CSS */  
</style>
```

# Vue.js 3 기초

## • 파일 구조



# Vue.js 3 기초

## • Vuejs의 대표적인 라이브러리

1. vuex : 상태 관리
2. pinia : 상태 관리
3. Vue Router : SPA를 구현할 때 사용되며 페이지 이동 기능을 지원
4. Axios : JavaScript의 ajax와 유사한 기능을 수행. 백엔드 서버와 URL로 통신하여 데이터를 동적으로 가져오기 위한 동작을 수행함

# Vue.js 3 기초

- **Vuex**

컴포넌트 통신 방식인 props, event emit을 사용해야하기 때문에 로직이 복잡해 질수록 중간에 거쳐야 할 컴포넌트가 많아지는 단점을 없애기 위해 사용되는 라이브러리

- **Pinia**

Vuex와 마찬가지로 상태 관리하는 라이브러리  
VueConf Toronto 2021에서 Evan You가 상태 관리 라이브러리를 Vuex가 아닌 Pinia를 추천 하였으며, Pinia 공식 문서에 따르면 Vuejs가 Pinia를 공식 지원한다고 함

\* 그렇다고 Vuex의 지원이 끊기는것이 아니라 그대로 유지됨

# Vue.js 3 기초

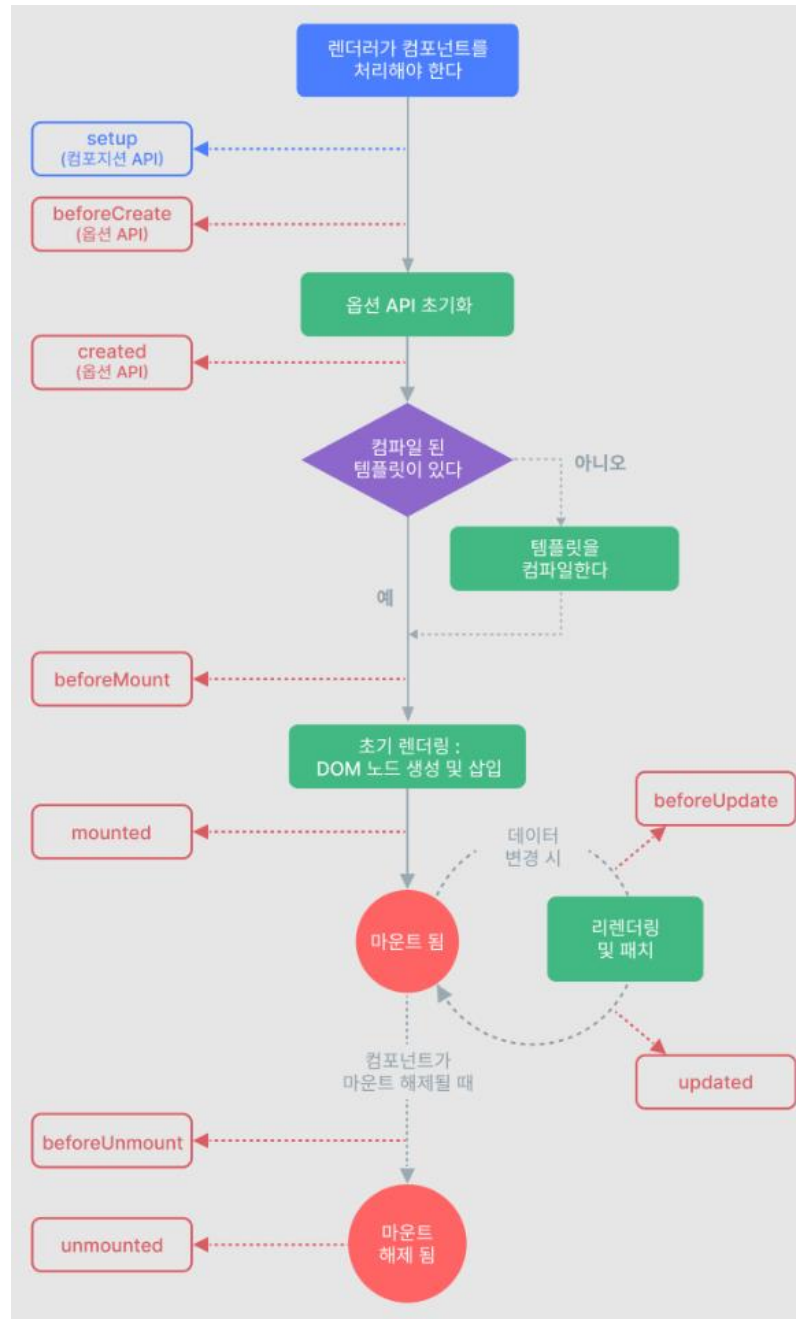
## • Pinia의 장점

1. Vuex보다 간단하고 직관적인 API
2. 여러 개의 모듈을 가질 수 있는 단일 store 제공
3. Devtool 지원
4. TypeScript 지원

# Vue.js3 Life Cycle



# Vue.js 3



# Vue.js 3 셋팅 - 1

## (Node.js)

# Vue.js 3 셋팅 – Node.js

- 설명

구글에 node.js 검색 후 다운로드 클릭



# Vue.js 3 셋팅 – Node.js


## • 설명

각자 환경에 맞게 설치 (LTS)

**다운로드**  
최신 LTS 버전: **18.16.0** (includes npm 9.5.1)  
플랫폼에 맞게 미리 빌드된 Node.js 인스톨러나 소스코드를 다운받아서 바로 개발을 시작하세요.

**LTS**  
대다수 사용자에게 추천

**현재 버전**  
최신 기능

  
Windows Installer  
node-v18.16.0-x64.msi

  
macOS Installer  
node-v18.16.0.pkg

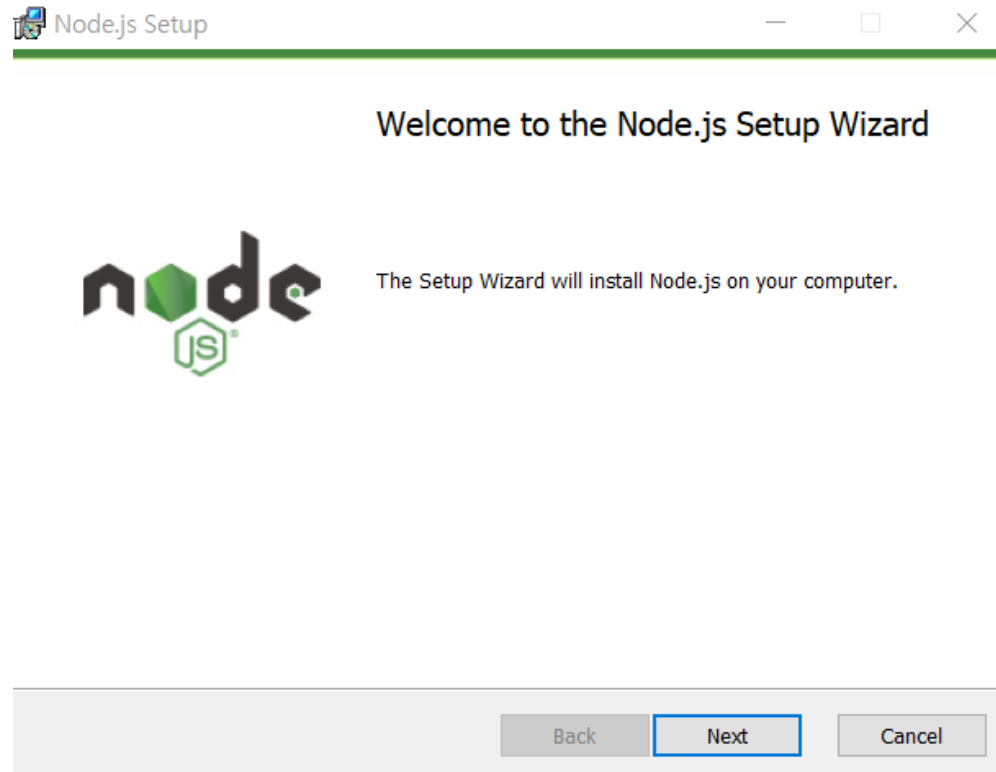
  
Source Code  
node-v18.16.0.tar.gz

Windows Installer (.msi)	32-bit	64-bit
Windows Binary (.zip)	32-bit	64-bit
macOS Installer (.pkg)	64-bit / ARM64	
macOS Binary (.tar.gz)	64-bit	ARM64
Linux Binaries (x64)	64-bit	
Linux Binaries (ARM)	ARMv7	ARMv8
Source Code	node-v18.16.0.tar.gz	

# Vue.js 3 셋팅 – Node.js

- 설명

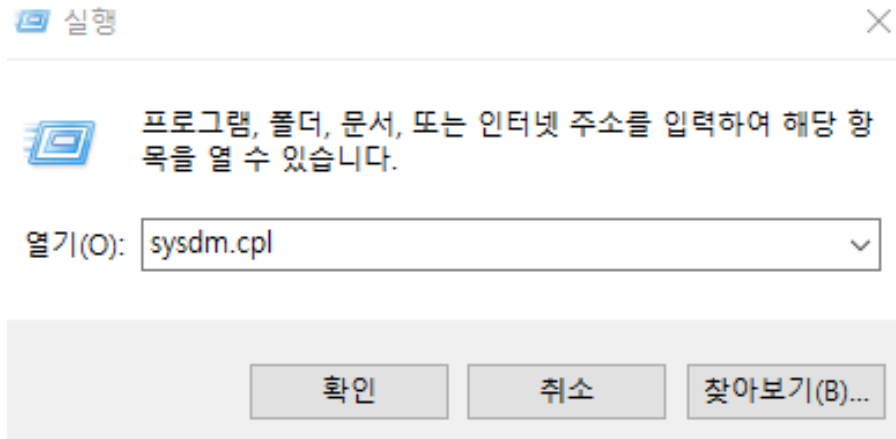
Setup 파일 실행 후 꼭 Next 클릭하여 설치 진행



# Vue.js 3 셋팅 – Node.js

## • 설명

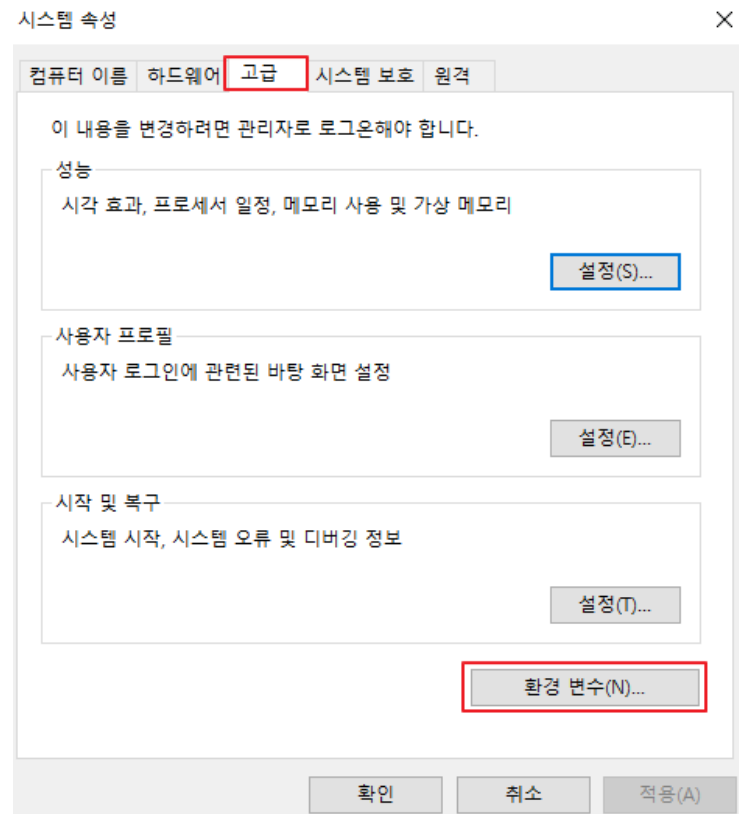
1. 윈도우키+R
2. sysdm.cpl 입력 후 엔터



# Vue.js 3 셋팅 – Node.js

## • 설명

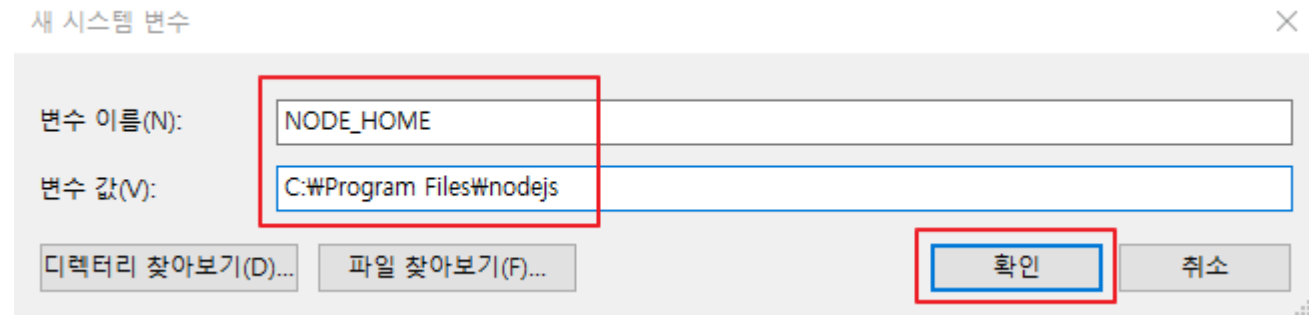
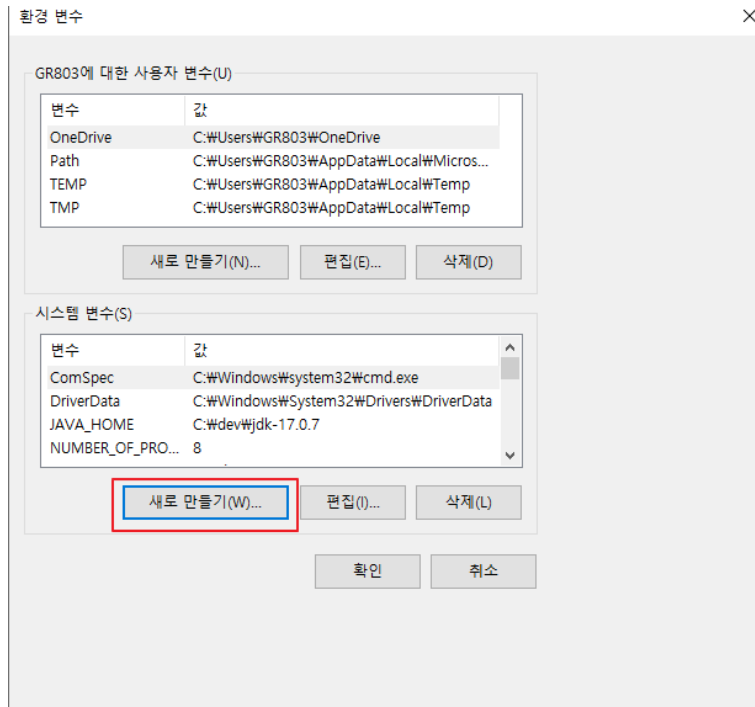
1. 고급 클릭
2. 환경 변수 클릭



# Vue.js 3 셋팅 – Node.js

## • 설명

1. 시스템 변수 → 새로 만들기
2. 변수 이름과 변수 값 작성 후 확인
  - 변수 이름 : NODE\_HOME
  - 변수 값 : C:\Program Files\nodejs





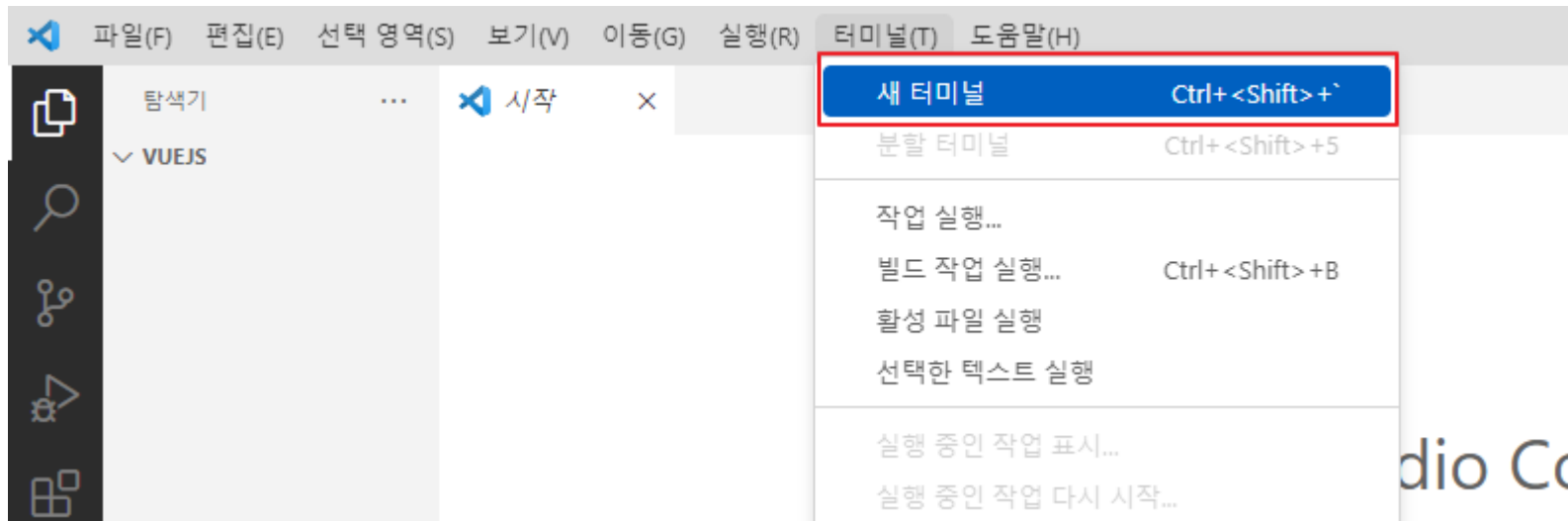
# Vue.js 3 셋팅 – 2

## (Vue install)

# Vue.js 3 셋팅 – Node.js

- 설명

VSCode 실행 후 작업 폴더를 따로 지정(ex. C:\wprj\vuejs) 후 터미널 → 새 터미널



# Vue.js 3 셋팅 – Node.js

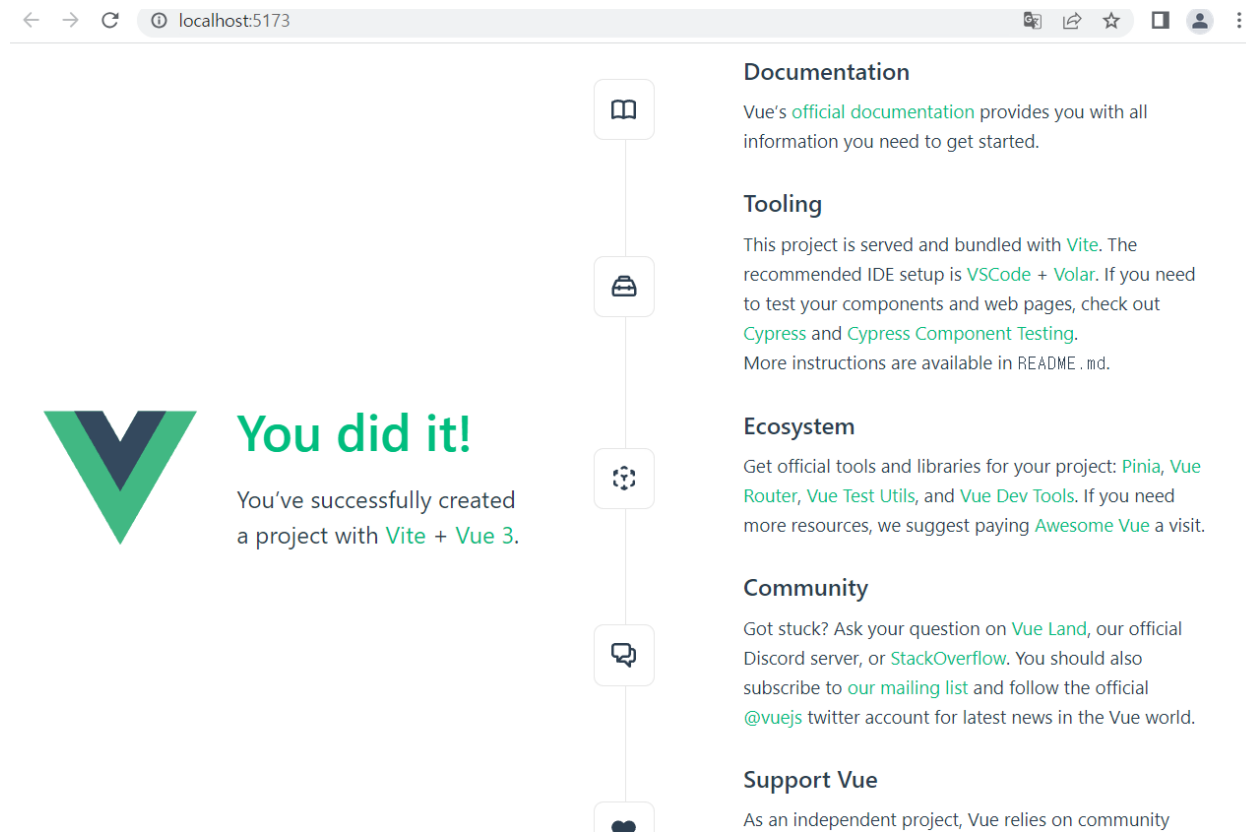
- 설명

1. npm init vue@latest  
→ Project name: 프로젝트명  
→ 그 외 No 선택
2. cd [프로젝트명]
3. npm install
4. npm run dev

# Vue.js 3 셋팅 – Node.js

## • 설명

아래와 같이 localhost 페이지에 접근이 가능하면 성공



# Vue.js 3 셋팅 – Node.js

## • 참고

자바스크립트 프로그래밍 언어를 위한 패키지 관리자로서 대부분의 필요한 패키지들은 npm으로 설치 가능함

- npm : Node.js의 기본 패키지 관리자
- yarn : 페이스북에서 개발한 패키지 관리자 (React 환경에서 많이 쓰임)

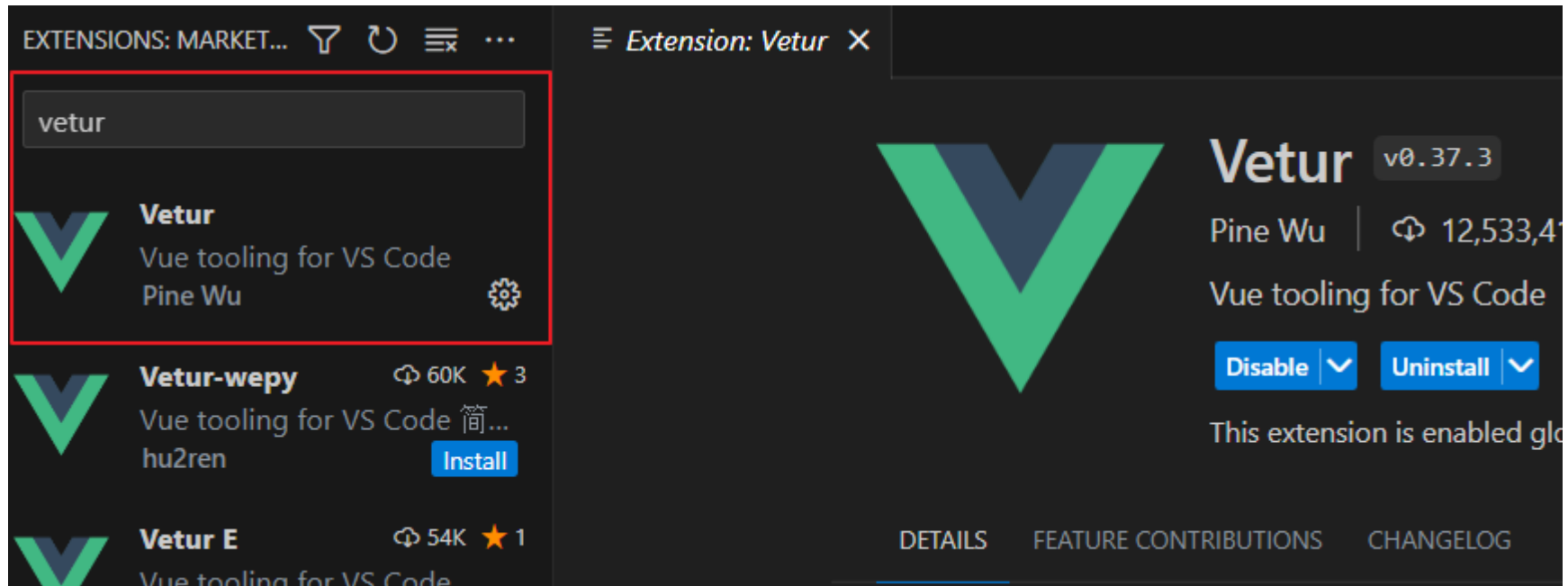
# Vue.js 3 셋팅 – 3

## (뷰 관련 플러그인)

# Vue.js 3 셋팅 – 뷰 관련 플러그인

- Vetur

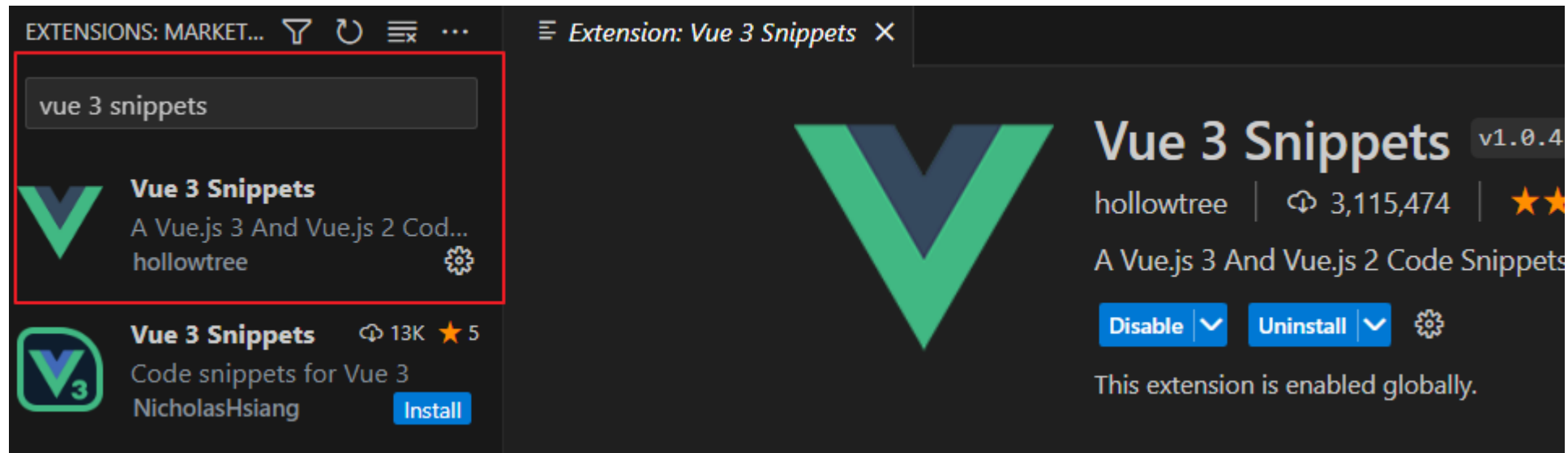
.vue 파일의 코드 하이라이팅 기능을 지원  
→ 언어의 문법에 맞춰 개발자가 보기 쉽게 해주는 기능



# Vue.js 3 셋팅 – 뷰 관련 플러그인

- **Vue 3 Snippets**

.vue 파일의 처음 생성할 때 필요한 코드를 스니펫으로 지정하여 사용 가능  
→ 초기 셋팅 : vuelnit

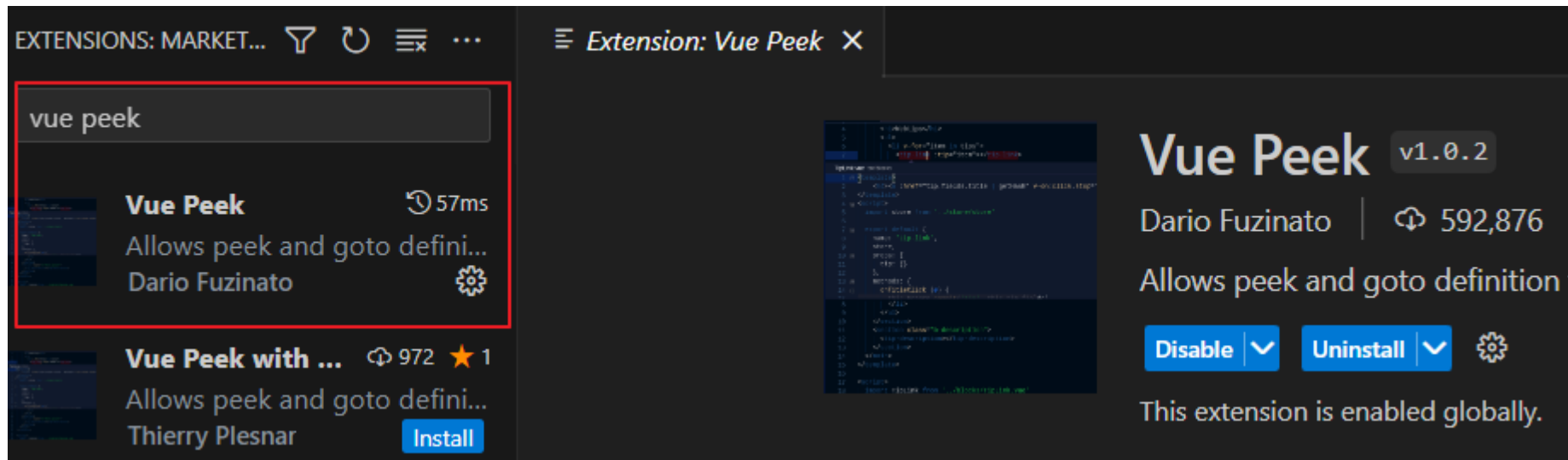




# Vue.js 3 셋팅 – 뷰 관련 플러그인

- Vue Peek

vue 프로젝트 내에서 참조되는 템플릿이나 모듈을 추적하는 기능



# Vue.js 3 셋팅 – 4

## (기타 플러그인)

# Vue.js 3 셋팅 – VSCode Plugin

- 설명

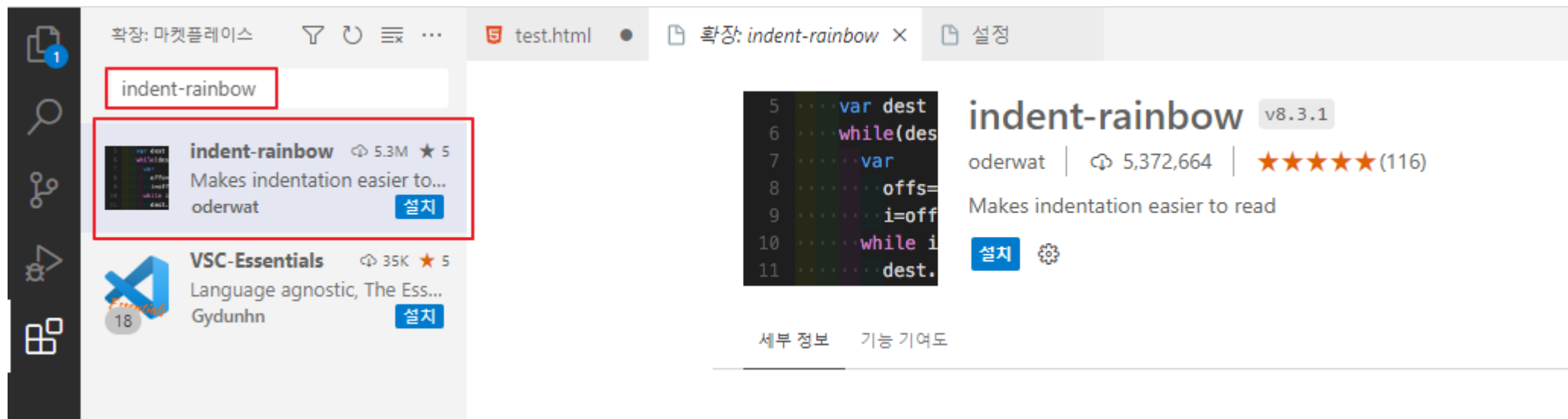
뷰 개발에 필요한 플러그인들과 사용하면 좀 더 편하게 개발할 수 있도록 도와주는  
플러그인들을 설치

→ 그 외 HTML, CSS, 가독성 향상 등의 플러그인들도 편한것들이 많으니 한번 찾아보면 좋음

# Vue.js 3 셋팅 – VSCode Plugin

- indent-rainbow

들여쓰기의 색상을 레인보우 컬러로 변경하여 코드의 가독성을 향상시켜주는 플러그인



# Vue.js 3 셋팅 – VSCode Plugin

- indent-rainbow 적용 후

들여쓰기 마다 색을 넣어줌

```
1  <!DOCTYPE html>
2  <html lang="ko-kr">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1">
6    <title>Document</title>
7  </head>
8  <body>
9    <div>
10     <ul>
11       <li></li>
12       <li></li>
13       <li></li>
14       <li></li>
15     </ul>
16   </div>
17 </body>
18 </html>
```

# Vue.js 3 셋팅 – VSCode Plugin

- **Bracket Paris (VSCode 내부 기능)**

최신 버전의 VSCode는 Bracket Paris라는 플러그인의 기능을 별도의 설치 없이 VSCode 내에서 설정이 가능하도록 되어있음

# Vue.js 3 셋팅 – VSCode Plugin

- Bracket Paris (VSCode 내부 기능)

1. Ctrl + ,
2. @id:editor.bracketPairColorization.enabled @id:editor.guides.bracketPairs 입력
3. 아래 사진의 영역을 false에서 true로 변경

Editor › Guides: Bracket Pairs

대괄호 쌍 안내선의 사용 여부를 제어합니다.

true



# Vue.js 3 셋팅 – VSCode Plugin

- 적용 후

각 괄호에 대한 영역을 안내선으로 표시해줌

```
<script>  
  function test() {  
    if(true) {  
        
    }  
  }  
</script>
```



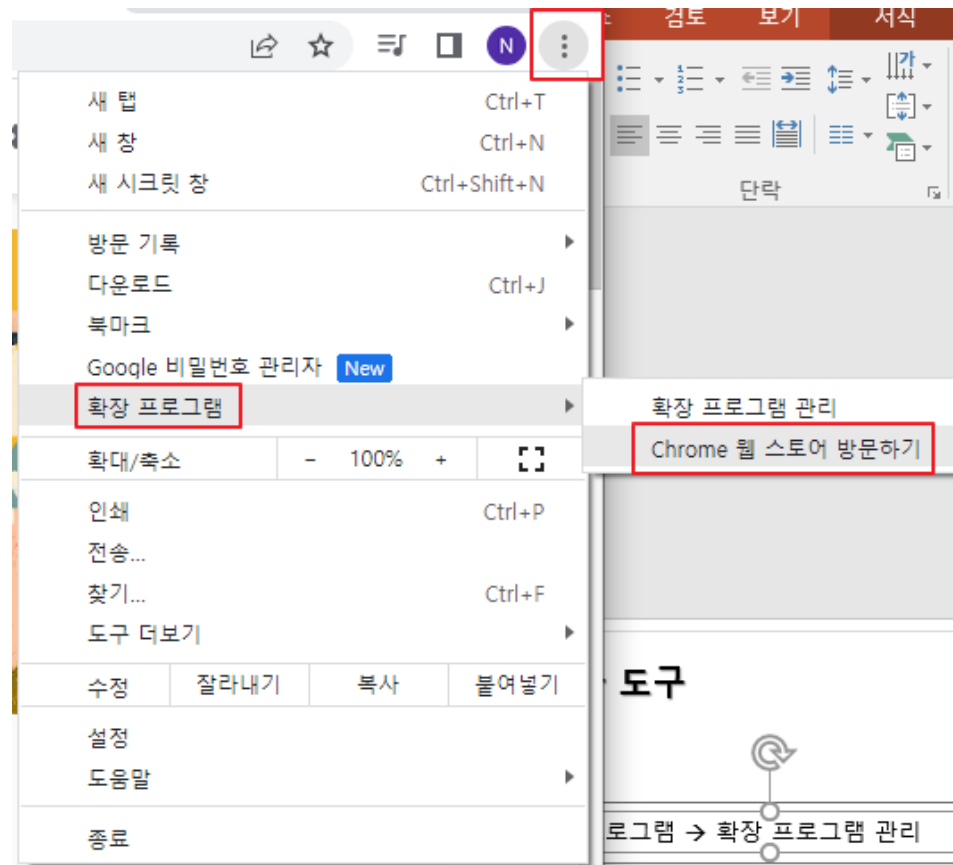
# Vue.js 3 셋팅 - 4

## (뷰 개발자 도구)

# Vue.js 3 셋팅 - 뷰 개발자 도구

## • 뷰 개발자 도구

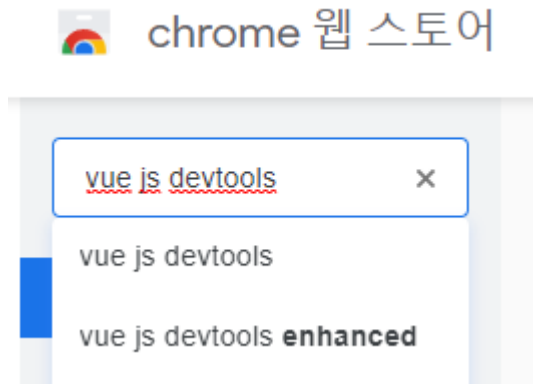
크롬 우측 상단 ... → 확장 프로그램 → Chrome 웹 스토어 방문하기



# Vue.js 3 셋팅 – 뷰 개발자 도구

- 뷰 개발자 도구

좌측에 'vue js devtools' 검색



# Vue.js 3 셋팅 - 뷰 개발자 도구

- 뷰 개발자 도구

클릭

확장 프로그램 ⓘ

[확장 프로그램 더보기](#)



Vue.js devtools

 [vuejs.org](https://vuejs.org)

Browser DevTools extension for debugging Vue.js applications.

★★★★★ 1,878 개발자 도구

# Vue.js 3 셋팅 – 뷰 개발자 도구

- 뷰 개발자 도구

Chrome에 추가

[홈](#) > [확장 프로그램](#) > [Vue.js devtools](#)



Vue.js devtools

 [vuejs.org](https://vuejs.org)

★★★★★ 1,878 ⓘ | [개발자 도구](#) | 사용자 2,000,000+명

Chrome에 추가

개요

개인 정보 보호 관행

리뷰

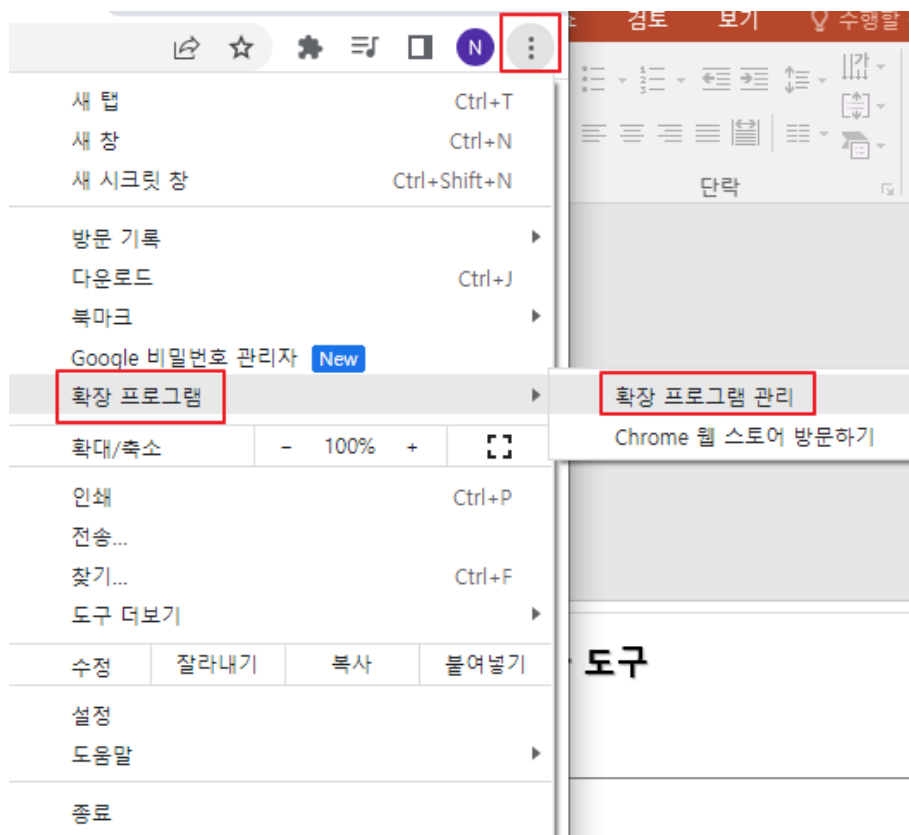
지원

관련 프로그램

# Vue.js 3 셋팅 – 뷰 개발자 도구

## • 뷰 개발자 도구

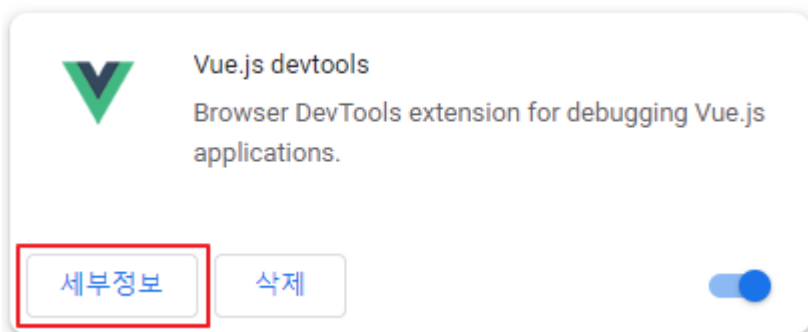
크롬 우측 상단 ... → 확장 프로그램 → 확장 프로그램 관리



# Vue.js 3 셋팅 - 뷰 개발자 도구

- 뷰 개발자 도구

세부 정보 클릭



# Vue.js 3 셋팅 – 뷰 개발자 도구

## • 뷰 개발자 도구

파일 URL에 대한 액세스 허용





# Vue.js 3 셋팅 - 뷰 개발자 도구

- 뷰 개발자 도구

Vue 페이지에서 개발자 도구켰을 때 Vue 생긴 내용 확인

\* 뷰 실행 후(npm run) 해당 페이지에서 확인해볼 것

