

공개SW솔루션 설치 & 활용 가이드

응용SW > 콘텐츠 배포



Vue.js

제대로 배워보자

How to Use Open Source Software

Open Source Software Installation & Application Guide



오픈소스 소프트웨어 통합지원센터
Open Source Software Support Center



CONTENTS

1. 개요
2. 기능요약
3. 실행환경
4. 설치 및 실행
5. 기능소개
6. 활용예제
7. FAQ
8. 용어정리

1. 개요



소개	<ul style="list-style-type: none">• 사용자 인터페이스를 만들기 위한 진보적인 자바스크립트 프론트엔드 프레임워크• 점진적으로 채택할 수 있게 설계된 프레임워크• 학습 곡선이 낮아 기반 지식이 있다면 쉽게 시작가능		
주요기능	<ul style="list-style-type: none">• HTML 템플릿문법을 사용하며, 선언적 렌더링을 함• 인스턴스 라이프사이클 혹 메소드를 사용함• 반응형이며, 컴포넌트 기반으로 코드를 캡슐화하는데 도움이 됨		
대분류	• 응용SW	소분류	• 콘텐츠 배포
라이선스형태	The MIT License (MIT)	사전설치 솔루션	Node.js (NPM을 사용할 경우)
		버전	v3.2.45(2022년 12월 기준)
특징	<ul style="list-style-type: none">• 양방향 데이터 바인딩과 Virtual DOM을 사용해 Angular와 React.js의 장점을 동시에 가짐• 라이브러리 사이즈가 작음 (58.8k)• 뷰와 상태 관리, SPA 라우터 등 핵심 요소에만 집중하며 높은 성능을 가짐• HTML과 CSS코드를 재사용 가능하게 함• 공식 문서의 품질이 좋음		
개발회사/커뮤니티	Evan You / https://dev.to/t/angular		
공식 홈페이지	https://angular.io/		



2. 기능요약

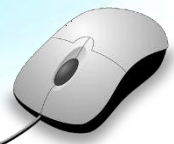


- Vue.js 주요 기능

HTML 템플릿 문법	Render 함수를 작성하지 않고 Virtual DOM으로 전환되는 HTML 템플릿으로 컴포넌트를 작성할 수 있으며 원하는 경우에 vue의 속성을 이용해 render 함수를 직접 작성 가능
컴포넌트	Vue 인스턴스를 컴포넌트로 등록하여 재사용할 수 있는 조합 가능한 컴포넌트로 만들 수 있음 컴포넌트 등록은 전역 또는 지역으로 가능
반응형	선언적 의존관계



3. 실행 환경



- Vue CLI를 사용하는 개발 환경 – Node.js 또는 yarn이 설치되어야 함
 - Windows
 - Linux
 - macOS 등
- Vue.js 지원 브라우저
 - Safari 6 이상
 - Chrome 23 이상
 - Firefox 21 이상
 - Edge 12 이상
 - IE 9 이상 (vue 3에서는 IE 11을 지원하지 않음)



4. 설치 및 실행



세부 목차

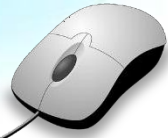
4.1 Vue CLI로 설치 및 실행 (Node.js와 yarn)

4.2 CDN을 직접 ``<script>``에 추가

- Vue.js 애플리케이션을 만들고 싶다면 4.1. 방법을,
앱을 바로 실행시켜보고 싶다면 4.3. 방법대로 설치 및 실행하는 것이 좋음
본 가이드에서는 코드 에디터로 VS Code를 사용하지만, 다른 텍스트 에디터를 사용해도 편집 가능
– VS Code 다운로드: <https://code.visualstudio.com/download>



4. 설치 및 실행

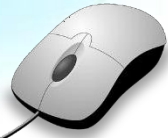


4.1 Vue CLI로 설치 및 실행 (Node.js와 yarn) (1/4)

- OS version: Windows 10
- Node.js version: v16.18.0
- Vue CLI를 이용하려면 Node.js와 yarn이 먼저 설치되어야 함
- Windows의 경우 <https://nodejs.org/ko/>에서 Node.js 설치
- Ubuntu의 경우 아래 순서대로 bash에서 진행
 - `$ sudo apt-get install curl`
 - `$ curl -sL https://deb.nodesource.com/setup_10.x | sudo -E bash -`
 - `$ sudo apt-get install nodejs`
- Node.js와 NPM 버전 확인 - Windows의 경우 cmd, Linux의 경우 bash에서
 - `~$ node -v`
 - `~$ npm -v`
 - 버전 숫자가 나오면 통과

```
soyun@soyun-Latitude-3520:~$ node -v
v16.18.0
soyun@soyun-Latitude-3520:~$ npm -v
8.19.2
```

4. 설치 및 실행



4.1 Vue CLI로 설치 및 실행 (Node.js와 yarn) (2/4)

- yarn 설치

```
~$ npm i -g yarn
```

- Vue CLI 설치

```
~$ yarn global add vue @vue/cli
```

- 버전 확인

```
~$ vue --version
```

- 프로젝트 생성

```
~$ vue create hello-world
```

- 화살표키를 이용해 아래 설정대로 생성

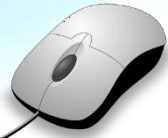
```
Vue CLI v5.0.8
? Please pick a preset: Default ([Vue 3] babel, eslint)
? Pick the package manager to use when installing dependencies: NPM
```

- 해당 디렉토리로 이동

```
~$ cd hello-world
```



4. 설치 및 실행



4.1 Vue CLI로 설치 및 실행 (Node.js와 yarn) (3/4)

- package.json의 scripts 확인

```
{
  "name": "hello-world",
  "version": "0.1.0",
  "private": true,
  > 디버그
  "scripts": {
    "serve": "vue-cli-service serve",
    "build": "vue-cli-service build",
    "lint": "vue-cli-service lint"
  },
}
```

- package.json에 있는 명령어를 사용해서 프로젝트를 로컬에서 실행

```
~$ yarn serve
```

```
DONE Compiled successfully in 3484ms

App running at:
- Local: http://localhost:8080/
- Network: http://192.168.0.22:8080/

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

- 콘솔에 나온 주소로 접속해서 앱 실행 여부 확인



Welcome to Your Vue.js App

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



4. 설치 및 실행



4.1 Vue CLI로 설치 및 실행 (Node.js와 yarn) (4/4)

- package.json에 있는 명령어를 사용해서 프로젝트를 빌드

```
$ yarn build
```

```
DONE Compiled successfully in 7721ms 14:39:55

File                Size      Gzipped
dist\js\chunk-vendors.1cc2a64c.js  82.98 KiB  29.99 KiB
dist\js\app.459c6d19.js           4.64 KiB   1.66 KiB
dist\css\app.e2713bb0.css          0.33 KiB   0.23 KiB

Images and other types of assets omitted.

DONE Build complete. The dist directory is ready to be deployed.
INFO Check out deployment instructions at https://cli.vuejs.org/guide/deployment.html

Done in 11.35s.
```

- 빌드된 결과물 확인

```
~$ ls distv
```

```
soyun@soyun-Latitude-3520:~/hello-world$ ls dist
css  favicon.ico  index.html  js
```

- 해당 빌드는 사용자 배포용으로 optimized되었음

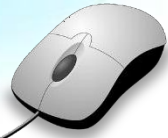


4. 설치 및 실행



4.2. CDN을 직접 `

5. 기능소개



세부 목차

5.1 HTML 템플릿문법

5.2 반응형, 조합 가능한 컴포넌트

5.3 data와 props

5.4 라이프사이클 훅 메소드

5.5 기능소개에서는 Vue CLI를 통해 생성한 프로젝트 소스를 사용



5. 기능소개



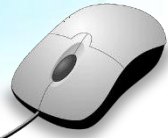
5.1 HTML 템플릿문법 (1/2)

- HTML 템플릿문법 사용 가능
- Vue.js 템플릿은 HTML 파서로 구문 분석이 가능한 유효한 HTML임
- Vue는 내부적으로 템플릿을 가상 DOM 렌더링 함수로 컴파일함
- 반응형 앱인 Vue는 앱의 상태가 변경될 때에 최소한의 DOM 조작을 통해 페이지를 변경

```
1 <template>
2   <div class="hello">
3     <h1>{{ msg }}</h1>
4     <p>
5       For a guide and recipes on how to configure / custom
6       check out the
7       <a href="https://cli.vuejs.org" target="_blank" rel=
8     </p>
9     <h3>Installed CLI Plugins</h3>
10    <ul>
11      <li><a href="https://github.com/vuejs/vue-cli/tree/de
12      <li><a href="https://github.com/vuejs/vue-cli/tree/de
13    </ul>
14    <h3>Essential Links</h3>
15    <ul>
16      <li><a href="https://vuejs.org" target="_blank" rel=
17      <li><a href="https://forum.vuejs.org" target="_blank
18      <li><a href="https://chat.vuejs.org" target="_blank"
19      <li><a href="https://twitter.com/vuejs" target="_bla
20      <li><a href="https://news.vuejs.org" target="_blank"
21    </ul>
22  </div>
23 </template>
```



5. 기능소개



5.1 HTML 템플릿문법 (2/2)

- 보간법

- 데이터를 바인딩하는 방법은 "콧수염 문법", 이중 중괄호를 사용
- 줄 3번의 `

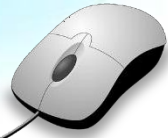
{{ msg }}

`에서, 실제 페이지에서는 {{ msg }}부분에 msg의 값이 담기게 됨

```
1 <template>
2   <div class="hello">
3     <h1>{{ msg }}</h1>
4     <p>
5       For a guide and recipes on how to configure / customi
6       check out the
7       <a href="https://cli.vuejs.org" target="_blank" rel="
8     </p>
9     <h3>Installed CLI Plugins</h3>
10    <ul>
11      <li><a href="https://github.com/vuejs/vue-cli/tree/d
12      <li><a href="https://github.com/vuejs/vue-cli/tree/d
13    </ul>
14    <h3>Essential Links</h3>
15    <ul>
16      <li><a href="https://vuejs.org" target="_blank" rel="
17      <li><a href="https://forum.vuejs.org" target="_blank"
18      <li><a href="https://chat.vuejs.org" target="_blank"
19      <li><a href="https://twitter.com/vuejs" target="_blan
20      <li><a href="https://news.vuejs.org" target="_blank"
21    </ul>
22  </div>
23 </template>
```



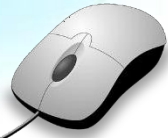
5. 기능소개



5.2 반응형, 조합 가능한 컴포넌트 (1/2)

- Vue는 단일 파일 컴포넌트 사용 가능
 - 단일 파일 컴포넌트: 하나의 파일에 템플릿, 컴포넌트, CSS를 담는 방식
 - 아래 사진은 앞의 HTML 템플릿 문법에도 나왔던 HelloWorld.vue 컴포넌트
 - `

5. 기능소개



5.2 반응형, 조합 가능한 컴포넌트 (2/2)

- HelloWorld보다 상위 컴포넌트인 App 컴포넌트에서 HelloWorld 컴포넌트를 사용한 모습
- App 컴포넌트를 렌더링하면 App 컴포넌트 템플릿의 내용이 페이지에 표시됨
- <div> 안쪽의 다음의 <HelloWorld> 자리에 HelloWorld 컴포넌트의 내용이 페이지에 표시됨
- 컴포넌트는 재사용 가능. 하단참조

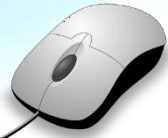
<https://vuejs.org/api/built-in-components.html>

```
<template>
  <div id="app">
    
    <HelloWorld msg="Welcome to Your Vue.js App"/>
    <HelloWorld msg="Welcome to Your Vue.js App"/>
  </div>
</template>
```

```
src > App.vue
You, 2 hours ago | 1 author (You)
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
```



5. 기능소개



5.3 data와 props (1/3)

- HelloWorld.vue 코드를 아래와 같이 변경
- <script>안에 data를 함수형으로 선언
 - 컴포넌트에서 data 안의 변수를 사용 가능
 - {{ customMessage }} 가 템플릿에서 data의 customMessage 변수를 사용한 모습
 - 페이지에서는 {{ customMessage }} 자리에 customMessage의 값인 'This is ...'가 나타나게 됨

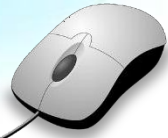
<https://vuejs.org/api/built-in-components.html>

```
4 <p>
5   {{ customMessage }}. Count: {{ count }}
6   <button v-on:click="inc">Increase Count</button>
7 </p>
8 <p>
9   For a guide and recipes on how to configure / cust
```

```
37 <script>
38 export default {
39   name: 'HelloWorld',
40   props: {
41     msg: String
42   },
43   data: () => ({
44     customMessage: 'This is customMessage in data',
45     count: 0,
46   }),
47   methods: {
48     inc() {
49       this.count = this.count + 1;
50     }
51   }
52 }
53 </script>
```



5. 기능소개



5.3 data와 props (2/3)

- props는 상위 컴포넌트에서 하위 컴포넌트로 데이터를 내려주는 방법
- 상위 컴포넌트에서는 하위로 값을 전달하고, 하위는 그 값을 사용
 - 하단 참조
- HTML의 속성 문법과 같은 형태로 하위에 전달, 하위에선 data 처럼 사용 가능
 - ``<HelloWorld msg="Welcome to Your Vue.js App" />``

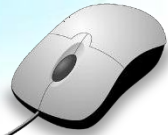
```
src > App.vue
You, 8 minutes ago | 1 author (You)
1 <template>
2   <div id="app">
3     
4     <HelloWorld msg="Welcome to Your Vue.js App"/>
```

```
src > components > HelloWorld.vue
You, 6 minutes ago | 1 author (
1 <template>
2   <div class="hello">
3     <h1>{{ msg }}</h1>
```

```
37 <script>
38 export default {
39   name: 'HelloWorld',
40   props: {
41     msg: String
42   },
```



5. 기능소개



5.3 data와 props (3/3)

- HelloWorld에서 msg를 렌더링한 모습



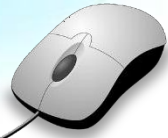
Welcome to Your Vue.js App

This is customMessage in data. Count: 0

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



5. 기능소개



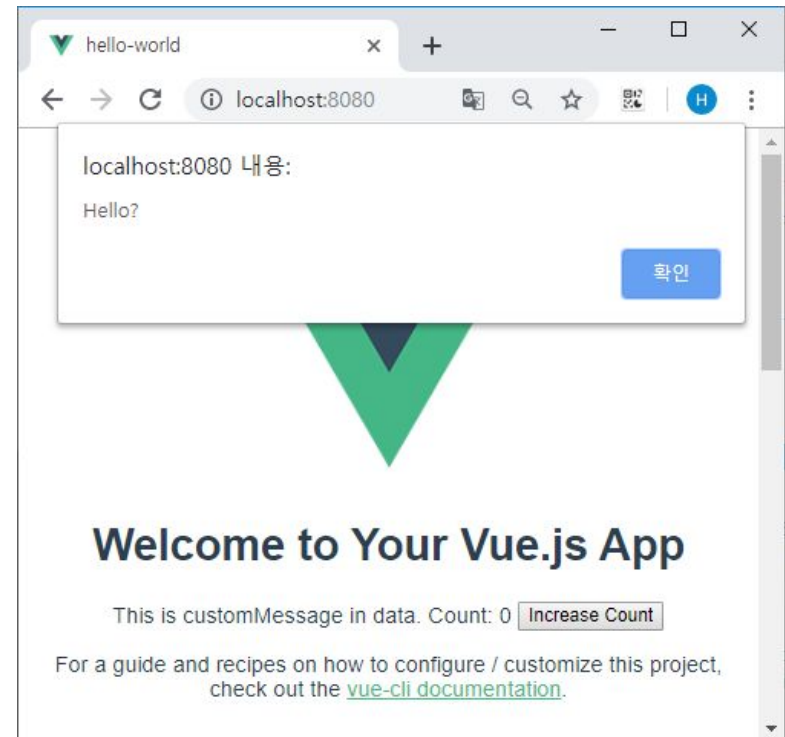
5.4 라이프사이클 훅 메소드

- Vue 인스턴스는 생성될 때 일련의 초기화 단계를 거치며, 단계마다 라이프사이클 훅이 호출됨
- 라이프사이클 훅에 유저가 로직을 정의해 넣을 수 있음
- 예를 들어, `mounted` 훅은 인스턴스가 마운트된 (렌더된) 후에 호출됨
- `mounted` 함수가 HelloWorld 컴포넌트 인스턴스가 마운트되고 실행된 모습

```
src > components > HelloWorld.vue
37 <script>
38   export default {
39     name: 'HelloWorld',
40     props: {
41       msg: String
42     },
43     mounted () {
44       window.alert('Hello?');
45     },
```

다양한 라이프사이클은 다음 참조 :

<https://vuejs.org/api/composition-api-lifecycle.html>



6. 활용예제



1. 버튼을 추가해서 이벤트 핸들링을 추가 (1/2)

- HelloWorld에 다음과 같이 버튼과 메소드를 추가
 - 템플릿에 버튼과 Count 변수를 나타내는 문장을 추가
 - <script>에 methods를 추가하고 그 안에 inc 함수를 추가
 - 주의: 화살표 함수 사용 금지
 - 화살표 함수를 사용하면 this.count에 접근 불가능 (this가 바인드되지 않음)

```
src > components > HelloWorld.vue
You, a few seconds ago | 1 author (You)
1 <template>
2   <div class="hello">
3     <p>
4       Count: {{ count }}
5       <button v-on:click="inc">Increase Count</button>
6     </p>
```

```
36 <script>
37 export default {
38   name: 'HelloWorld',
39   props: {
40     msg: String
41   },
42   mounted () {
43     window.alert('Hello?');
44   },
45   data: () => ({
46     customMessage: 'This is customMessage in data',
47     count: 0,
48   }),
49   methods: {
50     inc() {
51       this.count = this.count + 1;
52     }
53   },
54 }
55 </script>
```

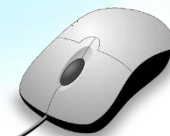
- 아래와 같이 페이지에 버튼과 Count가 추가됨
- 버튼을 누를때마다 카운트가 1씩 증가됨



Count: 0



6. 활용예제



1. 버튼을 추가해서 이벤트 핸들링을 추가 (2/2)

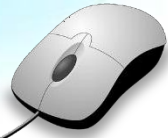
- button에 추가한 v-on:click="inc"으로 버튼을 클릭할 때 마다 inc 함수가 실행가능
 - v-on은 디렉티브로, DOM 이벤트를 듣고, 트리거 될 때 JavaScript를 실행 하도록 할 수 있음

```
src > components > HelloWorld.vue
You, a few seconds ago | 1 author (You)
1 <template>
2   <div class="hello">
3     <p>
4       Count: {{ count }}
5       <button v-on:click="inc">Increase Count</button>
6     </p>
```



- v-on 디렉티브는 버튼 뿐만 아니라 많은 DOM 오브젝트에 사용자가 적용 가능
 - 이를 통해 사용자가 페이지에 이벤트 리스너를 걸고 동적 페이지 조작이 가능





Q NPM을 Yarn 대신 사용해도 되나요?

&

A Vue CLI 가이드에선 NPM 또는 yarn을 사용하라고 안내합니다.

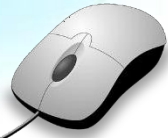
Yarn은 NPM에 비해 속도가 빠르고, 패키지를 verify하는 과정이 포함되어 있어 더 reliable 합니다. Yarn을 설치하고 싶지 않다면 NPM으로 진행해도 무방합니다.

Q CLI와 직접 `<script>` 추가하는 방식의 차이는 무엇인가요?

&

A CLI에는 프론트엔드 개발을 위한 빌드 설정을 같이 제공합니다. 핫 리로드, lint-on-save 및 프로덕션 빌드를 몇 분만에 할 수 있지만, CLI는 Node.js와 관련 빌드 도구에 대한 사전 지식이 필요합니다. `<script>` 방식은 빠르게 바로 개발을 시작할 수 있는 장점이 있습니다.





Q `<script>` 방식을 사용할 때 최소화 버전과 개발용 버전의 차이는 무엇인가요?

&

A 개발용 버전은 모든 경고와 디버그 모드를 포함하고, 배포용(최소화) 버전은 경고가 없는 대신 더 작은 크기로 압축되었습니다.

개발 중에는 일반적인 실수에 대해서 경고를 받을 수 있는 개발용 버전을 쓰는 것이 좋습니다.

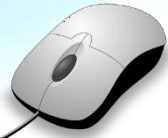
Q vue 3 버전이 나왔는데 기존 vue 2 버전과의 차이는 무엇인가요?

&

A vue 2 의 기능과 유사합니다. 다만 vue 3 버전에서는 vue 2에 없는 Teleport, Suspense 및 템플릿당 여러 루트 요소등이 포함되어 있습니다.

vue 3는 더 작은 번들 크기, 더 나은 성능, 더 나은 확장성 및 더 나은 TypeScript/IDE 지원을 제공합니다. 오늘 새 프로젝트를 시작하는 경우 Vue 3를 선택하는 것이 좋습니다.





Q Vue를 사용할 때 꼭 Typescript를 사용해야 하나요?

&

A Vue는 TypeScript로 구현이 되나 Vue API는 JavaScript와 TypeScript 모두에서 동일한 방식으로 작동하도록 설계되어 있습니다.

그러나, TypeScript를 사용하는 것은 장기적인 유지보수성에 있어 유리하기에 TypeScript를 사용하는 것을 권장합니다.



8. 용어정리



용어	설명
컴포넌트	컴포넌트는 Vue의 가장 강력한 기능 중 하나임 기본 HTML 엘리먼트를 확장하여 재사용 가능한 코드를 캡슐화하는 것에 도움이 됨 Vue 컴포넌트는 Vue 인스턴스이기도 함
CLI	커맨드 라인 인터페이스는 터미널을 통해 키보드로 컴퓨터와 상호 작용하는 인터페이스임 Vue CLI는 Vue 잘 설정된 프로젝트를 생성해주는 CLI임
화살표 함수	화살표 함수는 함수 선언식에 비해 짧은 구문으로 작성 가능하고, 익명 함수인 JavaScript 표현임 화살표 함수는 자신의 this를 바인딩하지 않으므로 Vue 앱의 일부에서는 사용하지 말아야 하는 곳이 있음



Open Source Software Installation & Application Guide



이 저작물은크리에이티브커먼즈[저작자표시- 비영리- 동일조건변경허락 2 . 0 대한민국라이선스]에 따라
이용하실 수 있습니다.