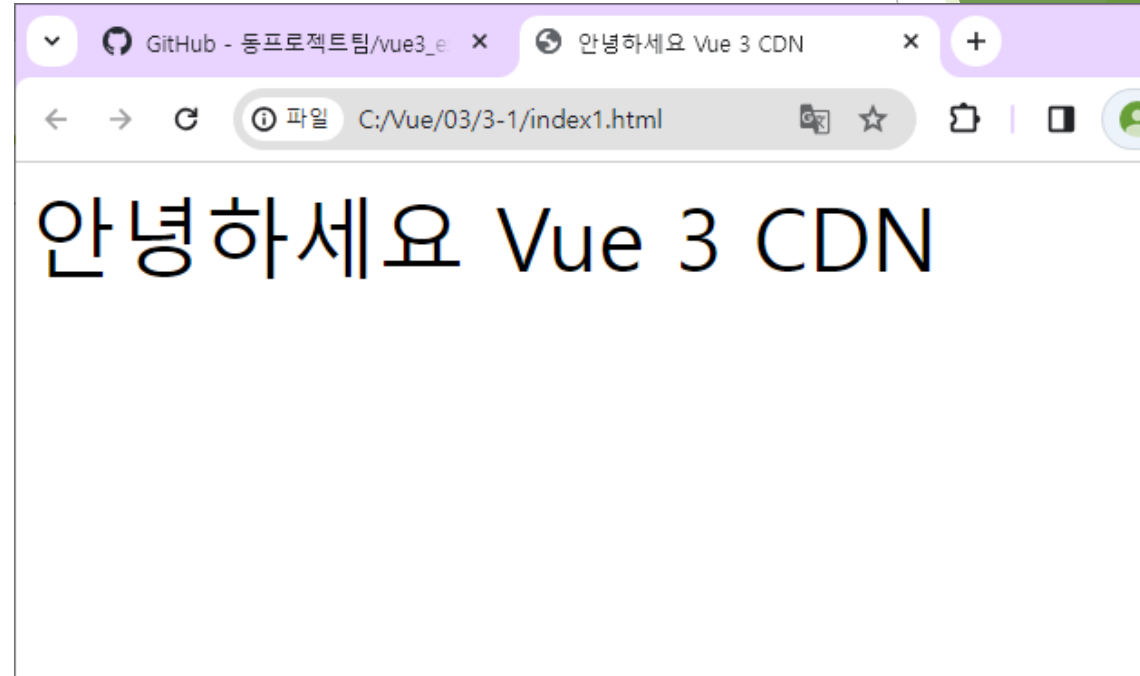


Vue 강의

CDN(Contents delivery Network)를 통한 Hello World 구현

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <title>Hello Vue 3 CDN</title>
5     <script src="https://unpkg.com/vue@next"></script>
6
7     <style>
8       .app {
9         font-size: 3em;
10      }
11    </style>
12  </head>
13  <body>
14    <div class="app">
15      {{ msg_hello }}
16    </div>
17
18    <script>
19      const App = {
20        data() {
21          return {
22            msg_hello: 'Hello Vue 3 CDN'
23          }
24        }
25      }
26
27      Vue.createApp(App).mount('.app')
28    </script>
29  </body>
30 </html>
31
```



- ▶ 스크립트 코드 내에서 **App** 객체를 생성 생성할 때 **msg_hello** 라는 변수를 선언하고 그 값으로 'Hello Vue 3 CDN'를 넣었음
- ▶ **Vue**에서 데이터 변수로 선언된 값은 템플릿 내 **HTML** 코드에서 두 개의 중괄호 (**{{ , }}**)로 접근이 가능한데, 이른 선언적 렌더링이라 부름
- ▶ 이는 **jQuery**와 같은 명령형 렌더링과 상반되는 개념으로 **DOM** 엘리먼트에서 다시 렌더링할 것을 명령하지 않고 **DOM**과 연결된 상태와 속성이 변경될 때 자동으로 **DOM** 엘리먼트가 업데이트 되는 것을 말함
- ▶ 대표적으로 **Vue**와 **React**가 선언적 렌더링을 사용함

뷰 CLI로 프로젝트 생성하기

- ▶ Vue CLI는 Vue 프로젝트를 빠르게 구성하고, 빌드, 디플로이 할 수 있게 도와주는 도구임
- ▶ Vue CLI 설치

```
C:\Windows\system32\cmd.exe
```

```
C:\WVue_ex>npm install -g @vue/cli
```

-g : 해당 패키지가 현재 디렉토리뿐만 아니라 앞으로 생성되는 모든 프로젝트에서 사용할 수 있는 **global** 패키지 등록

- ▶ Default 옵션으로 프로젝트 설치
 - vue 프로젝트 생성

vue create [프로젝트 명] 명령어를 실행합니다. 여기서 프로젝트 명은 'vue-project'입니다.

```
$ vue create vue-project
```

아래 그림과 같이 Default (Vue 3) 옵션을 선택해 줍니다.

```
Vue CLI v4.5.13
? Please pick a preset:
  Default ([Vue 2] babel, eslint)
> Default (Vue 3) ([Vue 3] babel, eslint)
  Manually select features
```

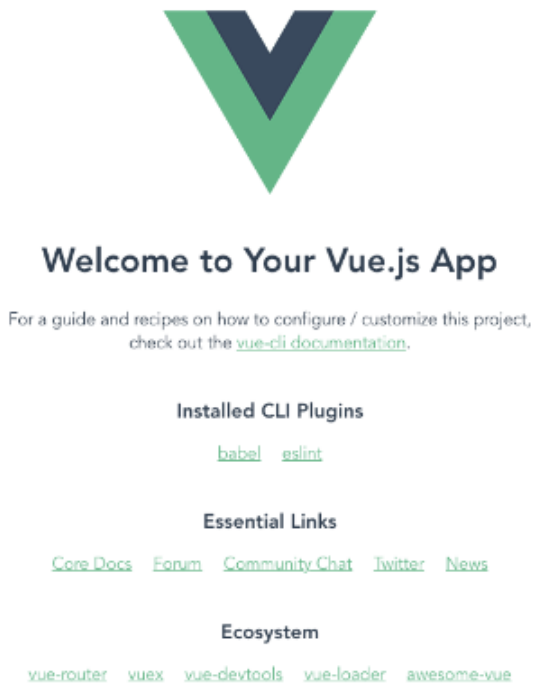
▶ Vue 프로젝트 실행

해당 프로젝트로 이동 후, `npm run serve` 명령어로 프로젝트를 실행해 줍니다.

- 참고: 8080 아닌 다른 포트 사용하는 경우 `npm run serve -- --port 3000` 명령어를 실행합니다. 그러면 3000 포트로 실행 가능합니다.

```
$ cd vue-project  
$ npm run serve
```

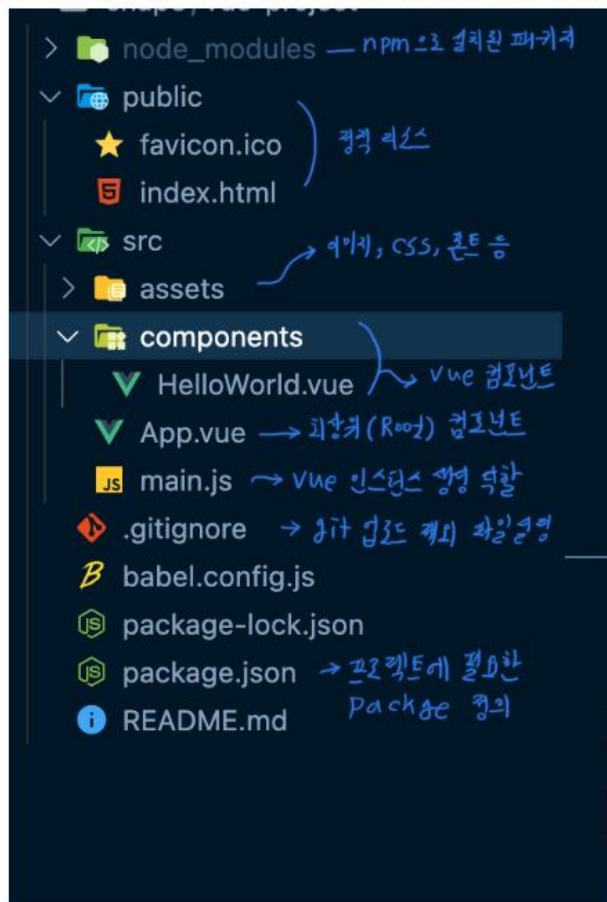
<http://localhost:8080/> 에 접속을 하여 아래 그림과 같이 웹 사이트 결과를 확인해줍니다.



VS 코드에서 생성된 프로젝트를 Open

2-3. vue 프로젝트 파일 구조

설치된 프로젝트 파일 구조는 아래와 같고 파일 및 폴더에 대한 의미는 아래 그림과 같습니다.



아래는 package.json 관련 파일에 대한 내용입니다.

```
[
  "name": "vue-project",
  "version": "0.1.0",
  "private": true,
  "scripts": {
    "serve": "vue-cli-service serve",
    "build": "vue-cli-service build",
    "lint": "vue-cli-service lint"
  },
  "dependencies": {
    "core-js": "^3.6.5",
    "vue": "^3.0.0"
  },
  "devDependencies": {
    "@vue/cli-plugin-babel": "~4.5.0",
    "@vue/cli-plugin-eslint": "~4.5.0",
    "@vue/cli-service": "~4.5.0",
    "@vue/compiler-sfc": "^3.0.0",
    "babel-eslint": "^10.1.0",
    "eslint": "^6.7.2",
    "eslint-plugin-vue": "^7.0.0"
  },
  "eslintConfig": {
    "root": true,
    "env": {
      "node": true
    },
    "extends": [
      "plugin:vue/vue3-essential",
      "eslint:recommended"
    ],
    "parserOptions": {
      "parser": "babel-eslint"
    },
    "rules": {}
  },
  "browserslist": [
    "> 1%",
    "last 2 versions",
    "not dead"
  ]
]
```

→ 프로젝트 명

npm run ~

사용 중인 패키지

개발 시에만 필요한 패키지

eslint : 일관성 있는 코드 작성 & 버그 회피 목적

→ 기존 분위기를 기함.

Vue Router 설치

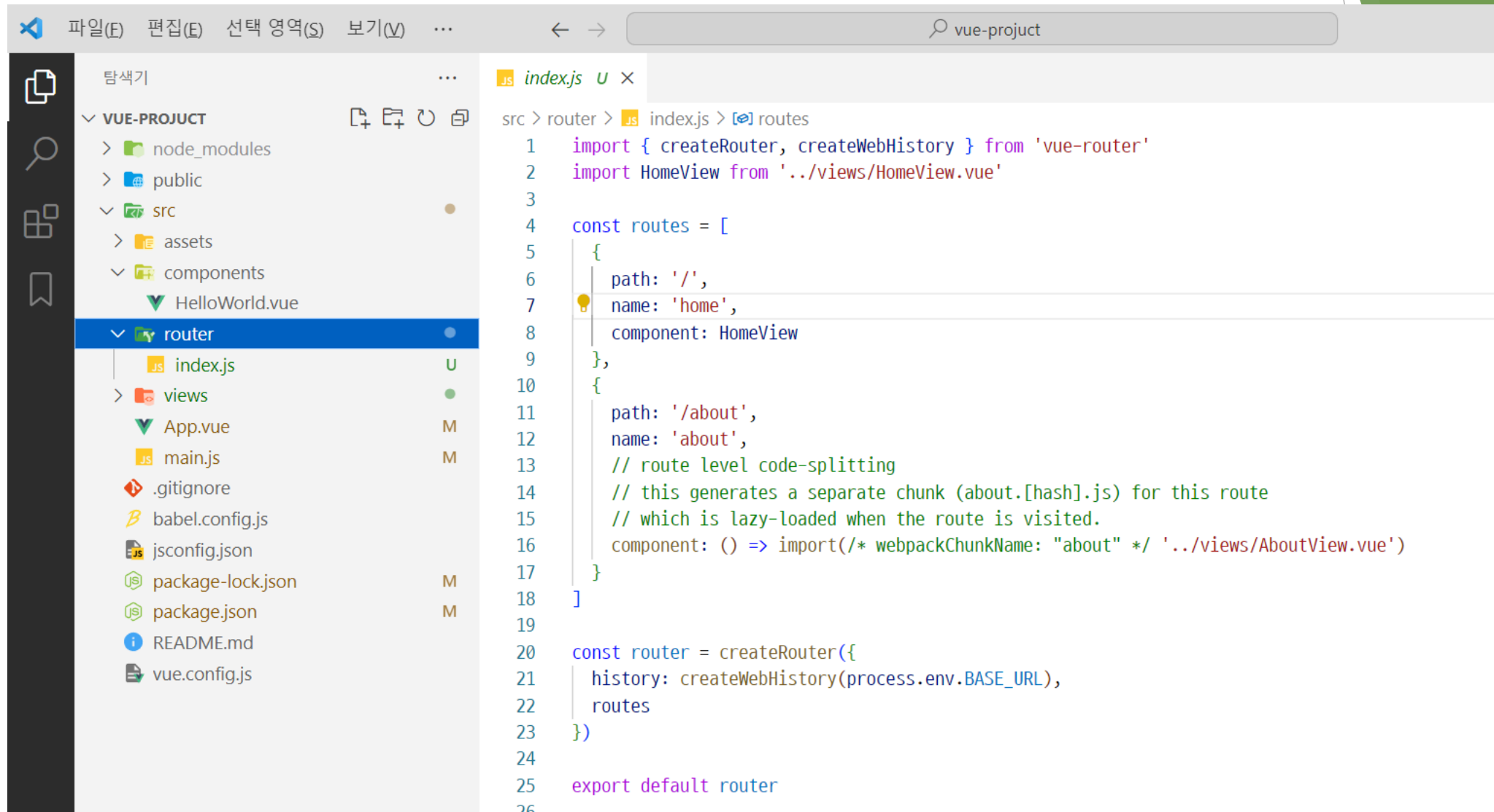
- ▶ Vue 내에서 페이지 이동을 위해 라우터를 설치함
- ▶ cmd 창에서 **Ctrl + C** 를 눌러 서버를 종료함
- ▶ 그리고 vue 프로젝트 폴더에서 **vue add router**를 입력하여 라우터를 설치해 줌

```
일괄 작업을 끝내시겠습니까 (Y/N)?  
^C  
C:\W\ue_ex\vue-project>vue add router  
  
❖ Installing @vue/cli-plugin-router...
```

- ▶ 이때 히스토리 모드 사용할거냐고 물어보는데 **Y** 를 선택함

```
Run `npm audit` for details.  
✓ Successfully installed plugin: @vue/cli-plugin-router  
? Use history mode for router? (Requires proper server setup for index fallback in production) (Y/n)
```


- ▶ 설치가 완료되면 다음과 같이 **router** 폴더와 **index.js** 파일이 생성 됨

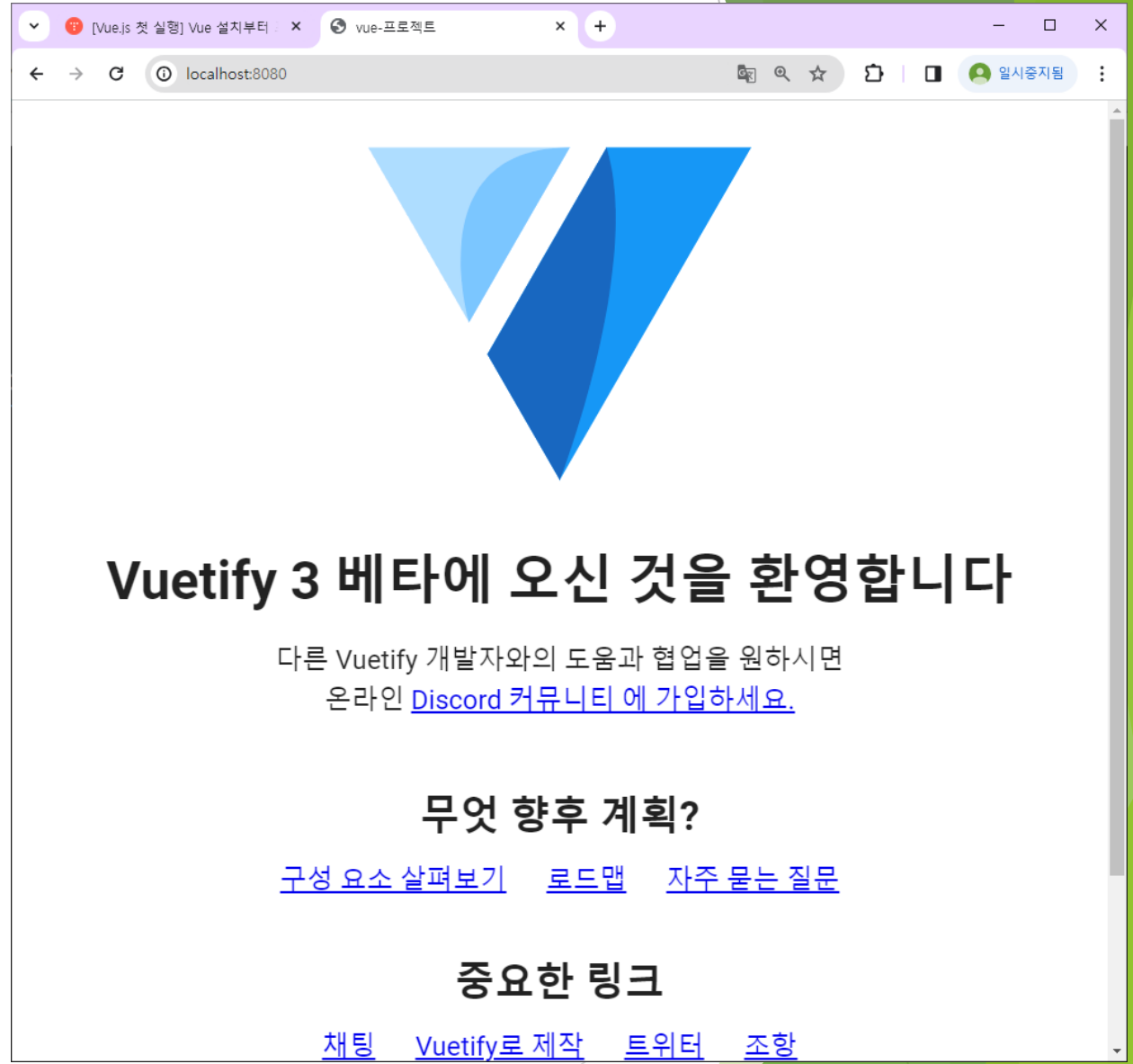


The screenshot shows the Visual Studio Code interface. On the left, the 'VUE-PROJECT' file explorer is expanded, showing the 'router' folder. The 'index.js' file is selected. The main editor displays the content of 'src > router > index.js', which defines the routes for the application.

```
src > router > index.js > routes
1  import { createRouter, createWebHistory } from 'vue-router'
2  import HomeView from '../views/HomeView.vue'
3
4  const routes = [
5    {
6      path: '/',
7      name: 'home',
8      component: HomeView
9    },
10   {
11     path: '/about',
12     name: 'about',
13     // route level code-splitting
14     // this generates a separate chunk (about.[hash].js) for this route
15     // which is lazy-loaded when the route is visited.
16     component: () => import(/* webpackChunkName: "about" */ '../views/AboutView.vue')
17   }
18 ]
19
20 const router = createRouter({
21   history: createWebHistory(process.env.BASE_URL),
22   routes
23 })
24
25 export default router
26
```

Vuetify 설치

- ▶ Bootstrap 등 여러 템플릿 중에서 Vuetify 설치
- ▶ Vue3 버전은 Vuetify3 버전을 선택함
- ▶ 프로젝트 폴더에서 **vue add vuetify** 를 입력함
- ▶ `npm run serve` 실행시 우측 화면처럼 보여짐



Vue.js(‘뷰’)란?

- ▶ 웹 페이지 화면을 개발하기 위한 프론트엔드 프레임워크
- ▶ 뷰는 화면단 라이브러리아자 프레임워크
- ▶ 뷰 코어 라이브러리는 화면단 데이터 표현에 관한 기능들을 중점적으로

Vue.js(‘뷰’)란?

- ▶ 웹 페이지 화면을 개발하기 위한 프론트엔드 프레임워크
- ▶ 뷰는 화면단 라이브러리아자 프레임워크
- ▶ 뷰 코어 라이브러리는 화면단 데이터 표현에 관한 기능들을 중점적으로

Vue.js(‘뷰’)란?

- ▶ 웹 페이지 화면을 개발하기 위한 프론트엔드 프레임워크
- ▶ 뷰는 화면단 라이브러리아자 프레임워크
- ▶ 뷰 코어 라이브러리는 화면단 데이터 표현에 관한 기능들을 중점적으로