

프론트엔드 개발자를 위한 Vue.js 시작하기

# Vue.js </> 시작하기

## Component 기본



한국기술교육대학교  
온라인평생교육원

## 학습내용

- Component 선언
- Component 이용

## 학습목표

- Component를 선언하는 방법에 대해 설명할 수 있다.
- Component의 Global 등록 및 Local 등록 방법에 대해 설명할 수 있다.

# Component 선언

## Component 선언

### 1 Component



#### Component 특징

- Vue의 핵심 구성요소
- UI를 위한 화면(DOM)과 로직(Javascript) 결합



Component



The Progressive  
JavaScript Framework

WHY VUE.JS?

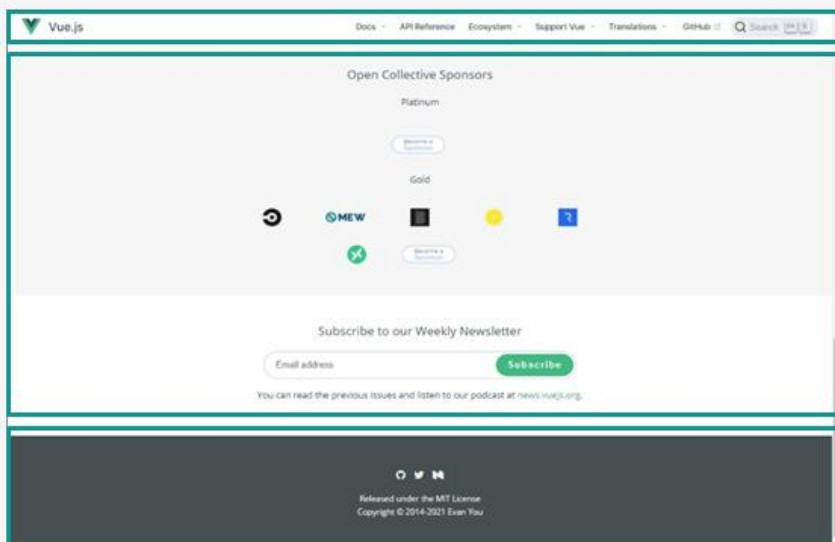
GET STARTED

GITHUB

## Component 선언

### 1 Component

- 재사용 가능한 동적 UI 구성을 목적으로 함



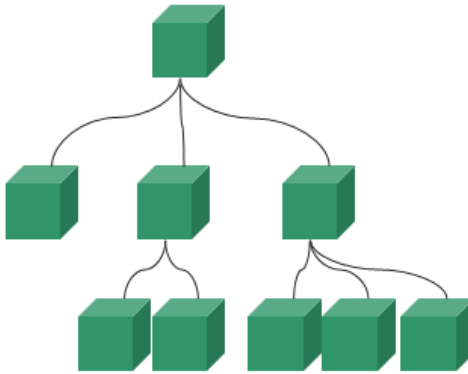
# Component 선언

## Component 선언

### 1 Component

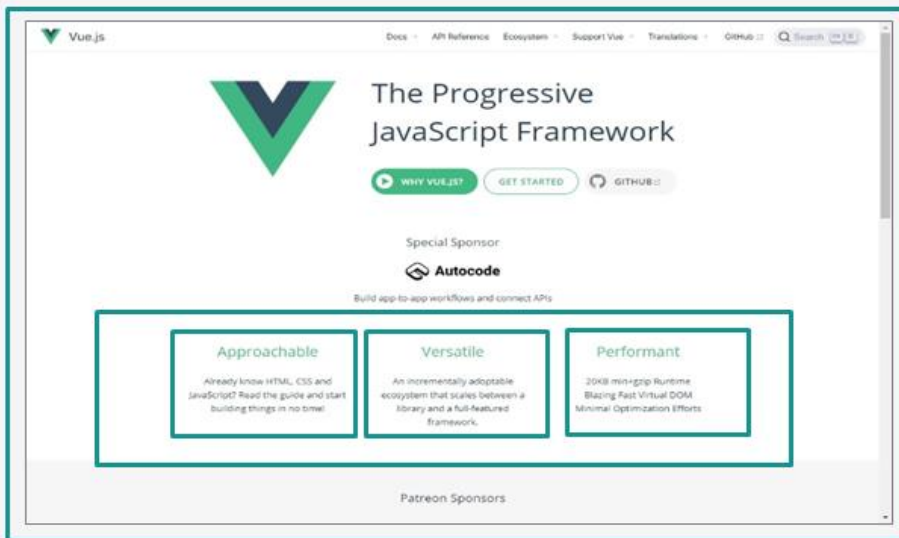
#### Component Tree 구조

→ 의미 단위로 Component를 구성하고  
구성된 Component를 조합해서 화면을 구성함



## Component 선언

### 1 Component



# Component 선언

## Component 선언

### 2 Object Literal로 Component 선언

- Template, Method, Data 등의 다양한 Option을 가지는 **Object literal**로 선언함
- Component의 Data는 **Data(){} 스타일**의 함수로 선언
- Component의 화면은 **Template 속성**으로 정의함

## Component 선언

### 2 Object Literal로 Component 선언

- Component의 Data를 Template에서 이용할 때는 **{{ }} 표현식** 이용

```
var myComponent = {
  data() {
    return {
      name: 'Hong Gildong'
    }
  },
  template: `
    <h3>Component Test {{name}}
  `
}
```

# Component 선언

## Component 선언

### 3 Application Instance를 이용해 Component 출력

- createApp() 함수로 Application Instance 생성
- 매개변수에 Root Component 등록

## Component 선언

### 3 Application Instance를 이용해 Component 출력

- mount() 함수로 화면 출력

```
var app = createApp(myComponent)
app.mount('#app')
```

# Component 선언

## Component 선언

### 4 Vue 파일로 Component 선언

- 확장자가 Vue인 파일

**<template>, <script>, <style> 태그로 Component 구성**

## Component 선언

### 4 Vue 파일로 Component 선언

```
<template>
  <div>
    Hello,
    {{ name }}
  </div>
</template>
<script>
export default {
  data(){
    return {
      name: ' vue...'
    }
  }
};
</script>
```

# Component 이용

## Component 이용

### 1 Global 등록 사용



Application Instance에 Component 등록



Application 전역에서 Component 이용



Application Instance의 component()함수로 등록

## Component 이용

### 1 Global 등록 사용

- 첫 번째 매개변수가 Component 태그명임

```
var app = createApp(Super)
app.component('Sub', Sub)
```



# Component 이용

## Component 이용

### 1 Global 등록 사용

#### ● 등록된 Component 이용

```
<template>
  <div>
    I am super component<br/>
    <Sub/>
  </div>
</template>
<script>
export default {
};
</script>
```

## Component 이용

### 2 Local 등록 사용



Component가 필요한 곳에서 등록 사용



components 속성을 이용해 등록

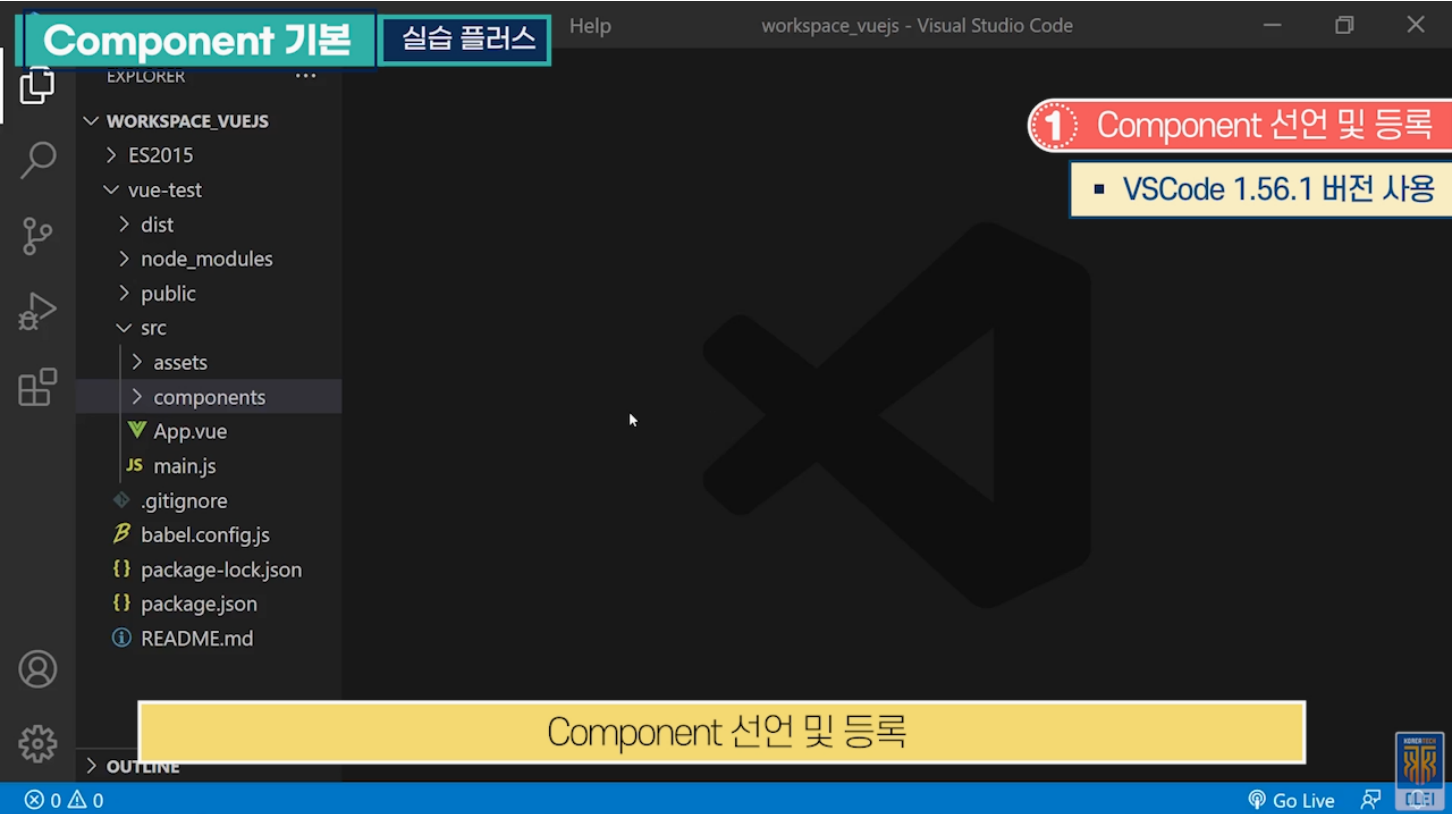
# Component 이용

## Component 이용

### 2 Local 등록 사용

```
<template>
  <div>
    I am super component<br/>
    <MySub/>
  </div>
</template>
<script>
import MySub from "./Sub.vue";
export default {
  components: {
    MySub
  }
};
</script>
```

# 실습 플러스



실습단계
Vue 프로젝트에 Component 생성 실습
src → ch7 서브 폴더 생성
파일명을 test1.js로 해서 자바스크립트 파일 생성
Component 정의
변수 값을 출력할 템플릿 정의
외부 파일에서 사용이 가능하도록 만들어진 Component를 export
해당 Component를 Vue 프로젝트에 적용시키기 위해 main.js 파일 생성
Vue 라이브러리와 작성한 js 파일을 import
Vue의 App 객체를 create
프로젝트를 화면에 출력하기 위한 선언
동일 폴더에 HTML 파일 생성, index.html
body 부분에 프로젝트 출력에 필요한 Tag만 명시
테스트 편의성, 개발자 임의의 폴더만 별도 테스트 진행
ch7 폴더에 Vue 설정을 위한 js 파일 생성, vue.config.js

# 실습 플러스

실습단계
ch7 폴더만 독립적으로 Vue CLI 명령으로 테스트 진행 → 테스트 편의성
Npm 명령어로 테스트를 위한 모듈 설치
Vue CLI 명령으로 실행
브라우저에 Component 출력
Component를 자바스크립트로 정의함
효율적인 방법은 확장자가 vue인 파일로 Component를 정의하는 것!
VS Code가 확장자가 vue인 파일을 인지하지 못함!
VS Code 플러그인 설치, Extensions 클릭
vetur 검색
Vue를 VS Code에서 인지할 수 있도록 플러그인 vetur 설치
Vue 파일 생성 가능!
Vue 파일에 Component 정의
스크립트 Tag 영역에 Vue Component 정의
변수 값이 템플릿에 출력되어서 템플릿 내의 내용이 화면에 출력되는 것
main.js 파일에 적용
저장을 하면 Hot Deploy에 의해 결과가 바로 출력됨
브라우저에 Vue 파일로 작성한 Component 출력