

Doit! Vue.js

2019.03.19 이경임 금융서비스부 매체서비스팀

INDEX

주제	세부토픽
Vue.js 알아보기	<ol style="list-style-type: none">1. 웹의 발전과 Vue2. Vue.js 알아보기3. Angular vs React vs Vue
Vue.js 기술적 특징	<ol style="list-style-type: none">1. 뷰 인스턴스 & 컴포넌트2. 뷰 라우터3. 뷰 HTTP 통신4. 뷰 템플릿
Vue.js로 화면개발하기	<ol style="list-style-type: none">1. Vue 프로젝트 구성방법2. Vue.js로 구현한 할일관리앱

01

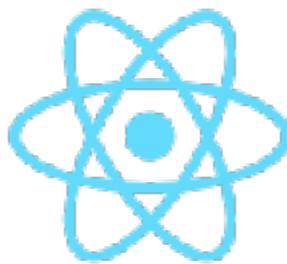
— Vue.js 알아보기

Subjects

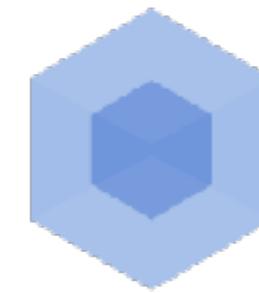
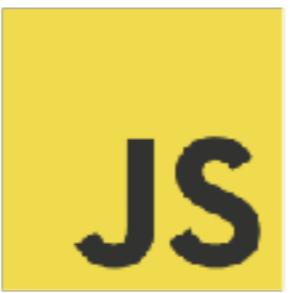
1. 웹의 발전과 Vue
2. Vue.js 알아보기
3. Angular vs React vs Vue

01

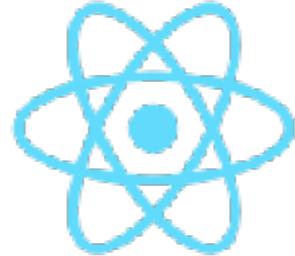
웹의 발전과 Vue



BABEL



BABEL



Vue.js의 비전



*I envision Vue's goal to be helping
more developers enjoy building
apps on the Web.*

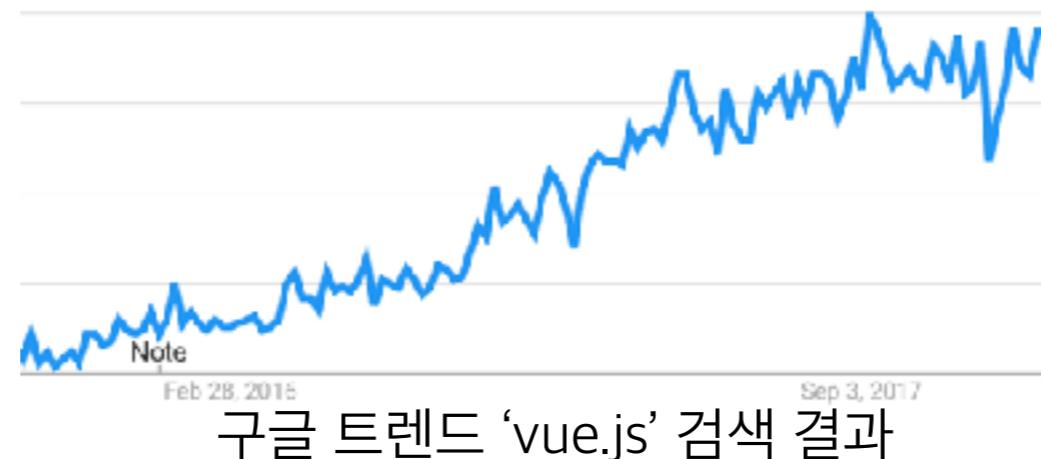


Evan You
Creator of Vue.js

“Vue.js의 초점은 더 많은 사람들이
쉽게 웹 앱을 개발하도록 도와주는데 있다.”

-에반 유(Vue.js 창시자)

통계 Vue에 대한 관심



	#1 Vue.js	+26.4k★
<i>A progressive, incrementally-adoptable JavaScript framework for... </i>		
	#2 React	+22.9k★
<i>A declarative, efficient, and flexible JavaScript library for building ... </i>		
	#3 Yarn	+21.0k★
<i>Fast, reliable, and secure dependency management. </i>		

2016년 1위

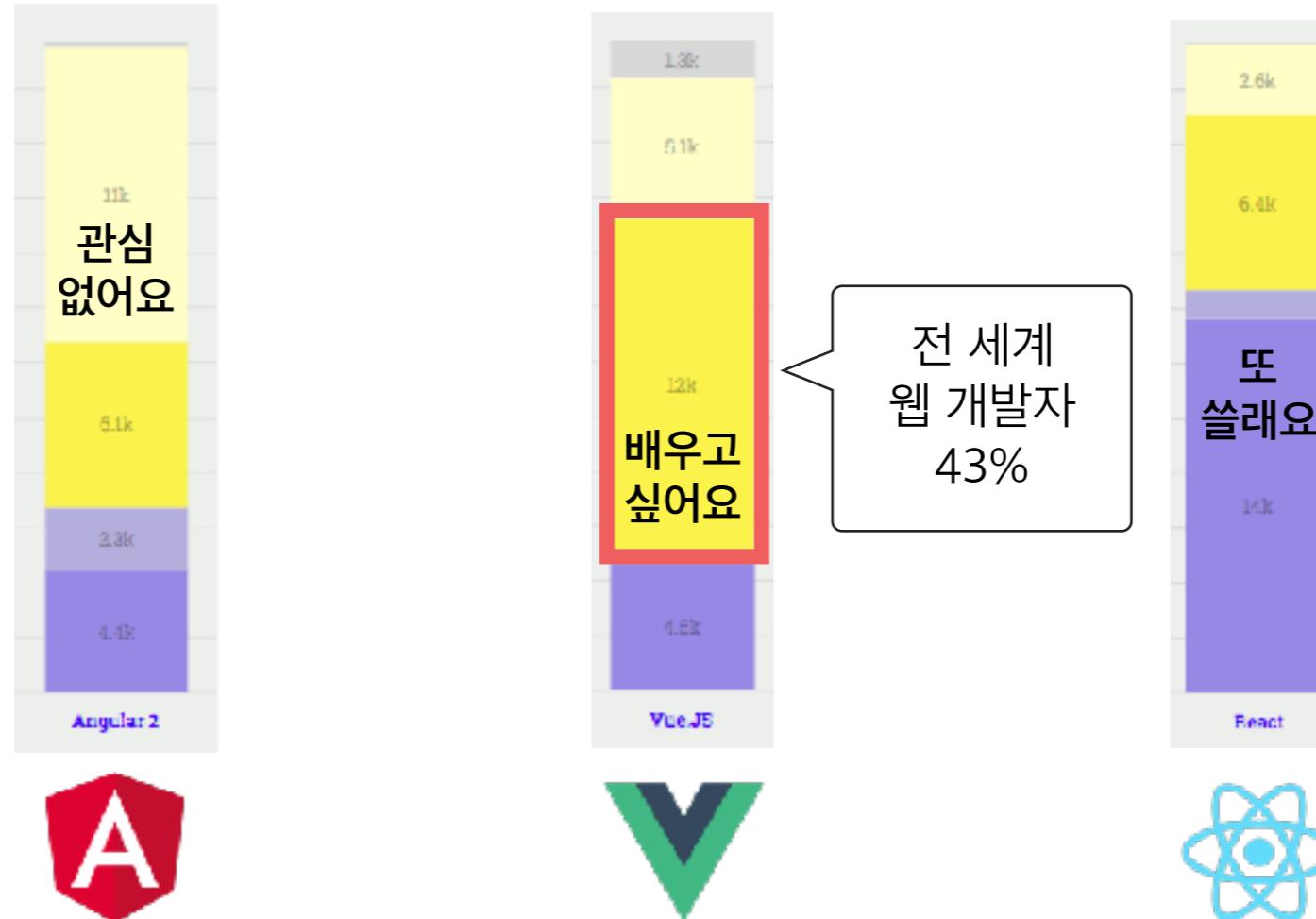
1	 Vue.js A progressive, incrementally-adoptable JavaScript framework...	+40.0k★
2	 React A declarative, efficient, and flexible JavaScript library for buildi...	+27.8k★
3	 Create React App Create React apps with no build configuration.	+22.5k★
4	 Puppeteer Headless Chrome Node API	+22.0k★
5	 Axios Promise-based HTTP client for the browser and node.js	+21.9k★

2017년 1위

구글 트렌드 & 깃헙

출처: <https://risingstars.js.org/> 7

통계 Vue에 대한 관심



2017년 전세계 웹 개발자 설문조사

출처: <https://stateofjs.com/2017/front-end/results> 8

통계 Vue에 대한 관심

96%

survey participants would use Vue.js again for their next project.

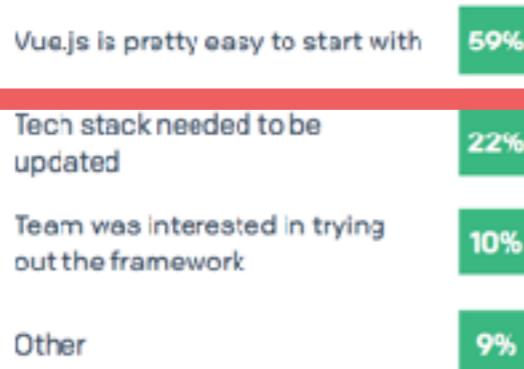
다음 프로젝트도 Vue.js를 쓸거예요

81%

of survey participants say that ease of integration is one of the chief advantages of having Vue in their organizations' tech stack.

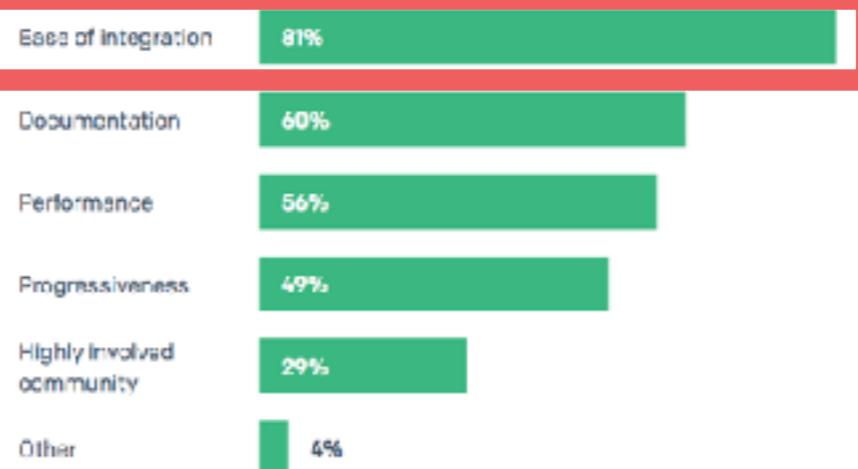
기존 기술 스택과 함께 사용하기 쉬워요

THE MOST IMPORTANT REASON BEHIND ADDING VUE TO THE TECH STACK



THE BIGGEST ADVANTAGE

Percentages do not sum up to 100% due to the multiple choices



State of Vue.js 2017

88개국. 개발자 1126명의 설문 결과

9

출처: State of Vue.js 2017 Report

사용성 얼마나 편리한가

jQuery

```
<body>
  <div>
    <p></p>
    <button>click me</button>
  </div>

  <script
    src="http://code.jquery.com/jquery-3.3.1.js"></script>
  <script>
    $('p').text('Hello World');
    $('button').click(function() {
      alert('hi');
    });
  </script>
</body>
```

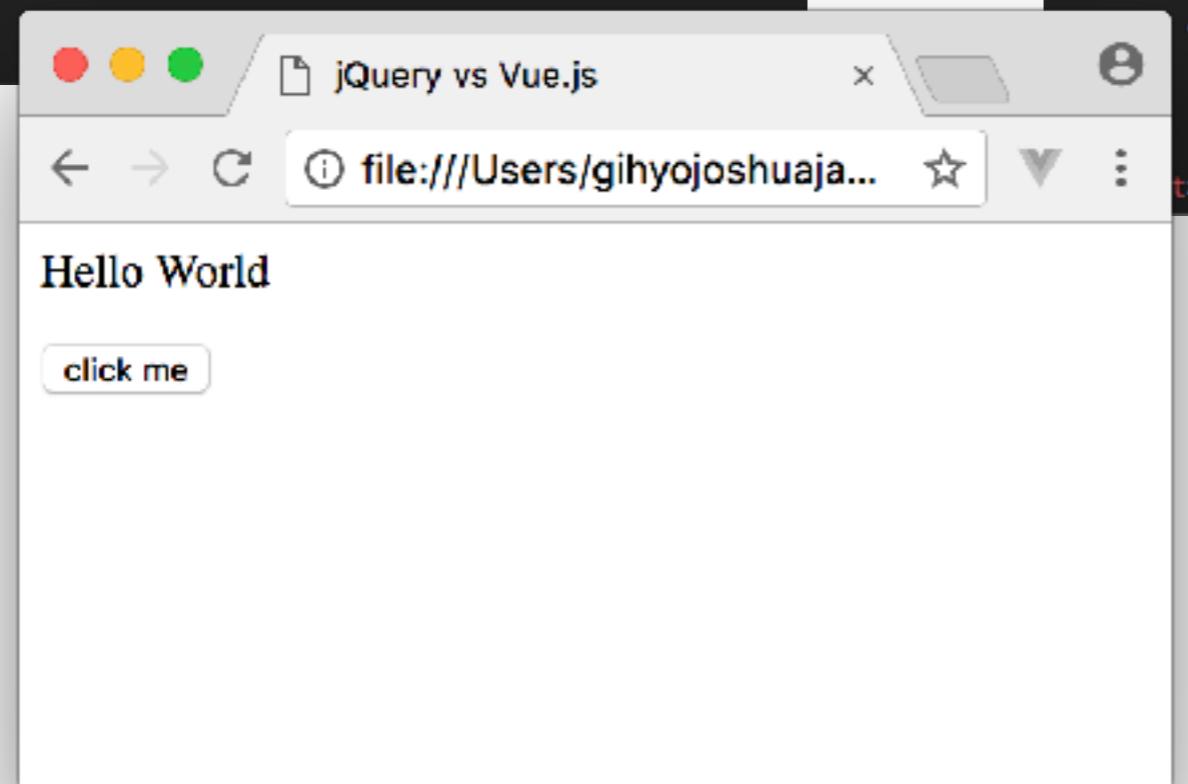
CDN 라이브러리 로딩

Vue.js

```
<div id="app">
  <p>{{ message }}</p>
  <button v-on:click="clickBtn">click me</button>
</div>

<script src="https://cdn.jsdelivr.net/npm/vue"></script>
<script>
  new Vue({
    el: '#app',
    data: {
      message : 'Hello World'
    },
    methods: {
      clickBtn: function() {
        alert('hi');
      }
    }
  })
</script>
```

CDN 라이브러리 로딩



사용성 얼마나 편리한가

QuickStart



Hello World



The easiest way to get started with React is to use [this Fiddle](#) or [this Pen](#) on [CodePen](#). You don't need to install anything; you can just open them in your browser and follow along as we go through examples. If you'd rather set up a development environment, check out the [Installation](#) section.

The smallest React example looks like this:

```
ReactDOM.render(  
  <h1>Hello, world!</h1>,  
  document.getElementById('root')  
)
```

It renders a header saying "Hello, world!" on the page.

The next few sections will gradually introduce you to using React. We will start by introducing the basic building blocks of React apps: elements and components. Once you master these concepts, you'll learn how to use them to create complex apps from small reusable pieces.

Getting Started



! The official guide assumes intermediate level knowledge of HTML, CSS, and JavaScript. If you are totally new to frontend development, it might not be the best idea to jump right into a framework as your first step - grasp the basics then come back! Prior experience with other frameworks helps, but is not required.

The easiest way to try out Vue.js is using the [JSFiddle Hello World example](#). Feel free to open it in another tab and follow along as we go through some basic examples. Or, you can [create an index.html file](#) and include Vue with:

```
<script src="https://cdn.jsdelivr.net/npm/vue"></script>
```

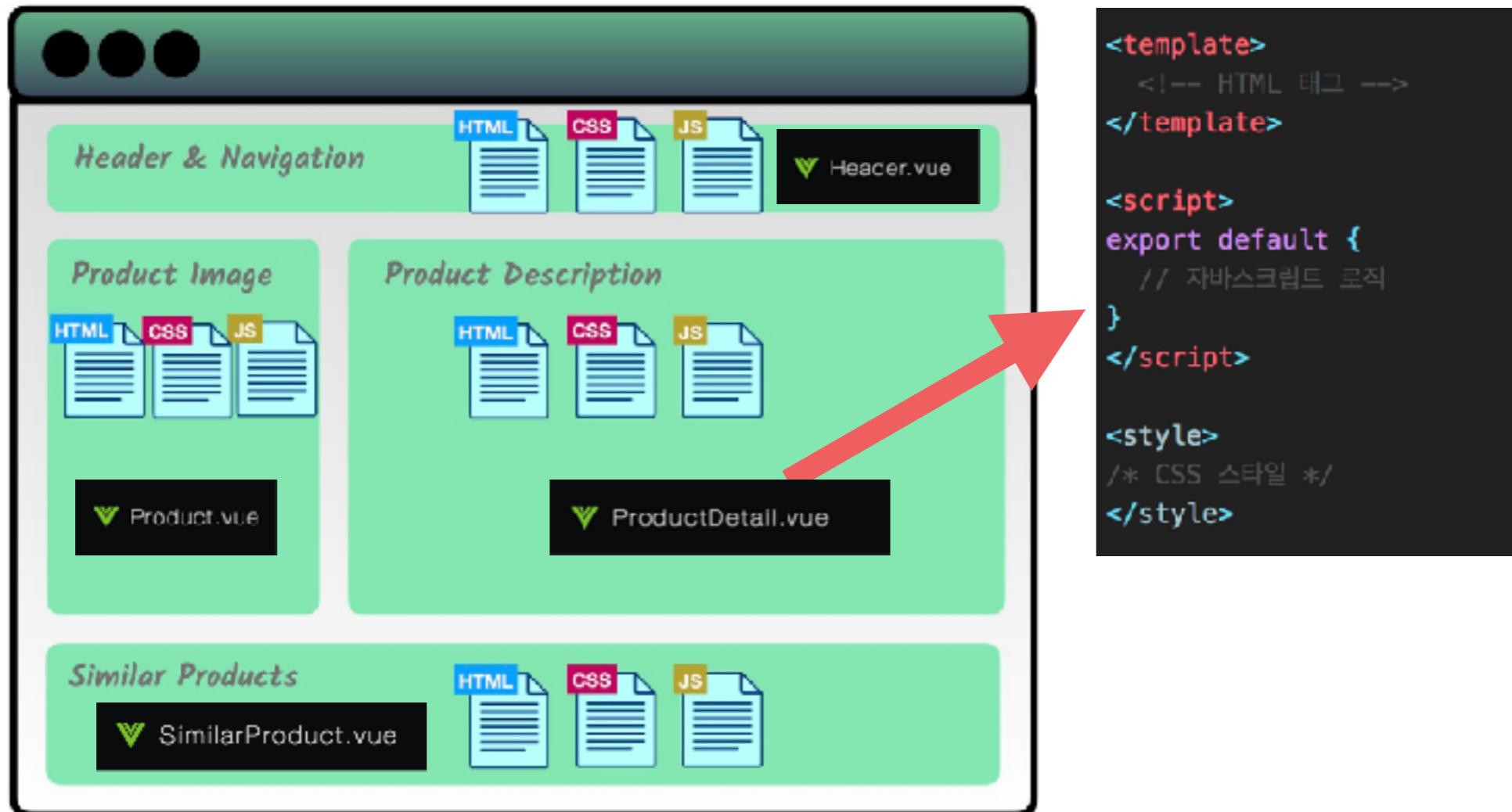
HTML

The [Installation](#) page provides more options of installing Vue. Note: We do not recommend that beginners start with [vue-cli](#), especially if you are not yet familiar with Node.js-based build tools.

입문자를 배려한 가이드 문서

11

사용성 얼마나 편리한가

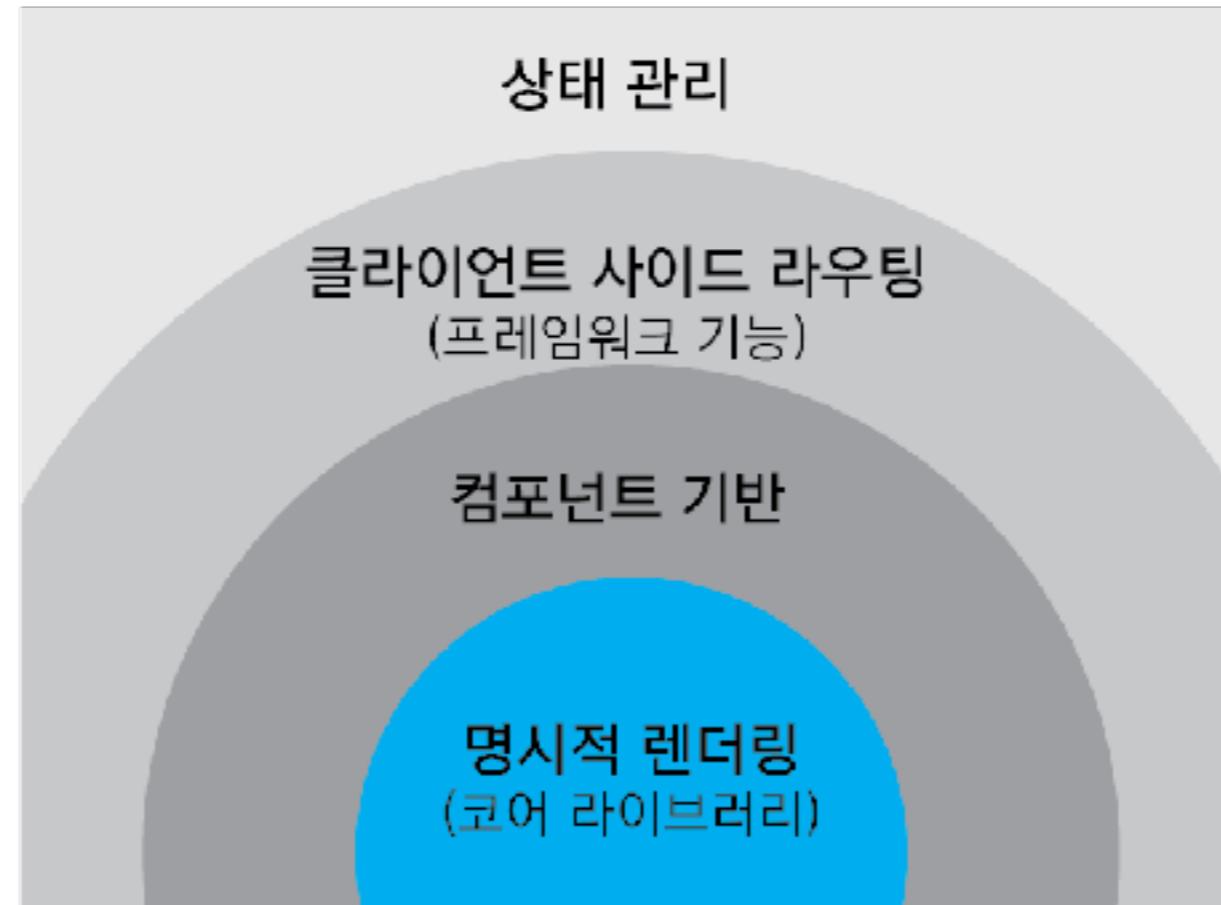


컴포넌트별 코드 관리

02

Vue.js 알아보기

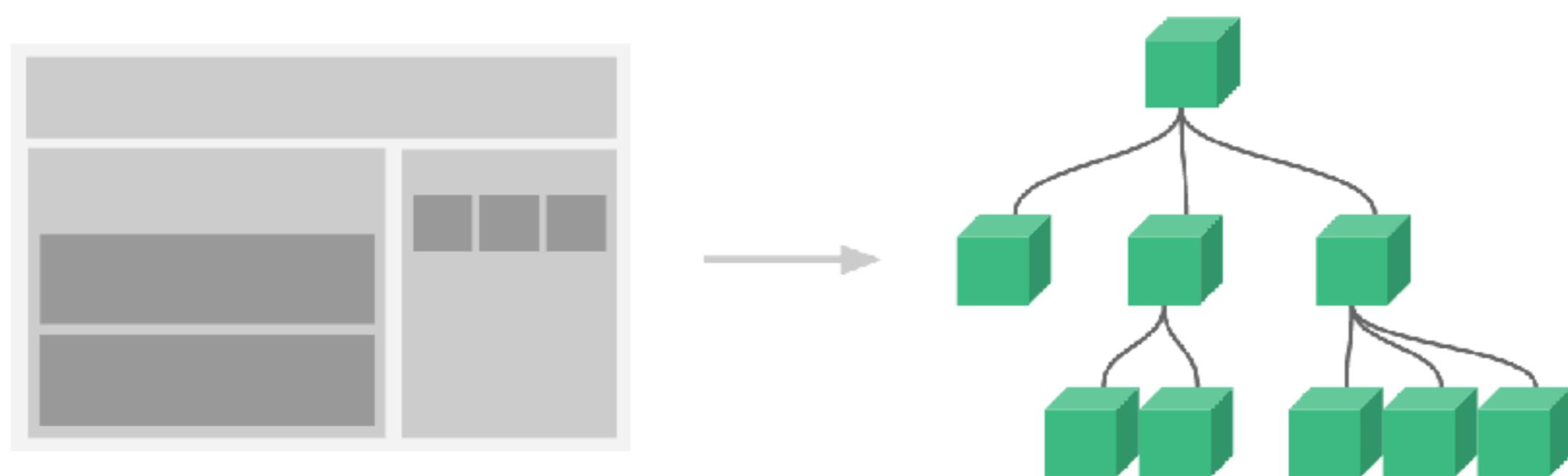
Vue.js란?



작은 화면단 라이브러리 역할부터
큰 규모의 웹 애플리케이션 개발을 돋는 프레임워크 역할까지
점진적으로 적용할 수 있는 프런트엔드 프레임워크

Vue.js 프레임워크 특징 3가지

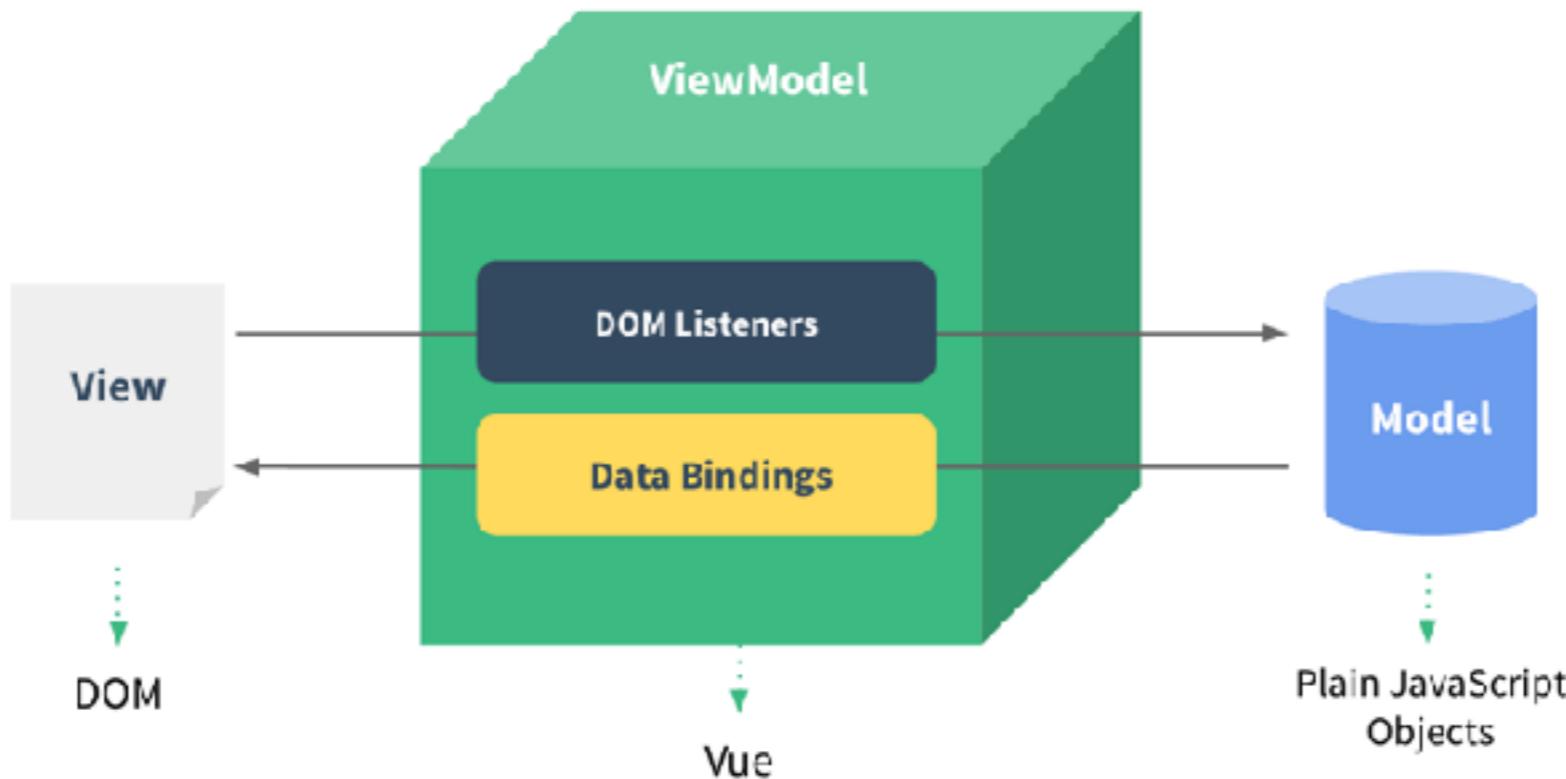
1. 컴포넌트 기반 개발 방식



화면을 여러 개의 작은 단위로 쪼개어 개발

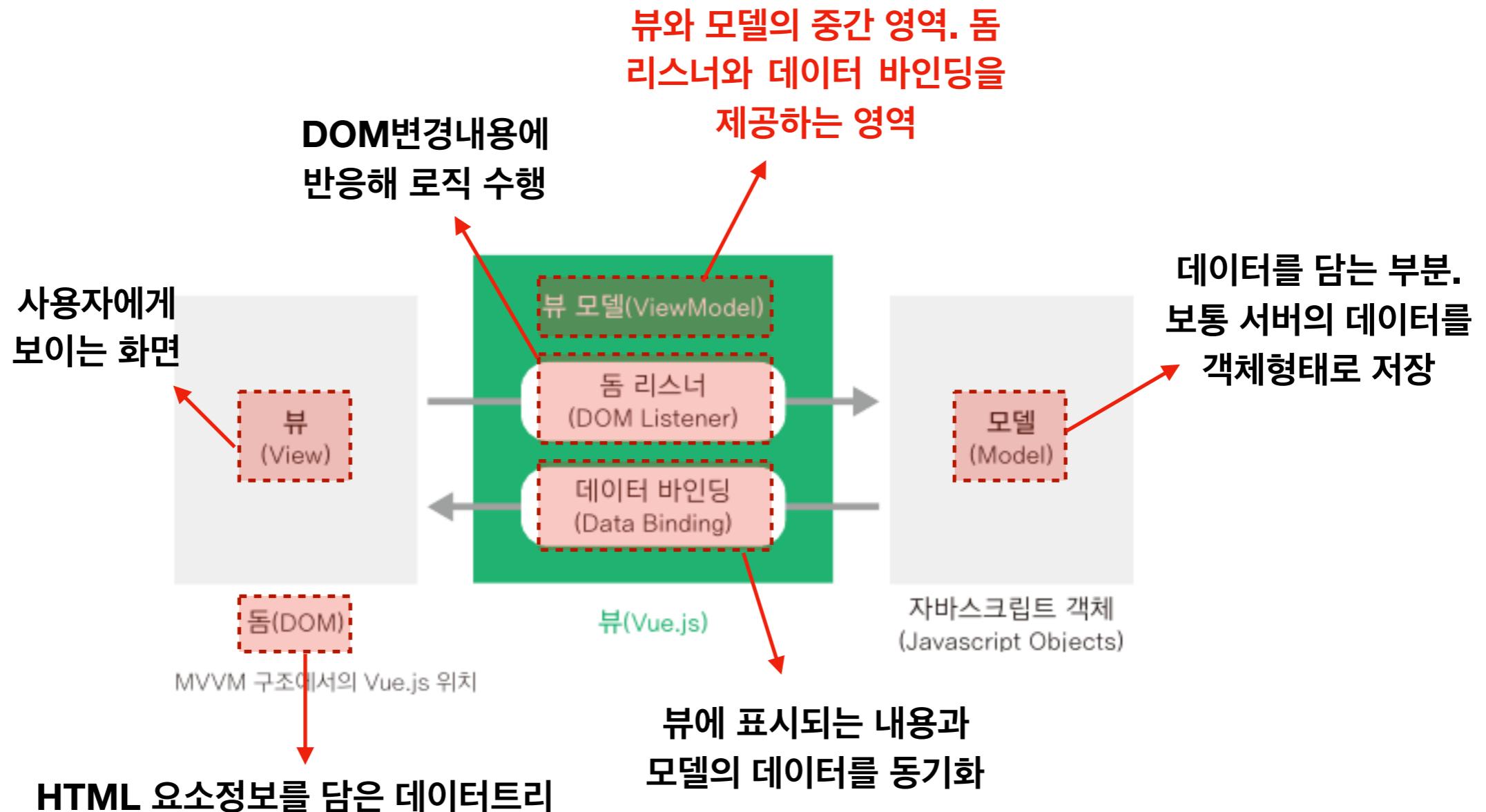
재사용성 ↑ 구현속도 ↑ 확인! 코드 가독성 ↑

2. MVVM 패턴

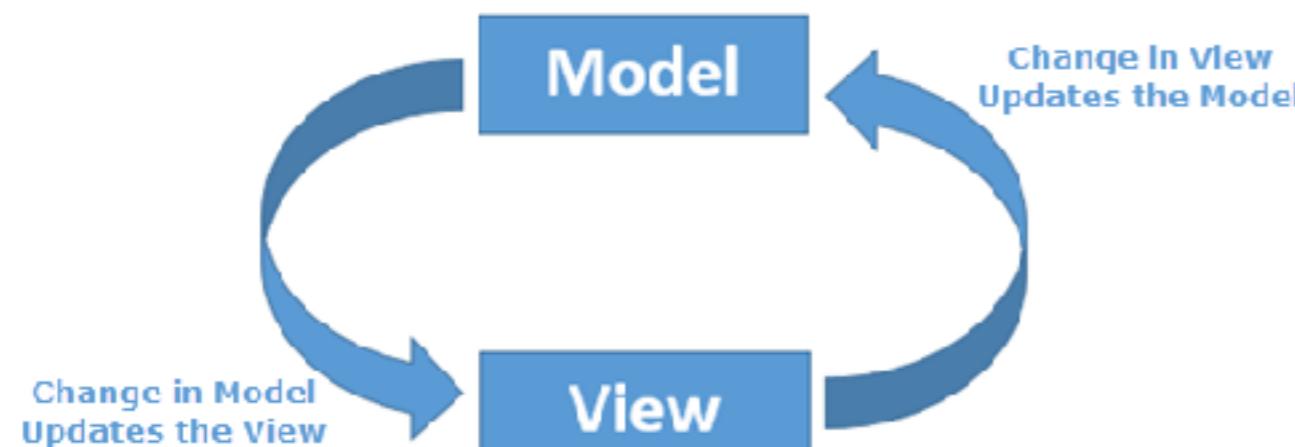


화면 UI 코드와 백엔드 데이터 처리 코드를 분리

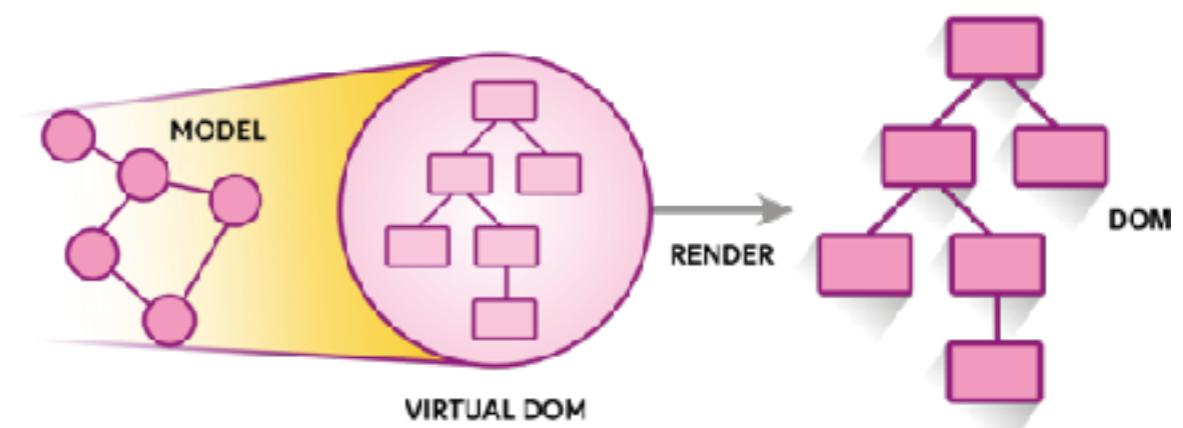
2. MVVM 패턴



3. React와 Angular의 장점을 흡수



Two-way data binding



Virtual DOM

03

Angular vs Vue vs React

프레임워크별 필요 기술스택



Vue vs React



데이터 바인딩 방식	데이터 바인딩을 프레임워크가 어느정도까지 알아서 해준다.	기본적으로 모든 페이지는 re-rendering이 되며, 예외의 경우에 대해 개발자가 코드상에서 직접 변경
적용사이트	Alibaba, Baidu, Tencent, Xiaomi, Didi Chuxing, Gitlab 등	instagram, netflix, dropbox, pinterest, twitter, naver facebook 등
Bundle Size	약 20.9KB	약 31.8KB
Performance	속도는 비슷하며, 메모리 소비는 Vue가 React에 비해 적다	

출처: <http://www.stefankrause.net/js-frameworks-benchmark7/table.html>

22

02

—— Vue.js 기술 상세

Subjects

1. 뷰 인스턴스 & 컴포넌트
2. 뷰 라우터
3. 뷰 HTTP통신
4. 뷰 템플릿

01

뷰 인스턴스 & 컴포넌트

뷰 인스턴스

뷰 인스턴스의 형식

```
new Vue({  
  ...  
});
```

- 뷰로 화면을 개발하기 위한 필수단위.
- 뷰 생성자로 생성한다.
- 미리 정해져 있는 인스턴스 옵션 속성을 설정하여 사용한다.

인스턴스 옵션 정의 및 인스턴스 생성

```
new Vue({  
  el: '#app',  
  data: {  
    message: 'Hello'  
  }  
});
```

속성

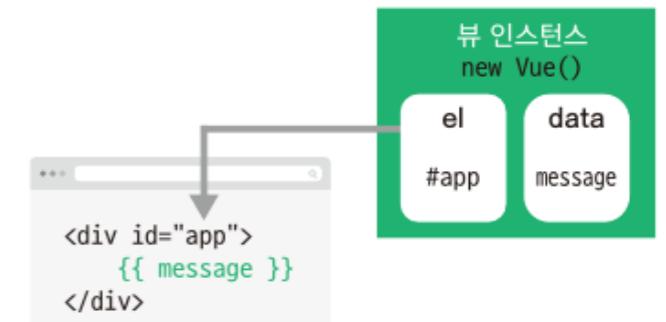
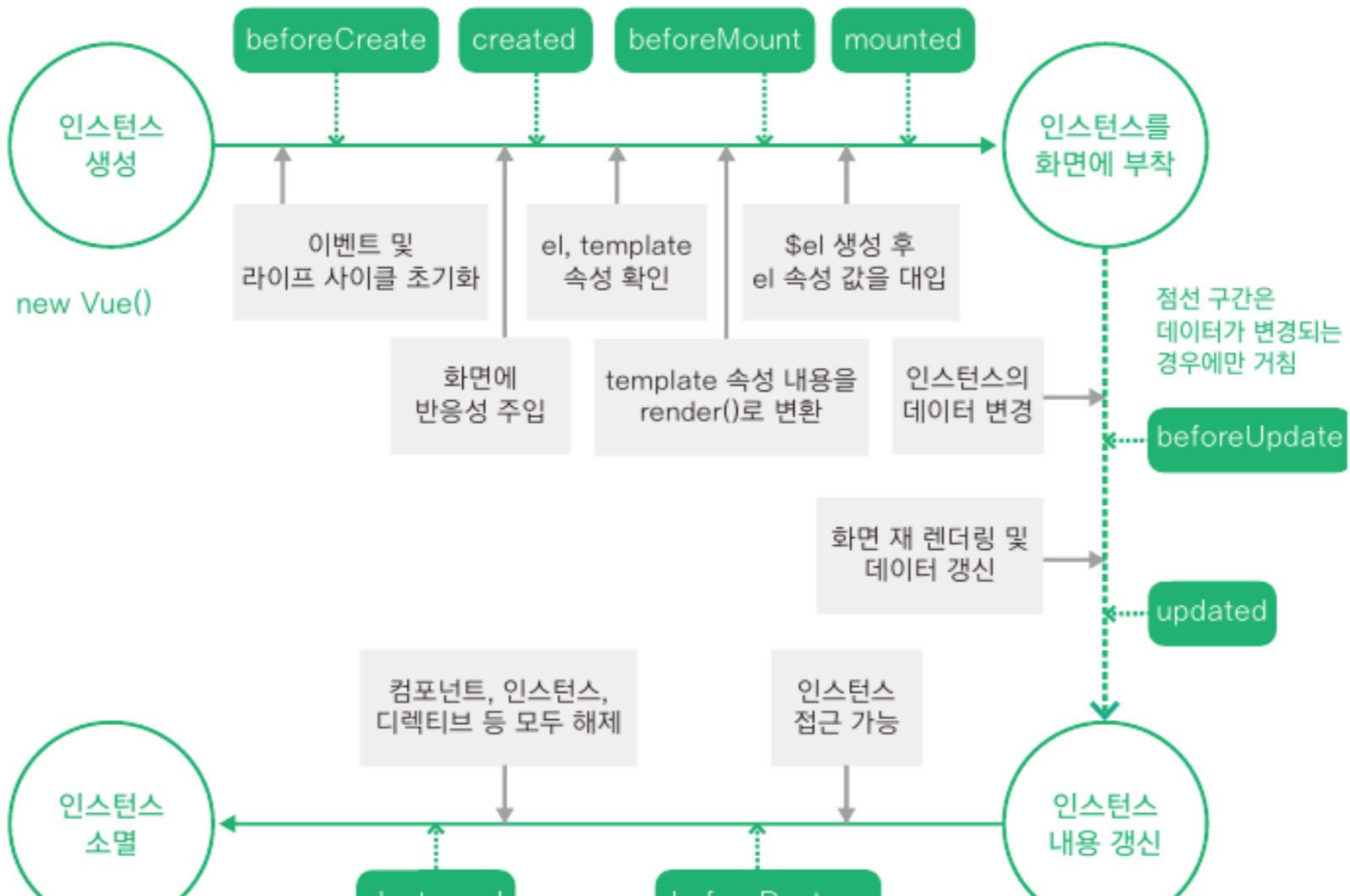
설명

template 화면에 표시할 HTML등의 마크업 요소를 정의하는 속성

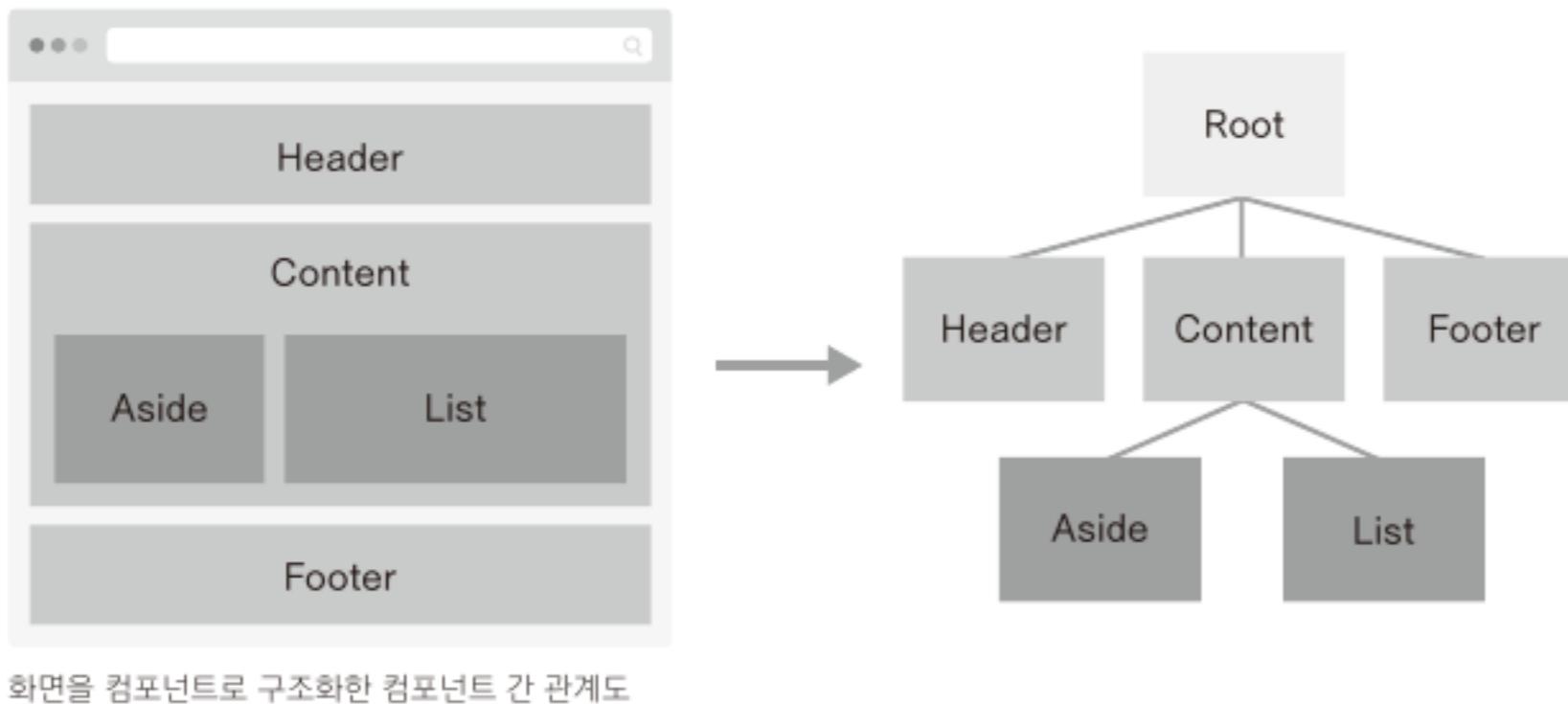
methods 이벤트 처리와 같이 화면 로직 제어와 관계된 메서드를 정의하는 속성

created 뷰 인스턴스가 생성되자마자 실행할 로직을 정의할 수 있는 속성

뷰 인스턴스 인스턴스 라이프 사이클



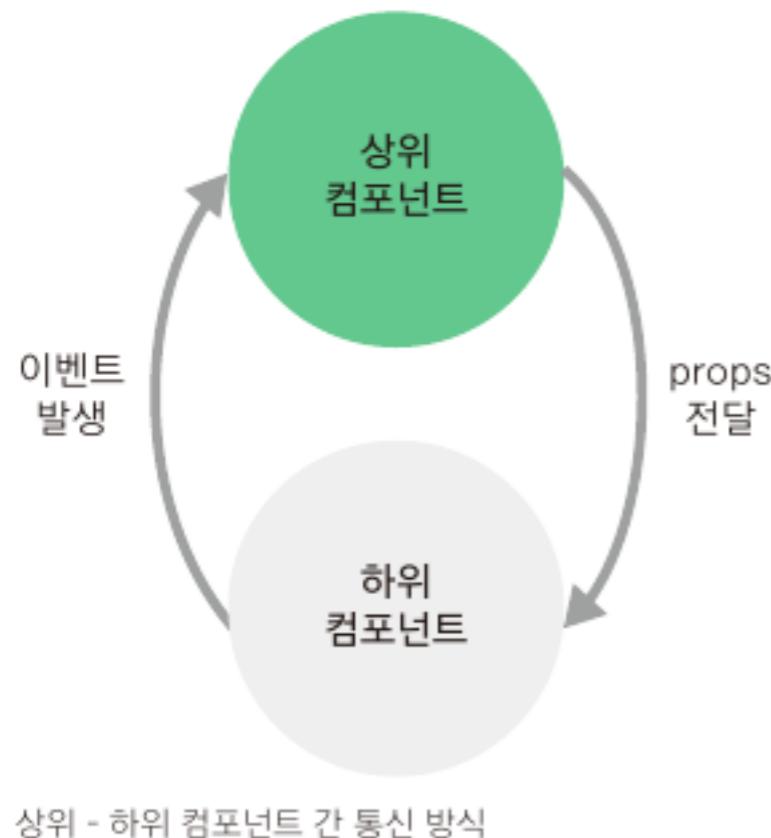
뷰 컴포넌트



- 조합하여 화면을 구성할 수 있는 **화면의 특정 영역**을 의미한다.
- 컴포넌트를 활용하면 화면을 빠르게 구조화해 일괄적 패턴으로 개발 가능.
- 각 컴포넌트는 자체적으로 고유한 유효 범위를 가진다.
- 각 컴포넌트는 트리와 유사한 관계의 구조를 지닌다.

컴포넌트 통신

- 뷰는 한 화면을 여러 컴포넌트로 구성한다. 따라서 같은 웹페이지라도 서로간에 직접 데이터를 공유할 수 없다.
- 이는 **컴포넌트마다 독립적인 유효범위(scope)**를 갖기 때문.



- 뷰의 기본적인 데이터 전달방법
: 상위(부모) -> 하위(자식)
- 상위 컴포넌트에 등록된 하위 컴포넌트로 **props**라는 특수한 속성 전달하여 값 전달.
- 하위 컴포넌트에서는 **이벤트**를 발생시켜 상위 컴포넌트에 신호를 보낸다.
 - \$emit : 이벤트 발생
 - v-on : 이벤트 수신
- 같은 레벨 간 컴포넌트 통신에는 이벤트버스 활용

이벤트를 발생시키고 수신하기

event2 이벤트는
<child>에 정의한
v-on:event2에
전달되고, v-on:
event2의 대상 메서
드인 최상위 컴포넌
트의 메서드 event3
가 실행된다.

event1() 메서드
안의 this.\$emit
(‘event2’)가 실행되
면서 event2
이벤트가 발생

```
<body>
  <div id="app">
    {{message}}
    <child v-on:event2="event3"></child>
  </div>

  <script src="https://cdn.jsdelivr.net/npm/vue@2.5.2/dist/vue.js"></script>
  <script>

    Vue.component('child',{
      template: '<button v-on:click="event1">I am a child component</button>',
      methods: {
        event1: function(){
          this.$emit("event2");
        }
      }
    });

    var app = new Vue({
      el: '#app',
      data: {
        message: "I am a parent component"
      },
      methods: {
        event3: function(){
          this.message = "Received event!";
        }
      }
    });
  </script>
</body>
```

버튼 클릭시 클릭
이벤트 v-on:
click="event1"
에 따라 event1()
메서드 실행.

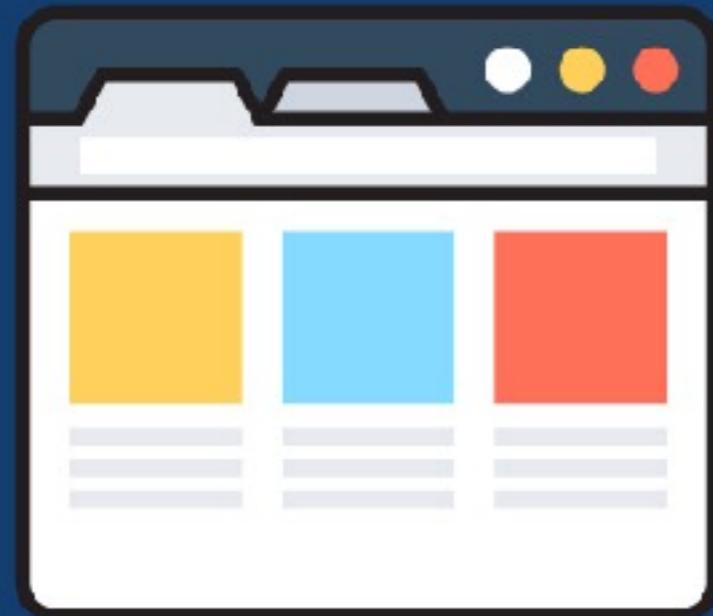
event3()는 컴포넌
트 data 속성의
message값을 바
꾼다. 데이터가 바
뀌면 화면도 반응하
여 바뀌게 된다.

02

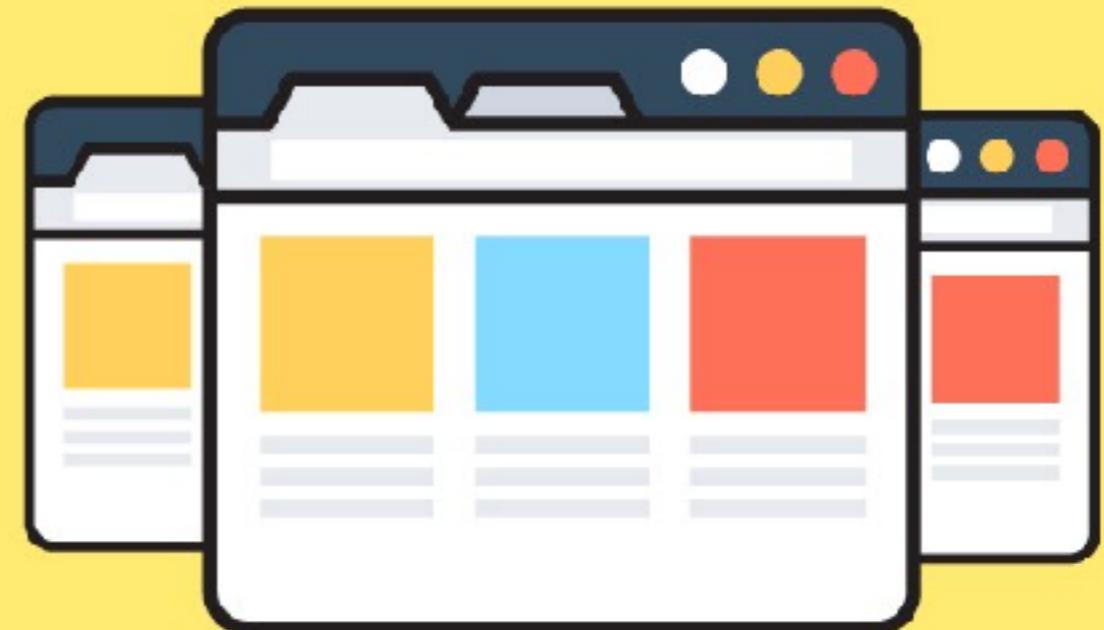
뷰 라우터

라우팅이란?

Multiple Page Applications



VS



Single Page Applications



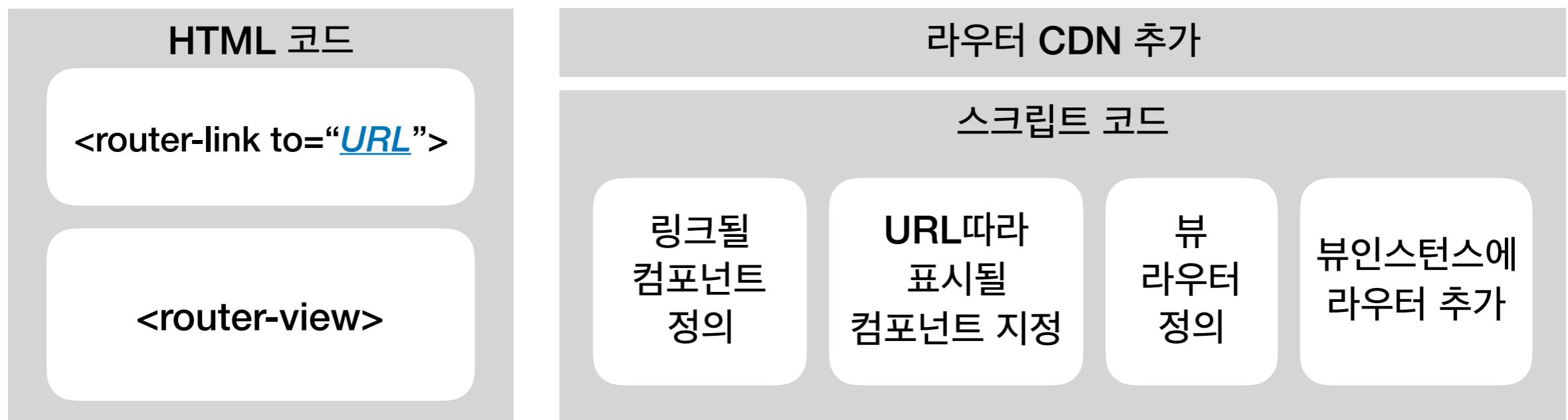
www.techuz.com

뷰 라우터

- 뷰 라우터는 뷰에서 라우팅 기능을 구현할 수 있도록 지원하는 공식 라이브러리이다.
- 뷰 라우터를 구현하기 위해 필요한 특수 태그와 기능은 아래와 같다.

태그	설명
<code><router-link to="URL값"></code>	페이지 이동 태그. 화면에서는 <code><a></code> 로 표시되며, 클릭하면 to에 지정한 URL로 이동합니다.
<code><router-view></code>	페이지 표시 태그. 변경되는 URL에 따라 해당 컴포넌트를 뿌려주는 영역입니다.

- 뷰 라우터 사용



네스티드 라우터 & 네임드 뷰

- 실제 웹 앱은 화면이 여러개로 분할된 경우가 많다.
- 네스티드 라우터와 네임드 뷰는 여러 개의 컴포넌트를 동시에 표시할 수 있는 라우터이다.

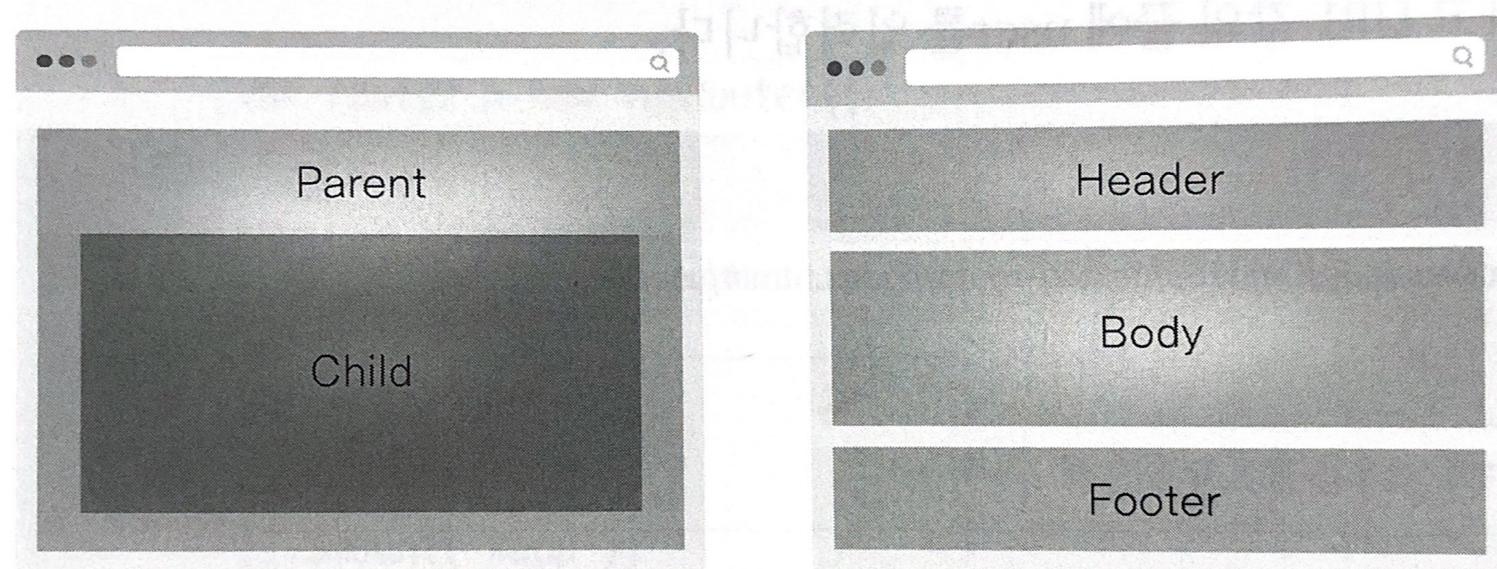
네스티드 라우터의 구조



- 상위 컴포넌트 1개에 하위 컴포넌트 1개를 포함하는 구조
- URL에 따라서 컴포넌트의 하위 컴포넌트가 다르게 표시된다.
- 링크되는 기본 컴포넌트(user) 내에도 하위 컴포넌트(post, profile) 내용을 표시할 `<router-view>`가 있다.
- 한번에 많은 수의 컴포넌트를 표시하는 데에는 한계가 있다.

네스티드 라우터 & 네임드 뷰

- 실제 웹 앱은 화면이 여러개로 분할된 경우가 많다.
- 네스티드 라우터와 네임드 뷰는 여러 개의 컴포넌트를 동시에 표시할 수 있는 라우터이다.

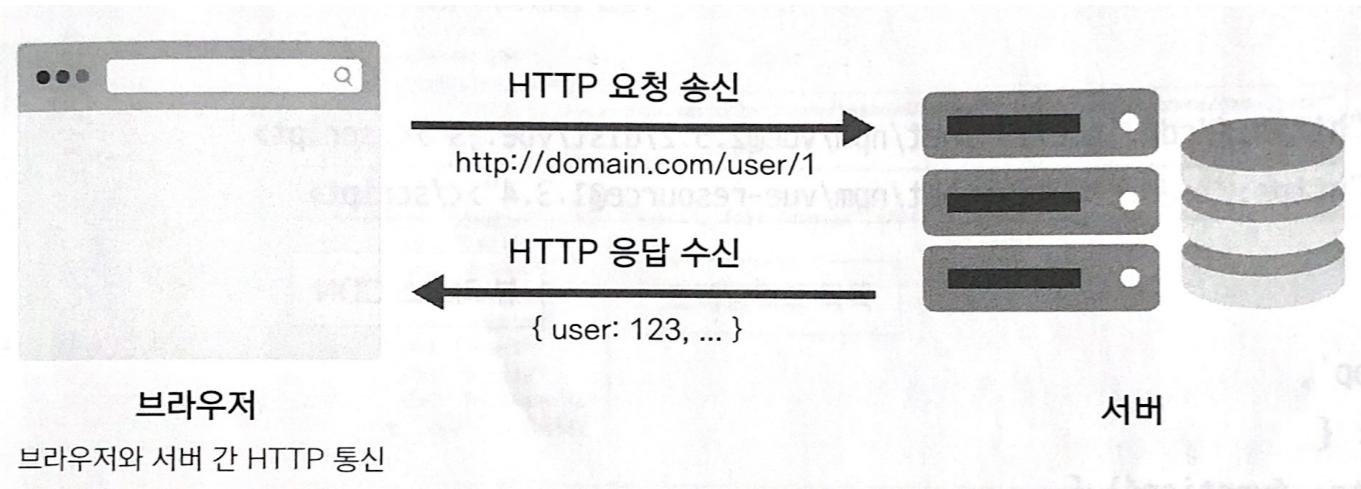


- 네임드 뷰는 특정 페이지로 이동했을 때 **여러개의 컴포넌트를 동시에 표시하는** 라우팅 방식이다.
- 네스티드 라우터는 상위 컴포넌트가 하위 컴포넌트를 포함하는 형식이라면
네임드 뷰는 **같은 레벨에서 여러개의 컴포넌트를 한번에 표시한다**.
- 이름과 같이 <router-view>에 name이 지정된 형태이다.

03

뷰 HTTP 통신

웹 앱의 HTTP 통신



- 사용자와 상호작용에 따라 동적으로 데이터를 화면에 표시해야 하는 현재, 웹 앱의 HTTP 통신구현은 필수다.
- 웹 앱 HTTP 통신의 대표적 사례는 제이쿼리의 ajax



- 서버에서 받아온 데이터를 표시할 때 화면 전체를 갱신하지 않고도 화면의 일부분만 변경할 수 있도록 하는 자바스크립트 기법
- React, Angular 등에서도 Ajax를 사용하고 있다.
- 뷰에서도 마찬가지로 Ajax를 위한 라이브러리 제공.

Axios(액시오스)

- 현재 뷰 커뮤니티에서 가장 많이 사용되는 HTTP 통신 라이브러리
- 별도 로직구현 필요 없이 제공되는 API 만으로도 간편하게 원하는 기능을 구현할 수 있다.

API 유형	처리 결과
<code>axios.get('URL').then().catch()</code>	<ul style="list-style-type: none">• 해당 URL에 대해 HTTP GET 요청을 보낸다.• 데이터를 정상적으로 받아오면 then()의 로직이, 오류가 발생하면 catch()의 로직이 실행된다.
<code>axios.post('URL').then().catch()</code>	<ul style="list-style-type: none">• 해당 URL에 대해 HTTP POST 요청을 보낸다.• then()과 catch() 로직의 동작은 동일
<code>axios({options})</code>	<ul style="list-style-type: none">• HTTP 요청에 대한 자세한 속성들을 직접 정의하여 보낸다.• (ex) 요청을 보낼 URL, 요청방식, 보내는 데이터 유형..

- 액시오스의 응답 데이터는 일반 문자열이 아닌 객체 형태이기 때문에 별도로 `JSON.parse()`를 사용하여 변환할 필요가 없다는 점에서 편리한다.
- 이외에도 초기에 뷰 코어 팀에서 공식적으로 권하는 HTTP 통신 라이브러리였던 뷰 리소스를 사용할 수 있다.(2016년 에반의 공식적 지원 중단됨.)

04

뷰 템플릿

뷰 템플릿이란?

- HTML, CSS 등의 마크업 속성과 뷰 인스턴스에서 정의한 데이터, 로직들을 연결하여 사용자가 브라우저에서 볼 수 있는 형태의 HTML로 변환해주는 뷰 인스턴스 속성이다.

```
<script>
  new Vue({
    template: '<p>Hello {{msg}}</p>'
  });
</script>
```

라이브러리 내부적으로 **template** 속성에서 정의한 데이터
를 가상-dom 기반의 render() 함수로 변환한다. 변환과정에
서 뷰의 반응성이 화면에 더해지며, 변환된 render() 함수는
화면을 그리게 된다.

템플릿에서 사용하는 주요 속성과 문법

01 데이터바인딩

{{ }} 콧수염괄호 (이중 중괄호)

뷰 인스턴스를 HTML 태그에 연결하는 가장 기본적인 텍스트 삽입방식

v-bind

HTML 속성값에 뷰 데이터값을 연결할때 사용하는 데이터 연결방식

02 디렉티브

뷰 디렉티브란 **HTML 태그 안에 v- 접두사를 갖는 모든 속성을** 의미한다.

뷰의 데이터 값 변경시 화면 요소들이 반응하여 갱신된다. 이처럼 화면 요소를 직접 제어할 필요 없이 뷰의 디렉티브 활용하여 조작 가능.

(ex) v-if, v-for, v-show, v-bind, v-on, v-model 등

03 이벤트처리

v-on 디렉티브와 methods 속성을 활용한 이벤트 처리

03

Vue.js로 화면개발하기

Subjects

1. Vue 프로젝트 구성방법
2. Vue.js로 구현한 할일관리 앱

01

Vue 프로젝트 구성방법

싱글 파일 컴포넌트 체계

- .vue 파일로 프로젝트 구조를 구성하는 방식. .vue 파일 1개는 컴포넌트 1개와 동일하다.

```
<template>  
  <!—HTML태그내용—>  
</template>
```

화면에 표시할 요소들을 정의하는 영역.
예) HTML + 뷰 데이터 바인딩

```
<script>  
export default {  
  // 자바스크립트 내용  
}  
</script>
```

뷰 컴포넌트의 내용을 정의하는 영역
예) template, data, methods 등

```
<style>  
/* CSS 스타일 내용 */  
</style>
```

템플릿에 추가한 HTML태그의 CSS
스타일을 정의하는 영역

개발환경 설정

1. Node.js 설치



2. Vue CLI 설치



3. Vue 개발자 도구 설치



4. VisualStudio Code 설치



5. Git 설치



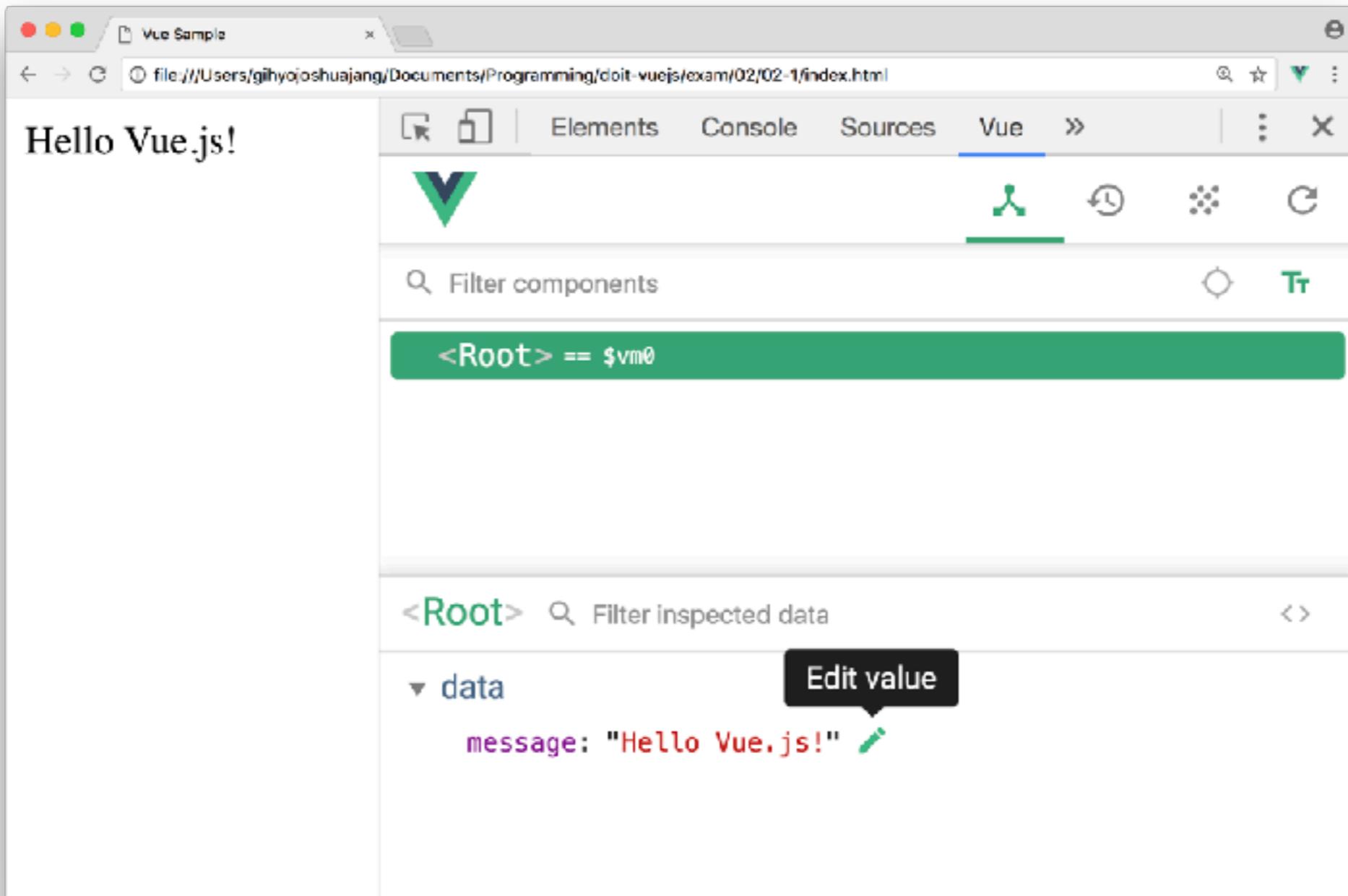
뷰 CLI 도구

- vue 개발자들이 편하게 프로젝트를 구성할 수 있도록 뷰 코어 팀에서 제공하는 도구.
- 커맨드 창에서 명령어로 뷰 개발을 위한 프로젝트 구성과 관련된 여러 작업을 수행할 수 있다.



뷰 개발자 도구

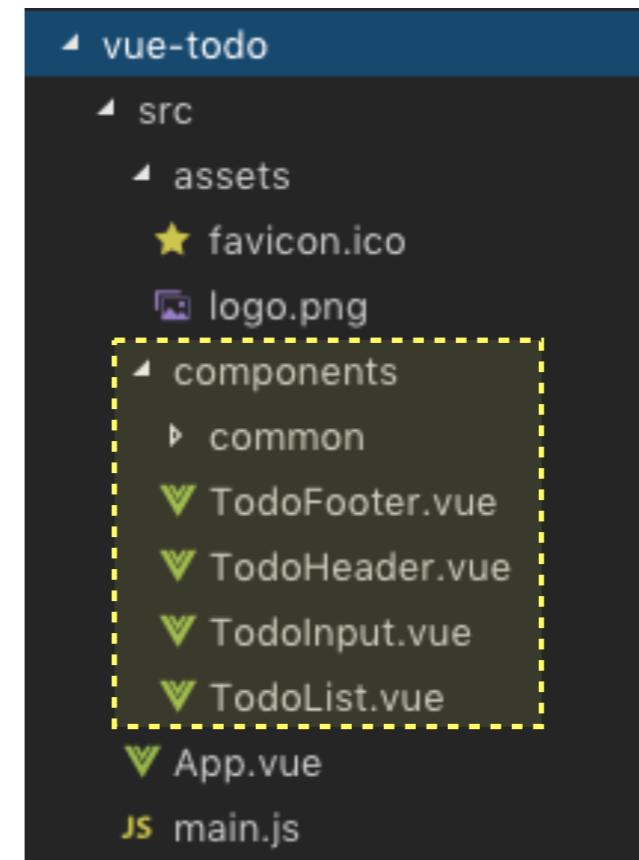
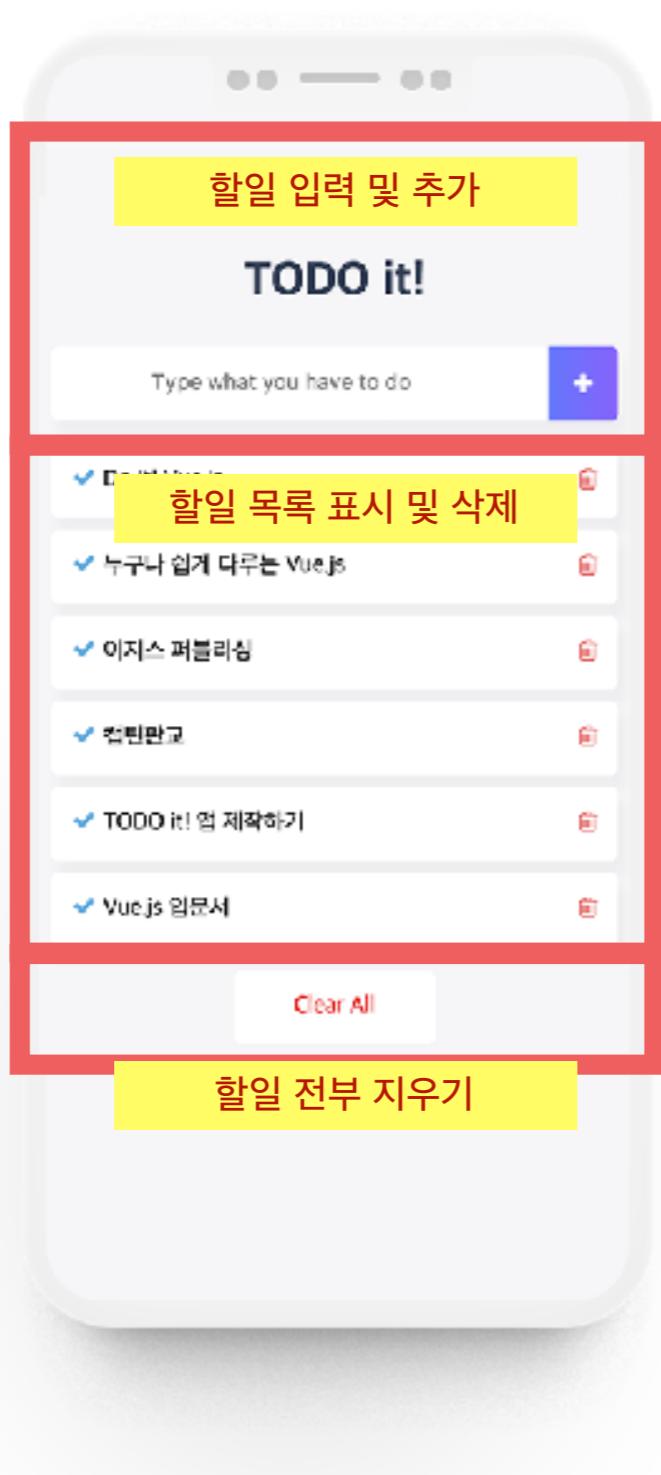
- 뷰 개발에 도움을 주는 뷰 크롬 플러그인. 뷰로 만든 웹앱의 구조를 디버깅, 분석할 수 있다.
- 크롬, 사파리, 파이어폭스 등에서 모두 지원된다.



02

Vue로 구현한 할일관리 앱

애플리케이션 설계



프로젝트 폴더구조

애플리케이션 코드 예시 할일 추가

HTML 태그 내용

```
<template>
  <div class="inputBox shadow">
    <input type="text" v-model="newTodoItem" placeholder="Type what you have to do" v-on:keypress.enter="addTodo">
    <button v-on:click="addTodo">추가</button>
  </div>
</template>
```

```
export default {
  data() {
    return {
      newTodoItem: ''
    }
  },
  methods: {
    addTodo() {
      if (this.newTodoItem !== "") {
        var value = this.newTodoItem && this.newTodoItem.trim();
        this.$emit('addTodo', value)

        localStorage.setItem(todoItem, todoItem);

        this.clearInput();
      },
      clearInput() {
        this.newTodoItem = '';
      }
    }
  }
}
```

자바스크립트 내용

```
<template>
  <!— HTML태그내용 —>
</template>
```

```
<script>
export default {
  // 자바스크립트 내용
}
</script>
```

```
<style>
  /* CSS 스타일 내용 */
</style>
```

애플리케이션 코드 예시 할일 추가

HTML 태그 내용

```
<template>
  <div class="inputBox shadow">
    <input type="text" v-model="newTodoItem" placeholder="Type what you have to do" v-on:keypress.enter="addTodo">
    <button v-on:click="addTodo">추가</button>
  </div>
</template>
```

```
export default {
  data() {
    return {
      newTodoItem: ''
    }
  },
  methods: {
    addTodo() {
      if (this.newTodoItem !== "") {
        var value = this.newTodoItem && this.newTodoItem.trim();
        this.$emit('addTodo', value)

        localStorage.setItem(todoItem, todoItem);

        this.clearInput();
      }
    },
    clearInput() {
      this.newTodoItem = '';
    }
  }
}
```

자바스크립트 내용

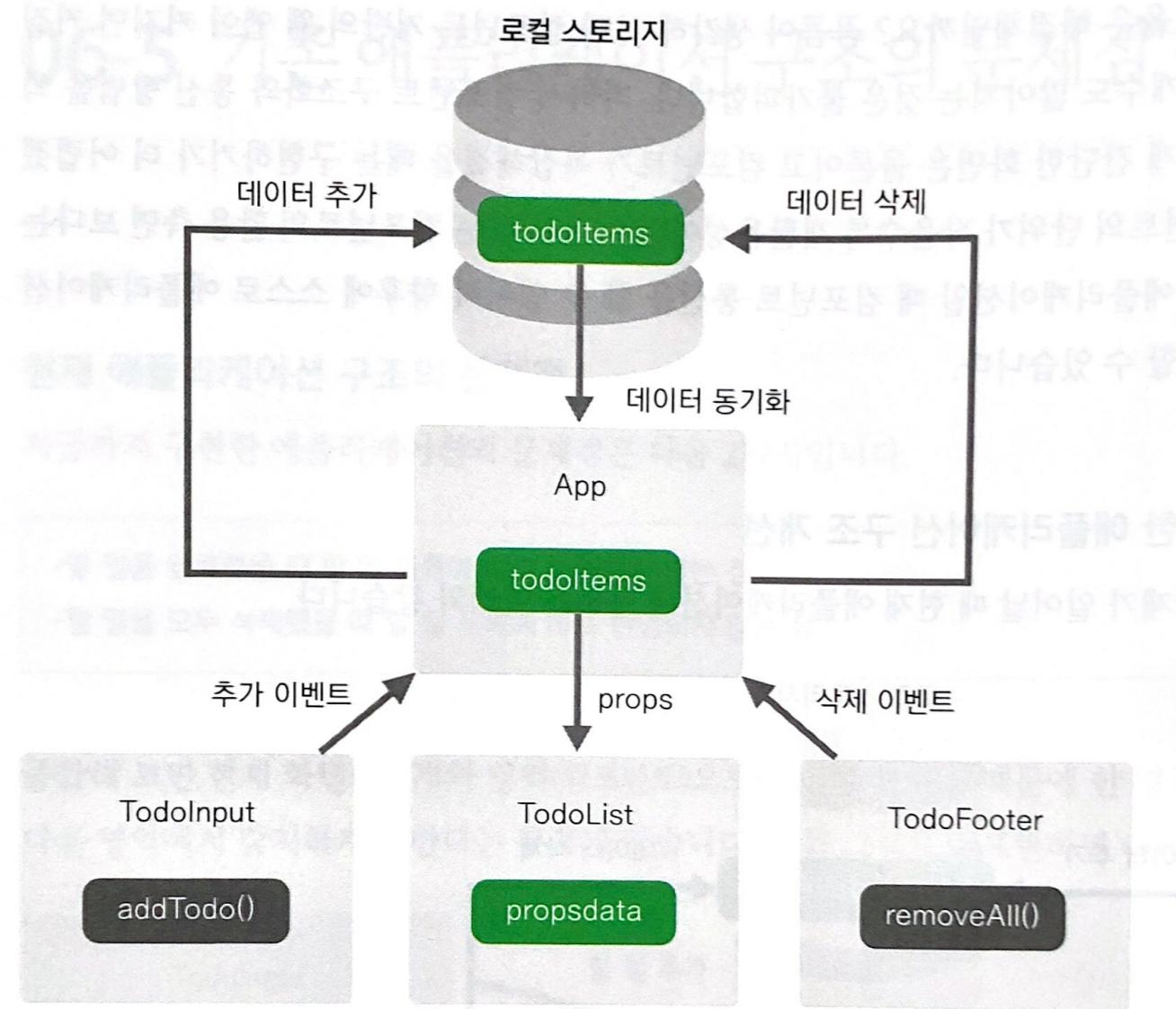
```
<template>
  <!— HTML태그내용 —>
</template>
```

```
<script>
export default {
  // 자바스크립트 내용
}
</script>
```

```
<style>
</style>
```

데이터의 변경(할일 추가/삭제)이 화면에 바로 반영되지 않는 구조!

애플리케이션 구조



- 로컬 스토리지의 데이터 조회, 추가, 삭제 등의 작업을 모두 App 컴포넌트에서 한다.
- 하위컴포넌트들은 데이터 표현 혹은 데이터 조작에 대한 요청(이벤트 발생)만 담당

애플리케이션 코드 예시 할일 지우기

The diagram illustrates the flow of an event from the `removeTodo` method in `TodoList.vue` to the `removeAll` event handler in `App.vue`.

TodoList.vue 코드

```
<template>
  <section>
    <li v-for="(todoItem, index) in propsdata" :key="todoItem" class="shadow">
      {{ todoItem }}
      <span class="removeBtn" type="button" @click="removeTodo(todoItem, index)">
        ...
      </span>
    </li>
  </section>
</template>

methods: {
  ...
  removeTodo(todoItem, index) {
    localStorage.removeItem(todoItem);
    this.todoItems.splice(index, 1);
  }
},
```

App.vue 코드

```
<template>
  <div id="app">
    <TodoHeader></TodoHeader>
    <TodoInput v-on:addTodo="addTodo"></TodoInput>
    <TodoList v-bind:propsdata="todoItems" @removeTodo="removeTodo"></TodoList>
    <TodoFooter v-on:removeAll="clearAll"></TodoFooter>
  </div>
</template>
```

TodoList컴포넌트에서 App으로 이벤트 전달

An orange arrow points from the `@click="removeTodo(todoItem, index)"` event in the `TodoList.vue` template to the `removeTodo` method in the `TodoList.vue` script. Another orange arrow points from the `@removeTodo="removeTodo"` event in the `App.vue` template to the `removeTodo` method in the `App.vue` script.

애플리케이션 코드 예시 할일 목록 표시

```
<template>
  <section>
    <li v-for="(todoItem, index) in propsdata" :key="todoItem" class="shadow">
      {{ todoItem }}
      <span class="removeBtn" type="button" @click="removeTodo(todoItem, index)">
      </span>
    </li>
  </section>
</template>
```

```
<script>
export default {
  props: ['propsdata'],
  methods: {
    removeTodo(todoItem, index) {
      this.$emit('removeTodo', todoItem, index);
    }
  }
}</script>
```

TodoList.vue 코드

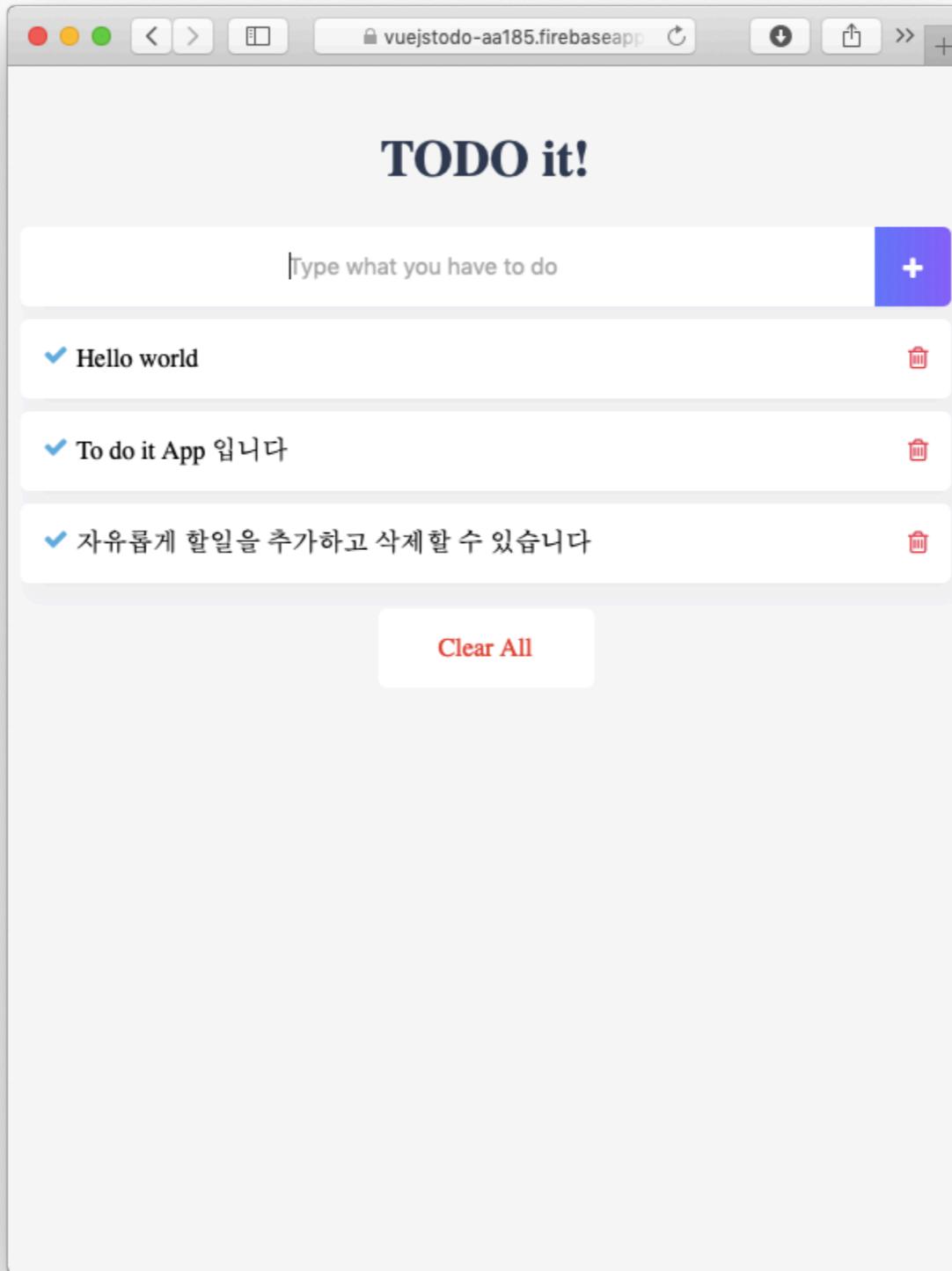
```
export default {
  data() {
    return {
      todoItems: []
    }
  },
  methods: {
    clearAll() {
      localStorage.clear();
      this.todoItems = [];
    },
    addTodo(todoItem) {
      localStorage.setItem(todoItem, todoItem);
      this.todoItems.push(todoItem);
    },
    removeTodo(todoItem, index) {
      localStorage.removeItem(todoItem);
      this.todoItems.splice(index, 1);
    }
  }
},
```

```
<template>
  <div id="app">
    <TodoHeader></TodoHeader>
    <TodoInput v-on:addTodo="addTodo"></TodoInput>
    <TodoList v-bind:propsdata="todoItems" @removeTodo="removeTodo"></TodoList>
    <TodoFooter v-on:removeAll="clearAll"></TodoFooter>
  </div>
</template>
```

App.vue 코드

선언한 todoItems 속성을 TodoList
컴포넌트에 props로 전달

최종 완성 앱



References

- 자료 제작에 참고한 문헌 및 사이트
 - Doit! Vue.js, 장기효 저
 - <https://vuejs.org/v2/guide/> : 뷰 공식 가이드
 - <https://github.com/joshua1988/doit-vuejs/> : 관련 예제 코드
 - <https://vuejstodo-aa185.firebaseioapp.com/> : TodoIt 앱 확인 링크
 - <https://codesandbox.io/>

감사합니다.