



Todo 앱을 Vue.js로 구현해봤으면 좋았을 걸

2018.04.22 장기효 a.k.a Captain Pangyo

Sponsored by  **이지스퍼블리싱**



PROFILE



“2018년 3월 26일 Evan You와 함께”

5년차 웹 개발자

Do it! Vue.js 입문 저자

연예인 예술협회 가수

패스트 캠퍼스 Vue.js 강사

에반 유 수업 수강생(O)



TIMETABLE

시간	세션
16:30 ~ 17:10(40분)	Part 1 - 현대 프런트엔드 개발과 Vue.js
17:10 ~ 17:20(10분)	Break Time
17:20 ~ 18:00(40분)	Part 2 - Vue.js 동작원리와 자가 학습 방법
18:00 ~ 18:10(10분)	Break Time
18:10 ~ 18:45(35분)	Part 3 - Vue.js로 구현하는 Todo App
18:45 ~ 19:00(15분)	Q & A



01

현대 프런트엔드 개발과 Vue.js

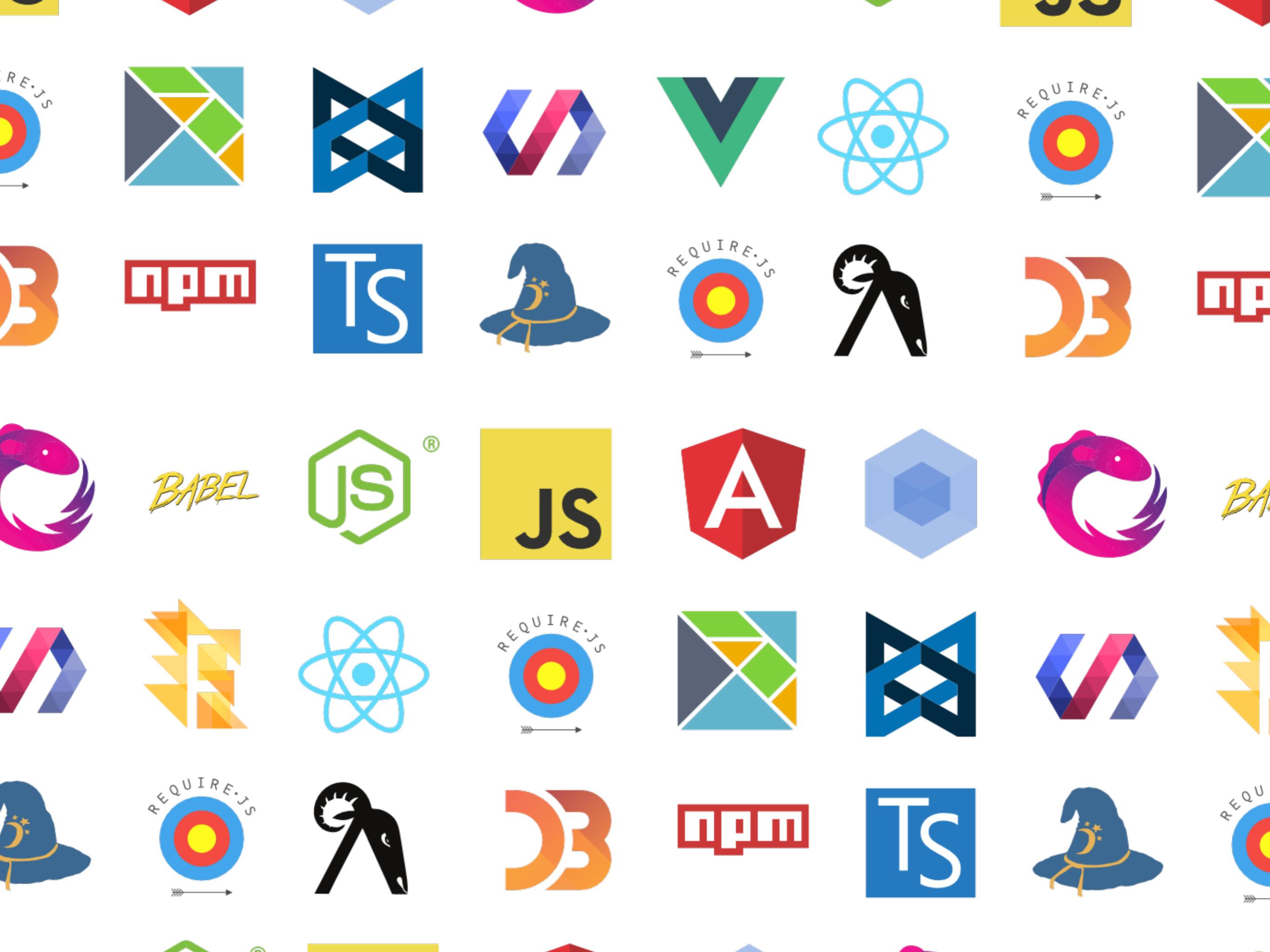
Subjects

1. 2018년 프런트엔드 개발 현황
2. Vue.js가 주목받는 이유
3. Angular vs React vs Vue
4. Vue.js 살펴보기



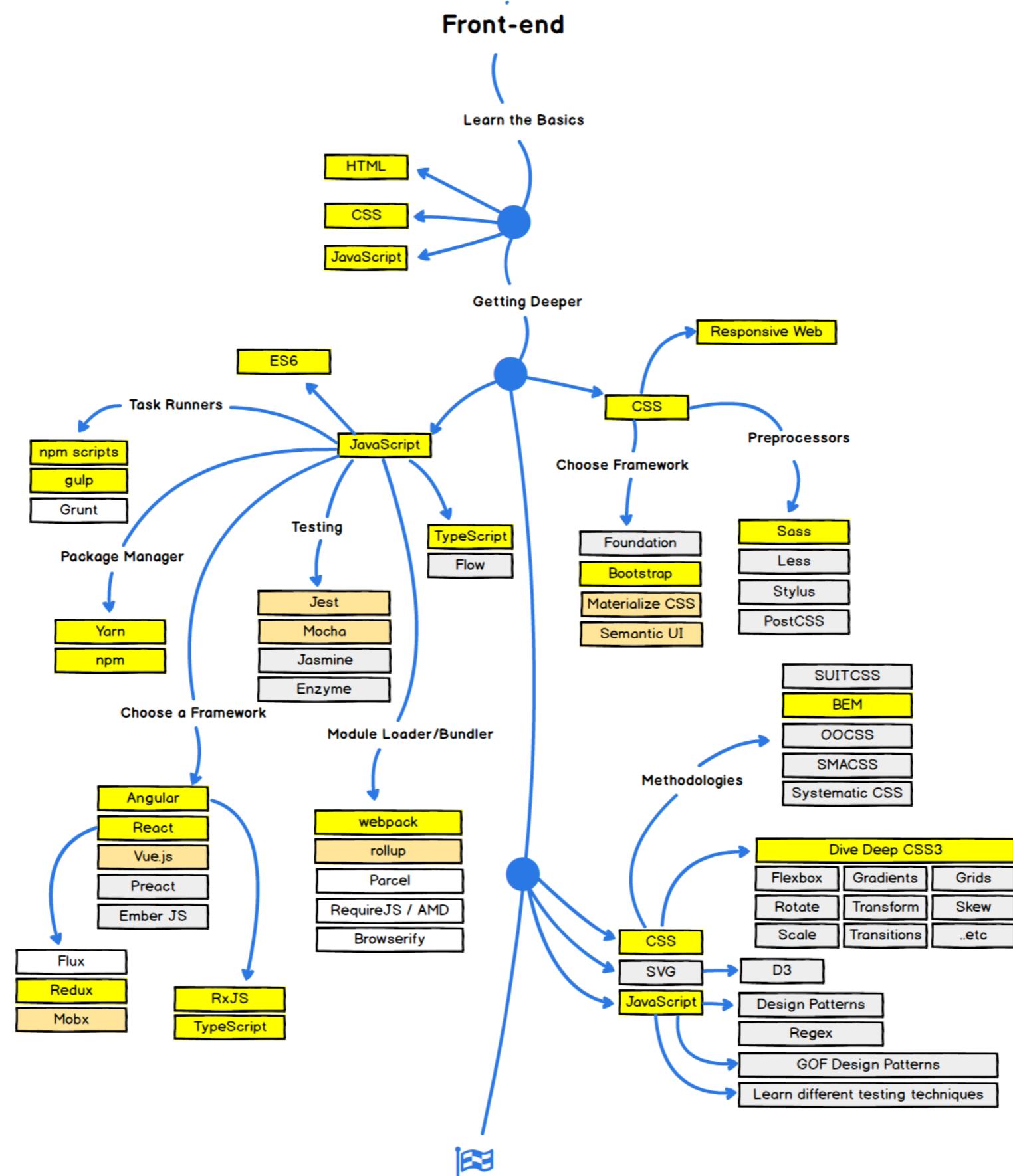
01

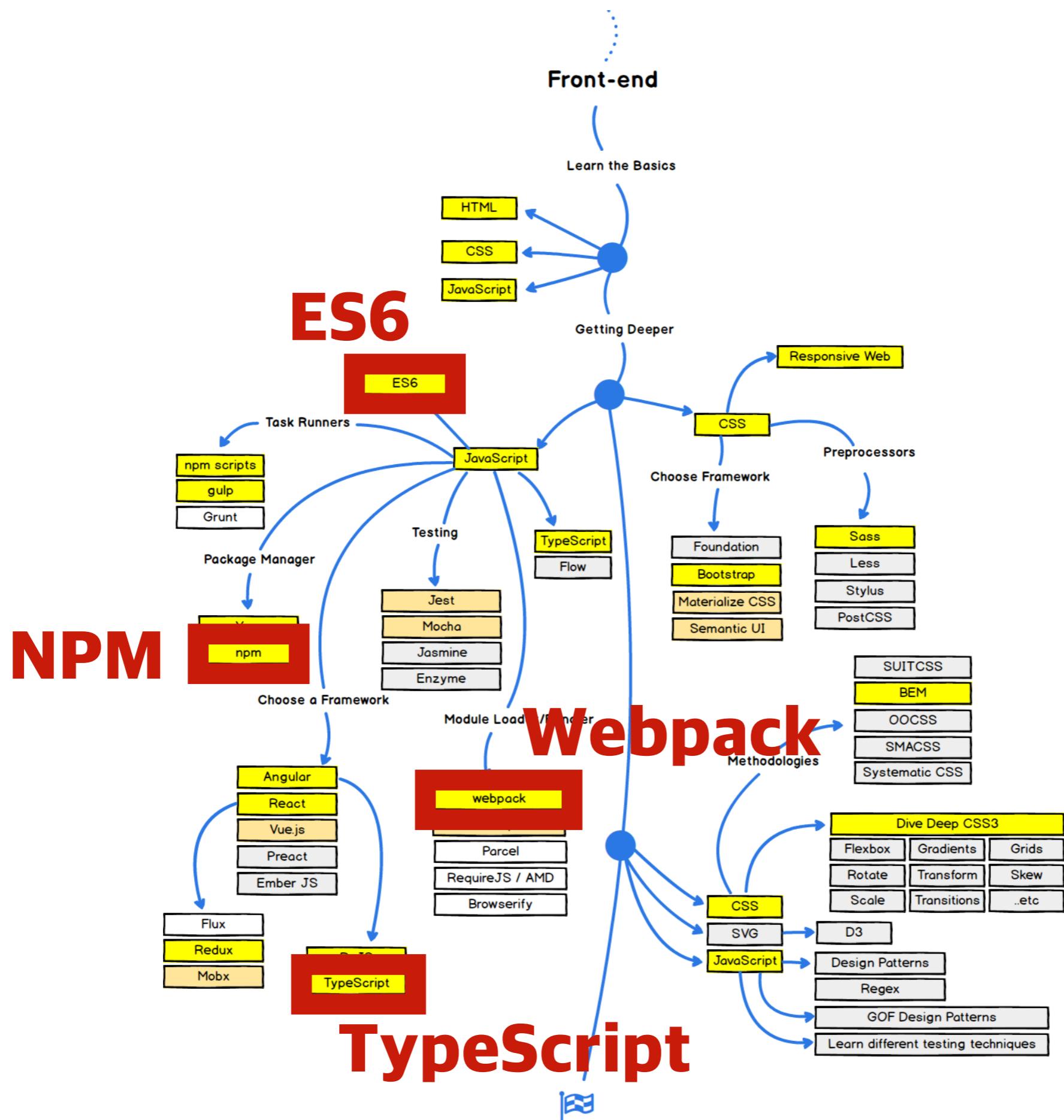
2018년 프런트엔드 개발 현황





2018년 프런트엔드 개발자가 되는 방법







**도구가 많아서 좋지만
입문자에게는 부담이…**



화면 개발 방식

과거



화면 개발 방식

과거



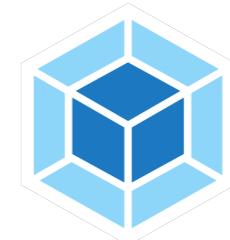
현재



패키지 관리 도구



최신
자바스크립트 문법



모듈 번들러



타입에 엄격한
자바스크립트



복잡하기만한 2018년 프런트엔드 개발 생태계



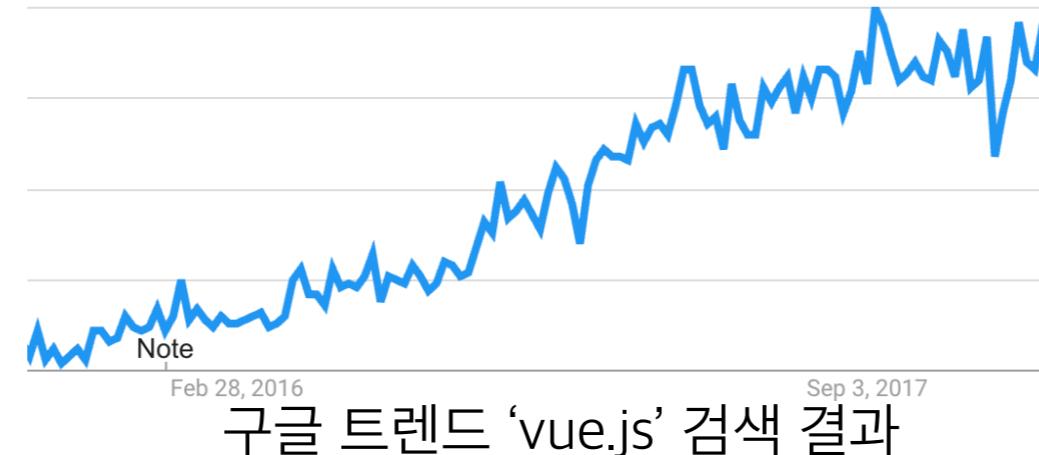
02

Vue.js가 주목받는 이유



1. 통계: 객관적 수치

1. 통계: 객관적 수치



	#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

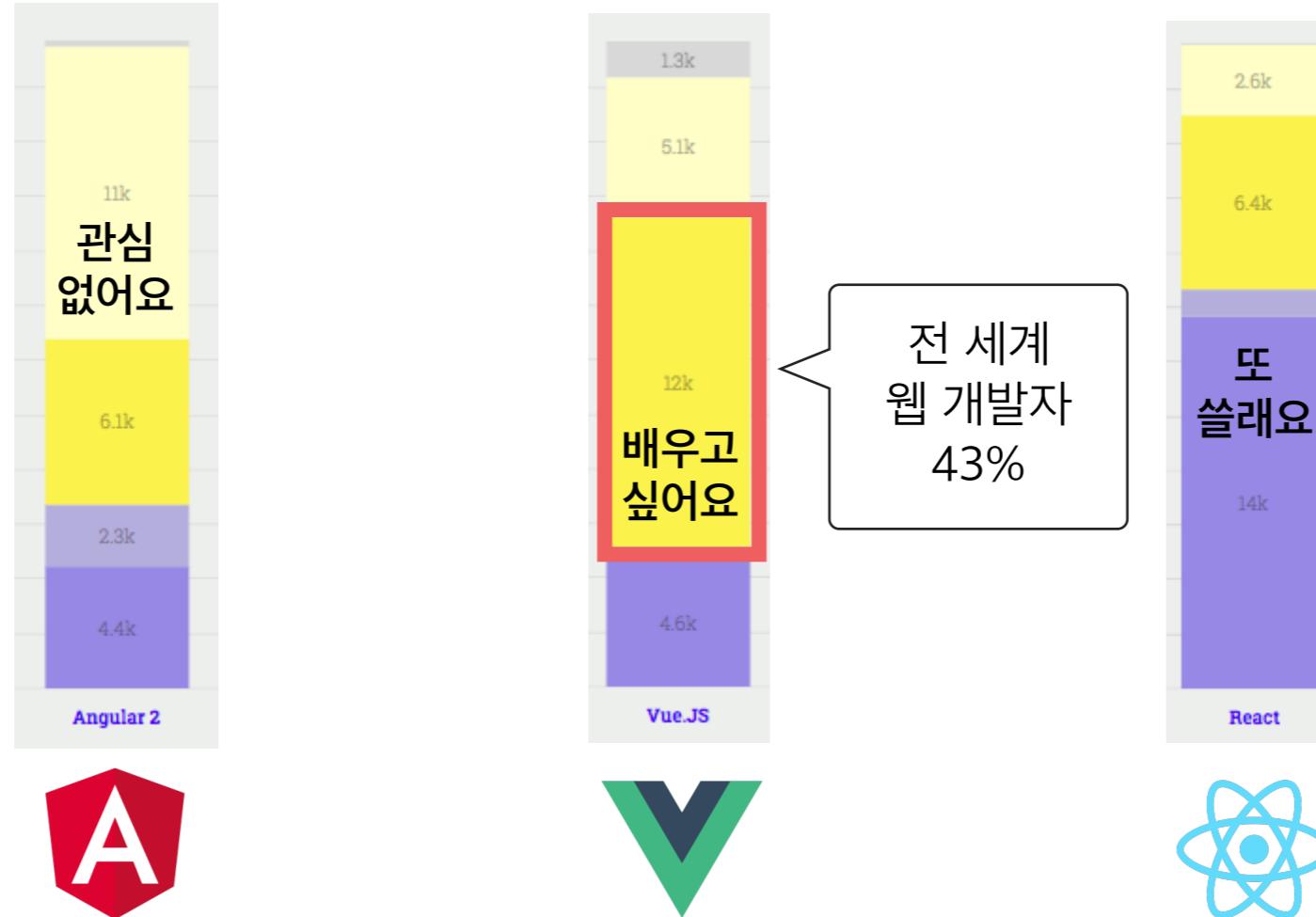
1	Vue.js A progressive, incrementally-adoptable JavaScript framew...	+40.0k★
2	React A declarative, efficient, and flexible JavaScript library for buildi...	+27.8k★
3	Create React App Create React apps with no build configuration.	+22.5k★
4	Puppeteer Headless Chrome Node API	+22.0k★
5	Axios Promise based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

2017년 1위

구글 트렌드 & 깃헙

출처: <https://risingstars.js.org/>

1. 통계: 객관적 수치



2017년 전세계 웹 개발자 설문조사

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



1. 통계: 객관적 수치

96%

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

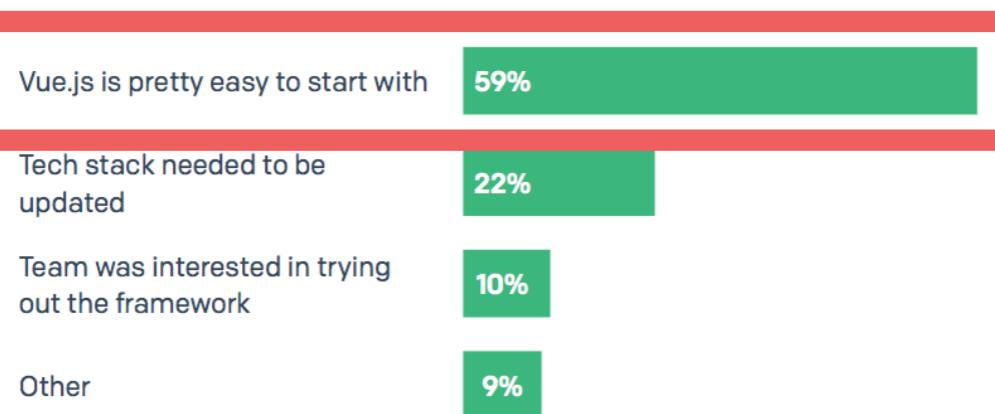
다음 프로젝트도 Vue.js를 쓸거에요

81%

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

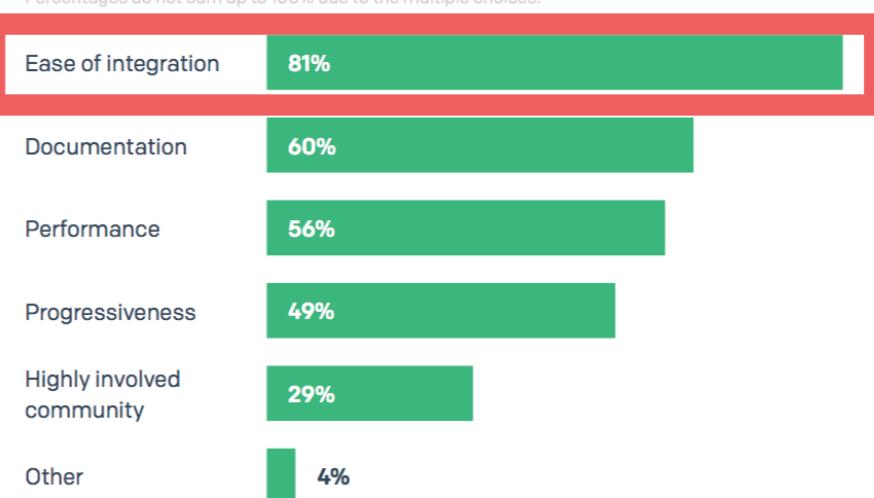
기존 기술 스택과 함께 사용하기 쉬워요

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

출처: State of Vue.js 2017 Report



2. 사용성: 얼마나 편리한가

2. 사용성: 얼마나 편리한가



2. 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>
```

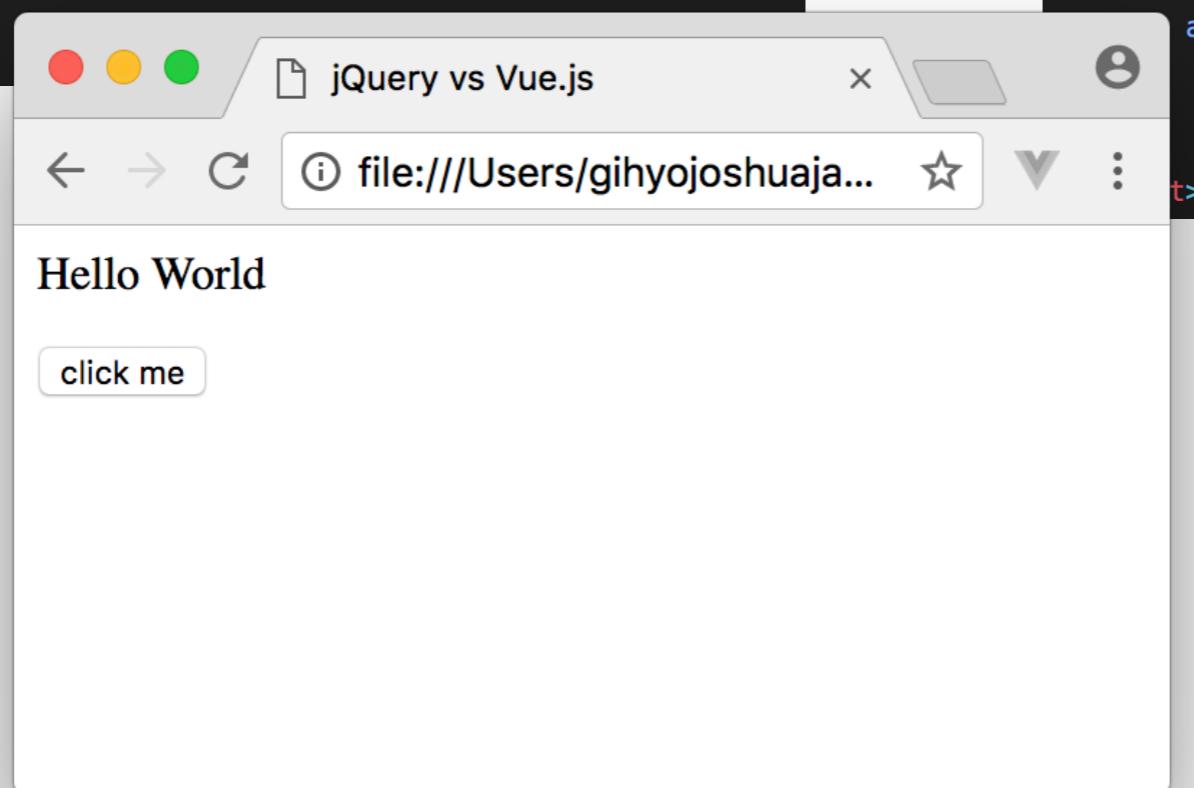
CDN 라이브러리 로딩

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



2. 사용성: 얼마나 편리한가



Good tools make application development quicker and easier to maintain than if you did everything by hand.

The [Angular CLI](#) is a *command line interface* tool that can create a project, add files, and perform a variety of ongoing development tasks such as testing, bundling, and deployment.

The goal in this guide is to build and run a simple Angular application in TypeScript, using the Angular CLI while adhering to the [Style Guide](#) recommendations that benefit every Angular project.

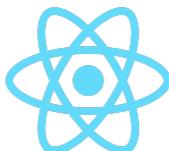
By the end of the chapter, you'll have a basic understanding of development with the CLI and a foundation for both these documentation samples and for real world applications.

2. 사용성: 얼마나 편리한가

QuickStart



Hello World



The easiest way to get started with React is to use [this Hello World example code on CodePen](#). You don't need to install anything; you can just open it in another tab and follow along as we go through examples. If you'd rather use a local 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 examine the building blocks of React apps: elements and components. Once you master them, you can create complex apps from small reusable pieces.

thing by hand.

orm a variety of ongoing development tasks such as testing,

Angular CLI while adhering to the [Style Guide](#)

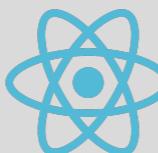
a foundation for both these documentation samples and for

2. 사용성: 얼마나 편리한가

QuickStart



Hello World



The easiest way to get started with React is to use [this JSFiddle example](#) or [this one on CodePen](#). You don't need to install anything; you can just copy and paste the code and follow along as we go through examples. If you'd rather set up your own 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 be able 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:

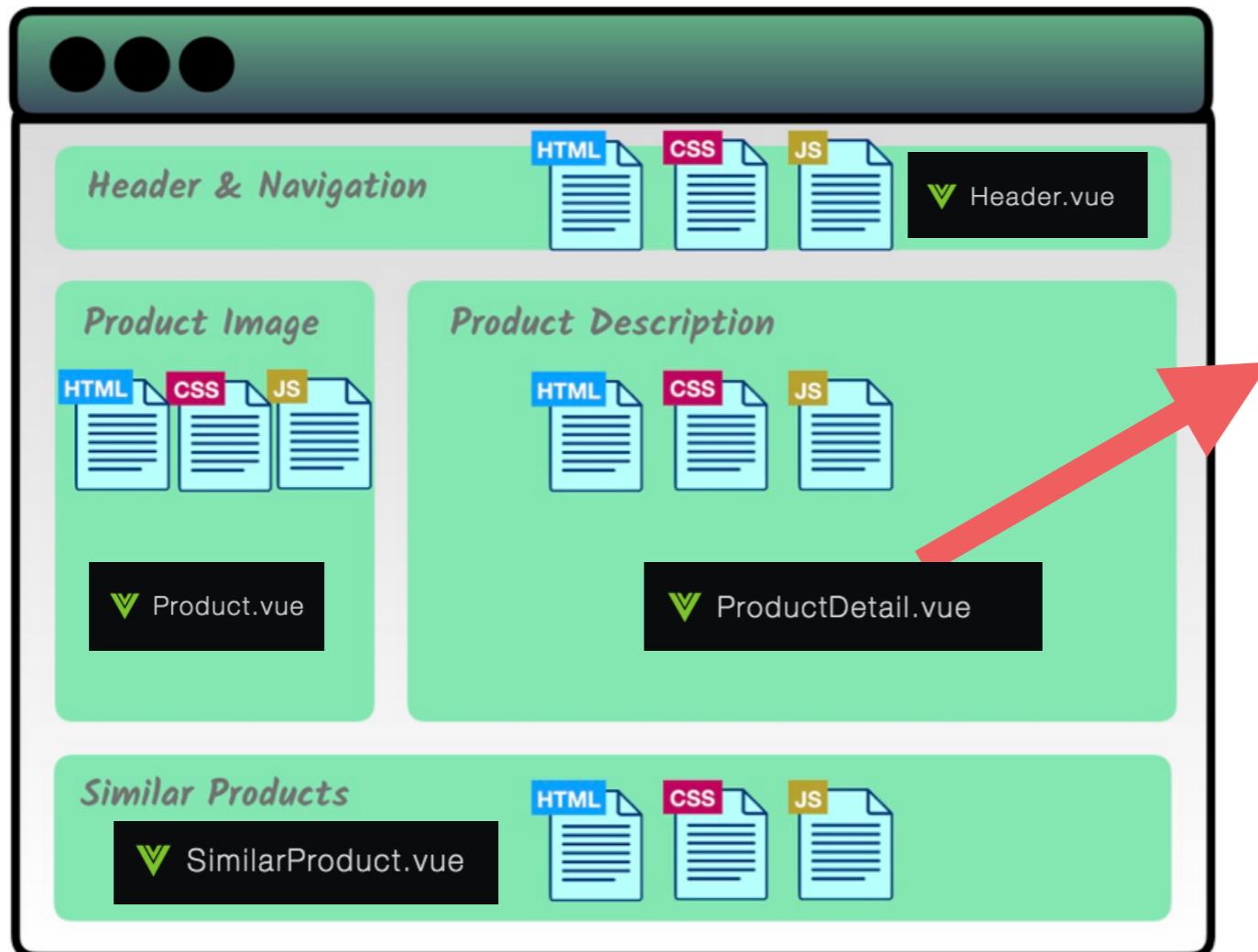
```
<script src="https://cdn.jsdelivr.net/npm/vue"></script>
```

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

2. 사용성: 얼마나 편리한가



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

<script>
export default {
  // 자바스크립트 로직
}
</script>

<style>
/* CSS 스타일 */
</style>
```

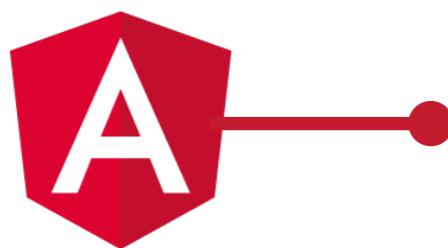
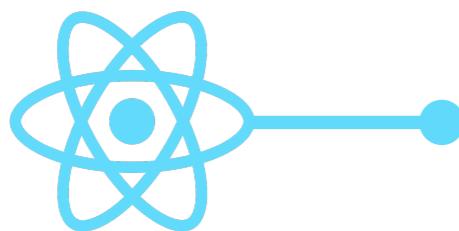
컴포넌트별 코드 관리



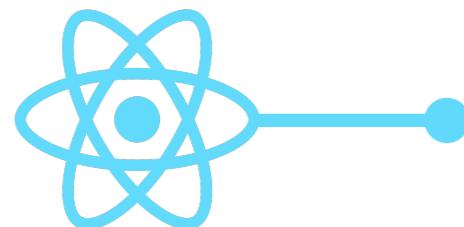
03

Angular vs React vs Vue

프레임워크별 필요 기술스택



공식 문서를 보고 Hello World 짹기



<https://reactjs.org/docs/hello-world.html>



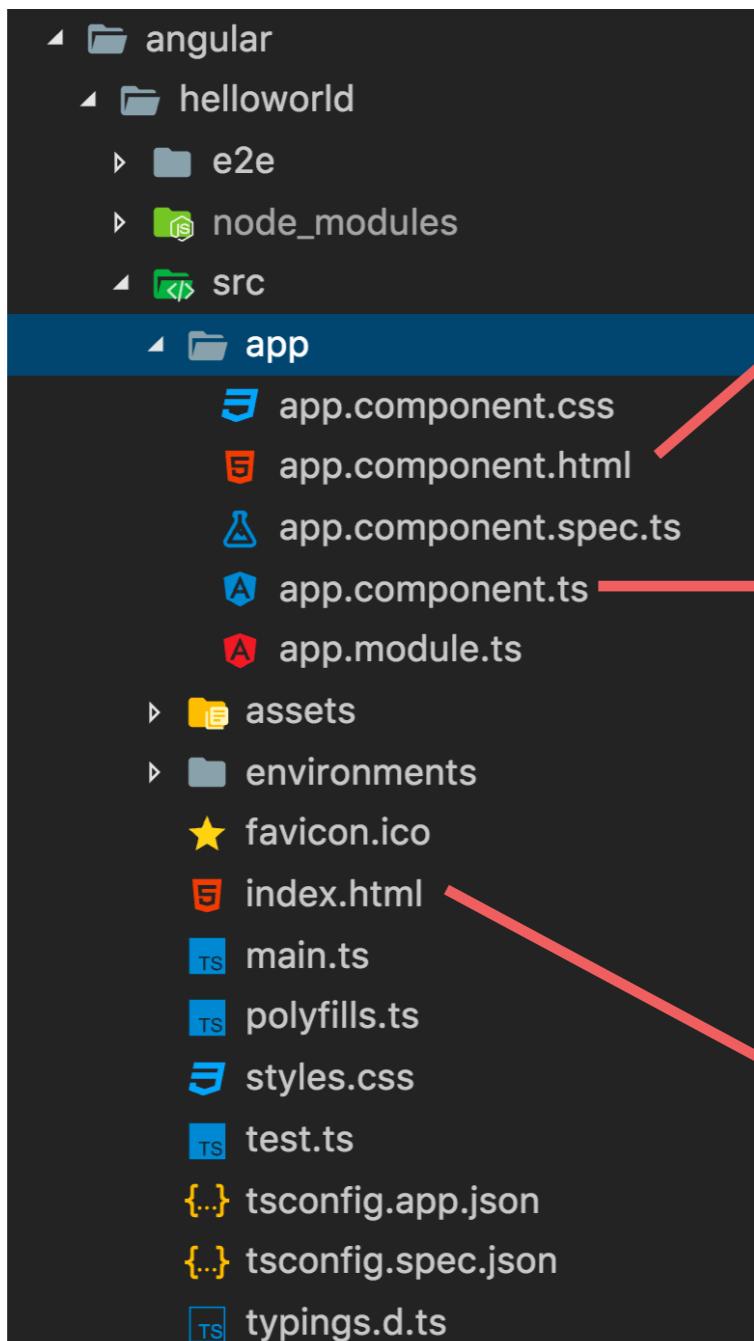
<https://angular.io/guide/quickstart>



<https://vuejs.org/v2/guide/#Getting-Started>



Angular



프로젝트 구조

app.component.html

```
<div>
  {{ message }}!
</div>
```

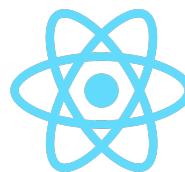
app.component.ts

```
import { Component } from '@angular/core';

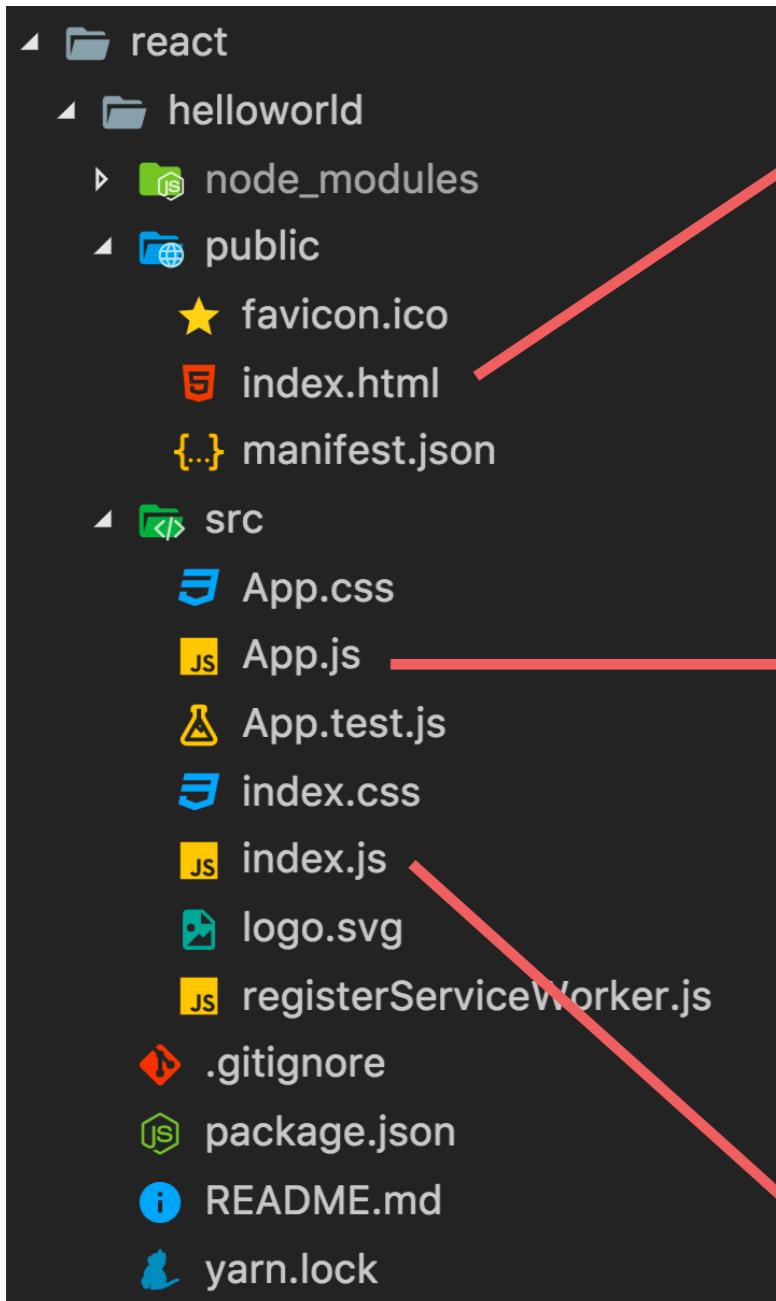
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  message = 'Hello Angular';
}
```

index.html

```
<body>
  <app-root></app-root>
</body>
```



React



프로젝트 구조

index.html

```
<div id="root"></div>
```

App.js

```
import React, { Component } from 'react';
import './App.css';

class App extends Component {
  render() {
    return (
      <div className="App">
        {this.props.message}
      </div>
    );
  }
}

export default App;
```

index.js

```
import React from 'react'; 7.1K (gzipped: 3.1K)
import ReactDOM from 'react-dom'; 93.4K (gzipped: 29.8K)
import './index.css';
import App from './App';

ReactDOM.render(<App message="Hello React"/>, document.getElementById('root'));
```



index.html

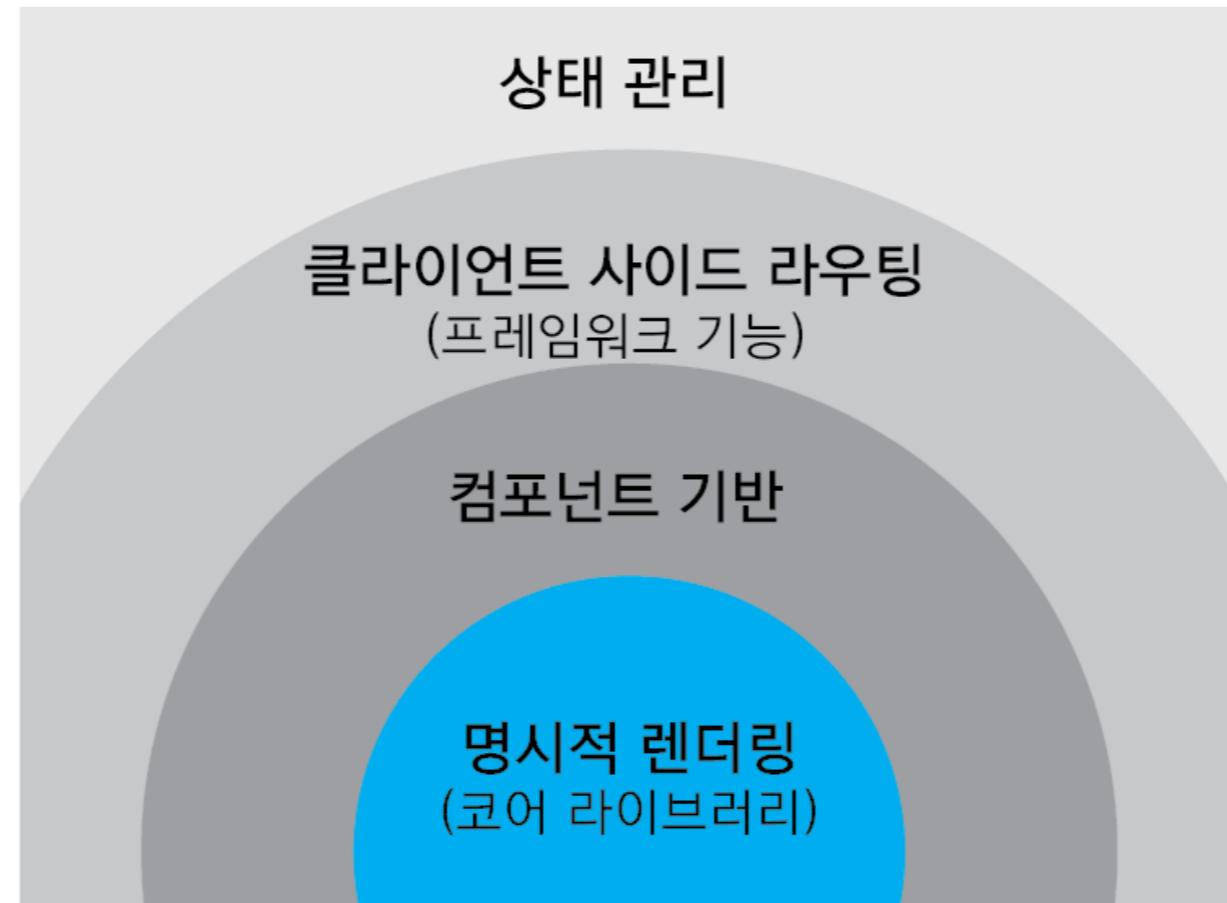
```
<html>
  <head>
    <title>Vue Sample</title>
  </head>
  <body>
    <div id="app">
      {{ message }}
    </div>

    <script src="https://cdn.jsdelivr.net/npm/vue@2.5.2/dist/vue.js"></script>
    <script>
      new Vue({
        el: '#app',
        data: {
          message: 'Hello Vue.js'
        }
      });
    </script>
  </body>
</html>
```



04 Vue.js 살펴보기

Vue.js가 뭐예요?



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

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

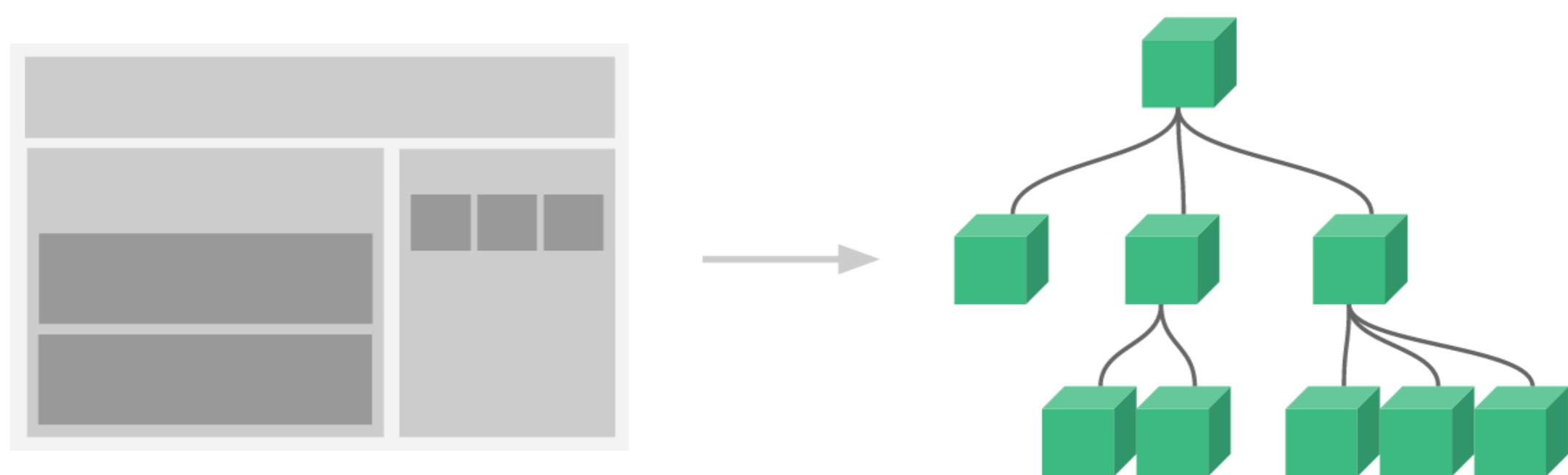
“Vue.js의 초점은 더 많은 사람들이
쉽게 웹 앱을 개발하도록 도와주는데 있다.”

-에반 유(Vue.js 창시자)



Vue.js 프레임워크 특징 3가지

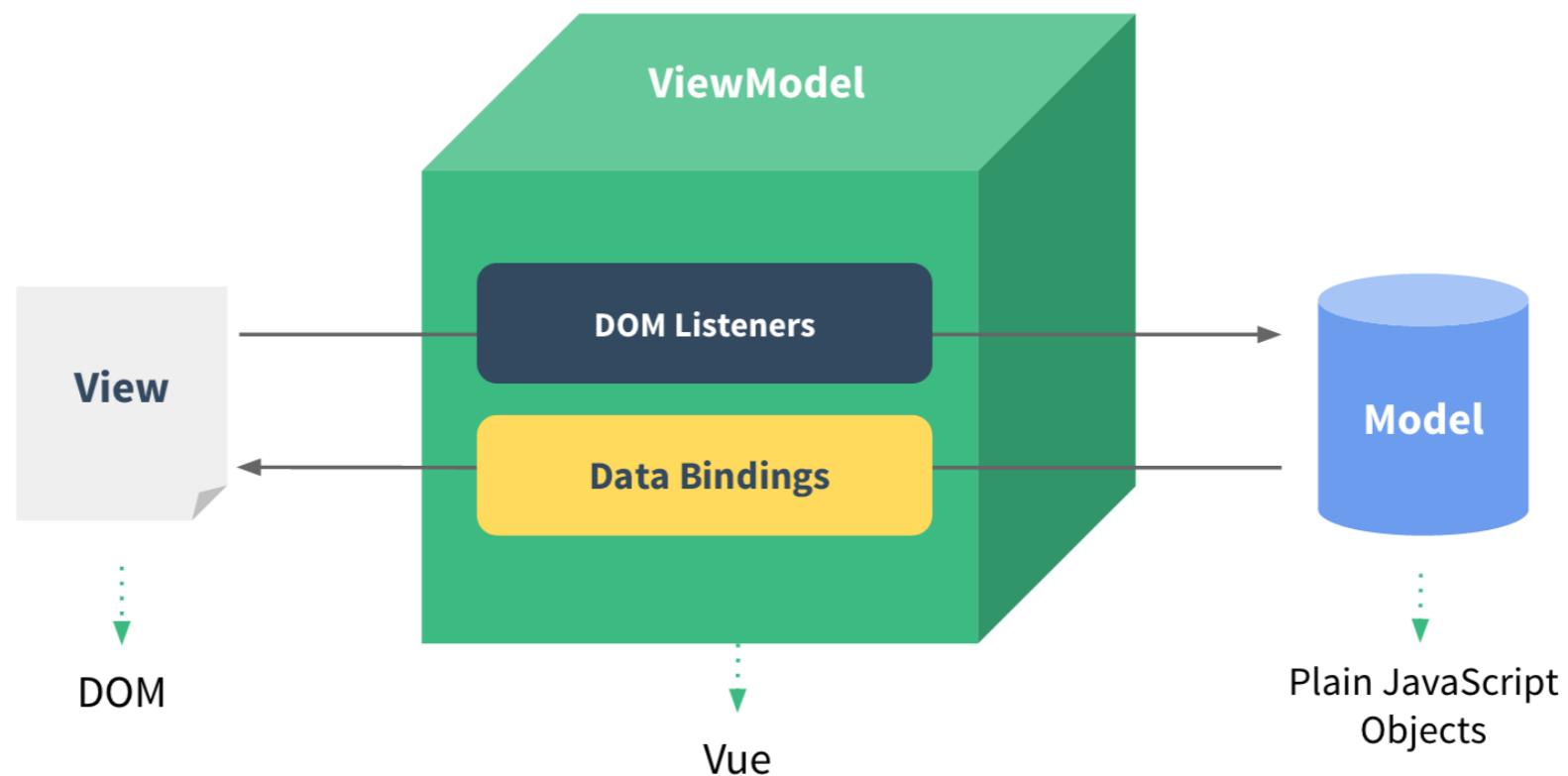
1. 컴포넌트 기반 개발 방식



화면을 여러 개의 작은 단위로 쪼개어 개발

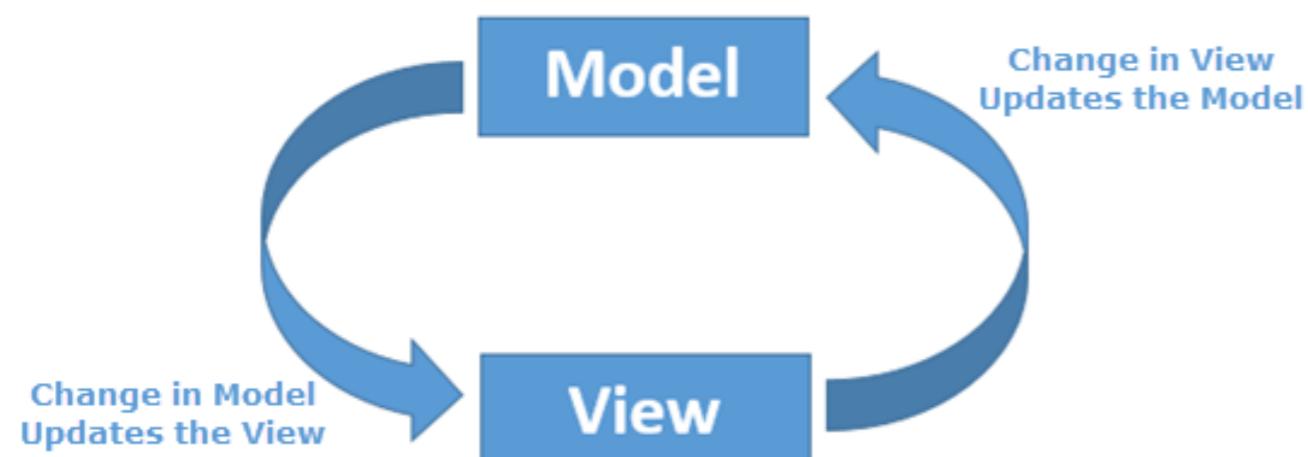
재사용성 ↑ 구현속도 ↑ 확인! 코드 가독성 ↑

2. MVVM 패턴

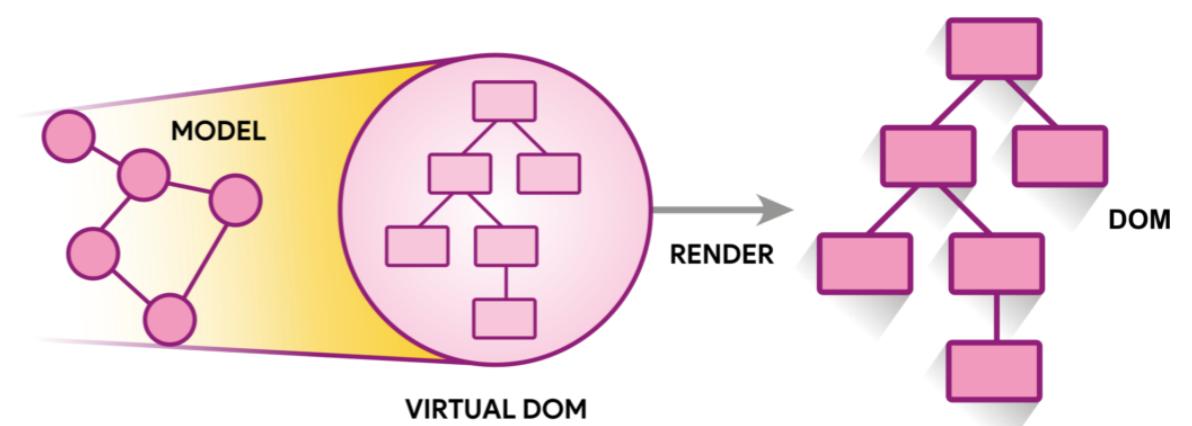


화면 UI 코드와 백엔드 데이터 처리 코드를 분리

3. 리액트와 앵귤러의 장점을 흡수



Two-way data binding

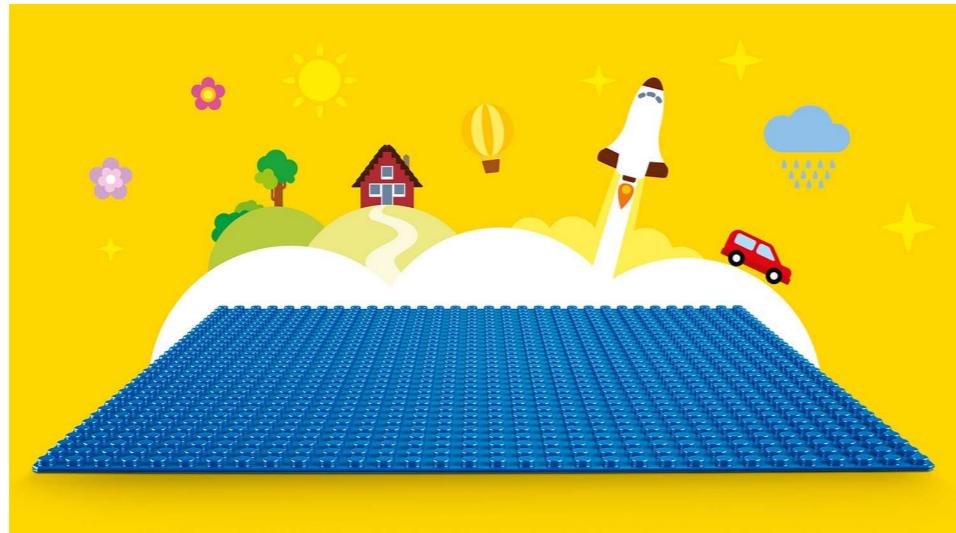


Virtual DOM

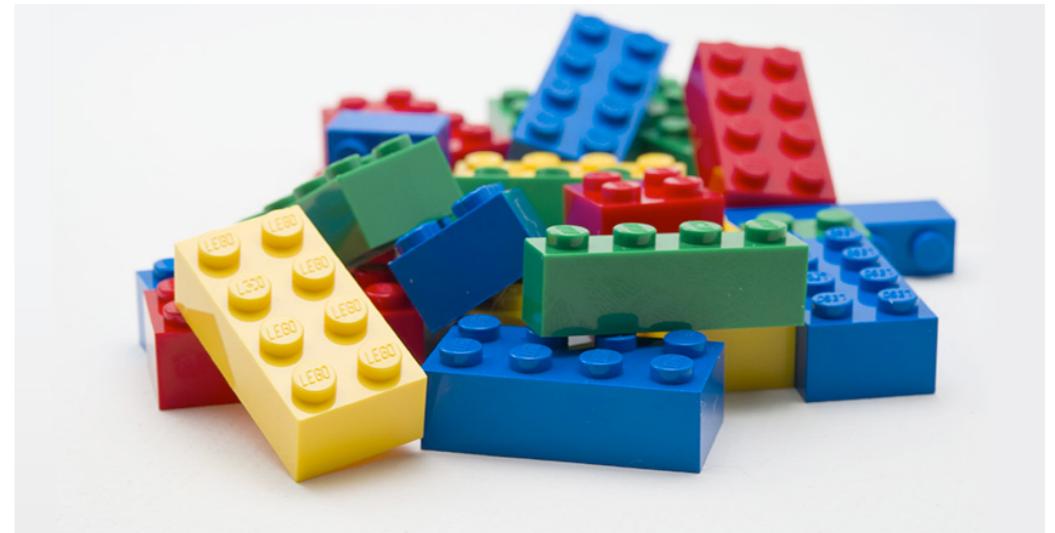


레고로 이해하는 Vue.js 구성 요소

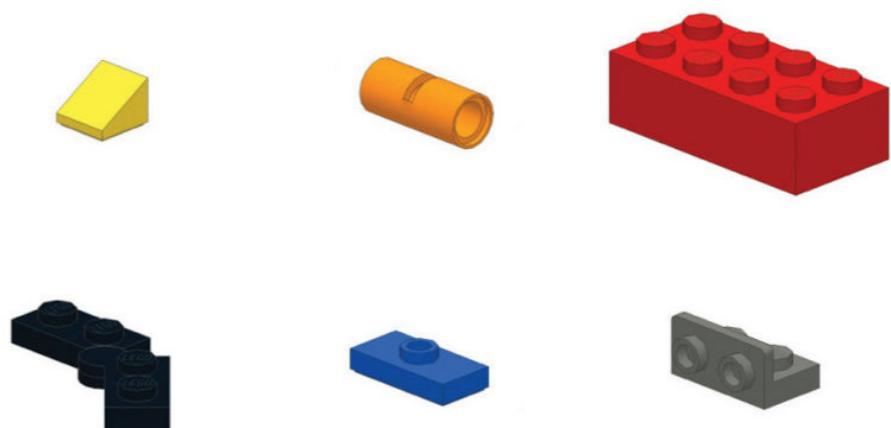
Vue.js 구성 요소



Vue Instance



Vue Component

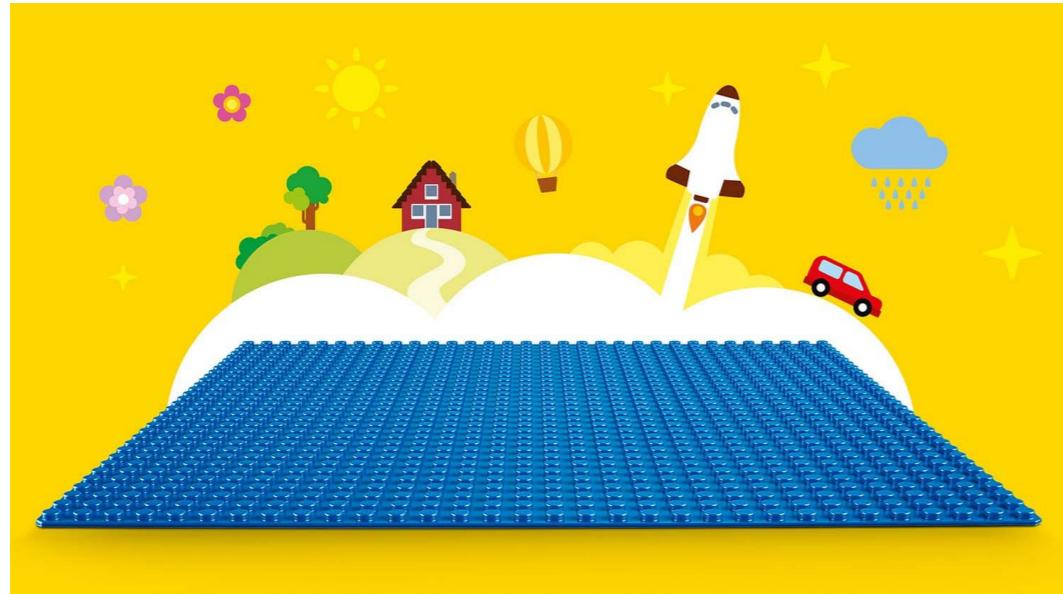


Vue Router

Vue Template

출처 : <https://www.lego.com/>

1. 뷰 인스턴스



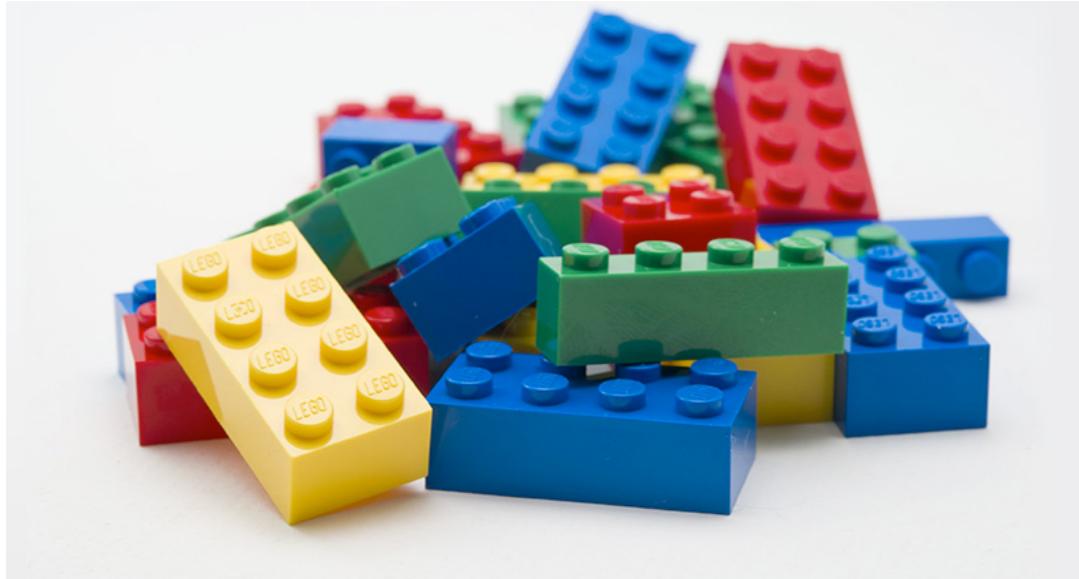
Vue Instance

```
new Vue({  
  el: // 인스턴스가 뿌려질 지점  
  template: // 인스턴스의 화면 내용  
  data: // 인스턴스가 갖는 데이터  
  methods: // 인스턴스의 이벤트 정의  
  created: // 인스턴스 라이프 사이클  
});
```

인스턴스 코드

뷰로 화면을 개발할 때 필수로 생성해야 하는 단위

2. 뷰 컴포넌트



Vue Component

```
Vue.component('my-cmp', {  
  // 인스턴스 옵션 속성과 동일  
  template: // 인스턴스의 화면 내용  
  data: // 인스턴스가 갖는 데이터  
  methods: // 인스턴스의 이벤트 정의  
  created: // 인스턴스 라이프 사이클  
});
```

컴포넌트 코드

화면을 구조적으로 설계하기 위한 요소

3. 뷰 라우터



Vue Router

```
var Main = { template: '<div>main</div>' };
var Login = { template: '<div>login</div>' };

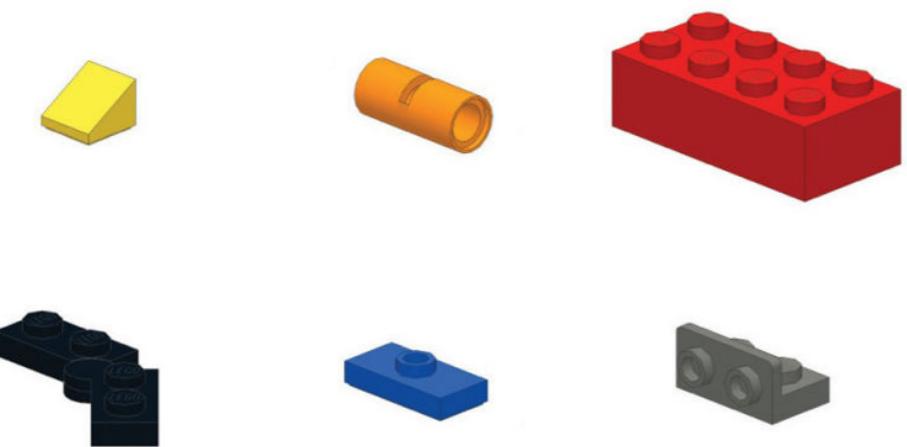
var router = new VueRouter({
  routes: [
    { path: '/main', component: Main },
    { path: '/login', component: Login }
  ]
});

var app = new Vue({
  el: '#app',
  router: router
});
```

라우터 코드

여러 개의 화면 간에 이동하는 방법

4. 뷰 템플릿



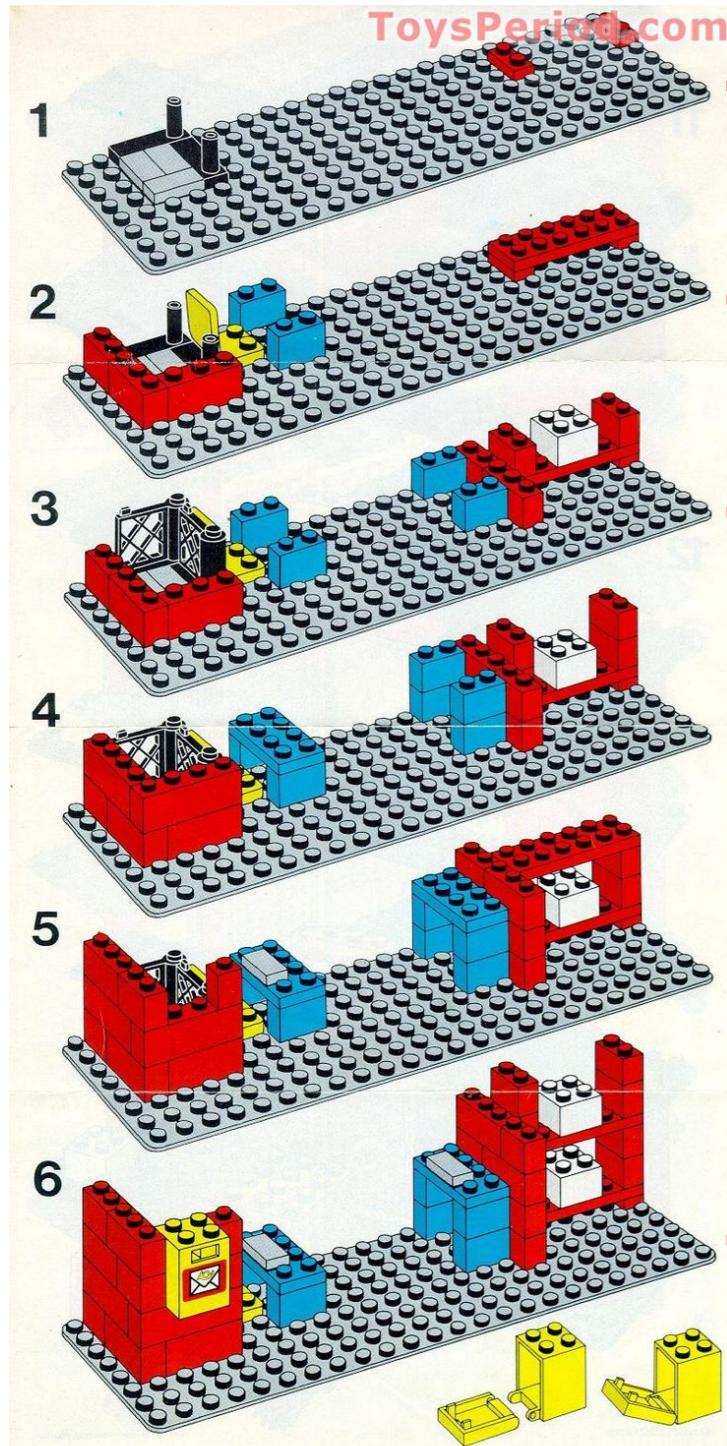
Vue Template

```
 {{ message }},  
 {{ message.split('').reverse().join('') }}  
 <p v-bind:id="uid">인스턴스의 값과 연결</p>  
 <a v-if="seen">seen 값에 따라 표시 여부 결정</a>  
 <ul v-for="item in items">{{ item.name }}</ul>  
 <button v-on:click="pAlert">클릭하면 경고창 표시</button>
```

템플릿 코드

화면을 구체적으로 꾸미는 방법 & 문법

레고를 조립하는 방법

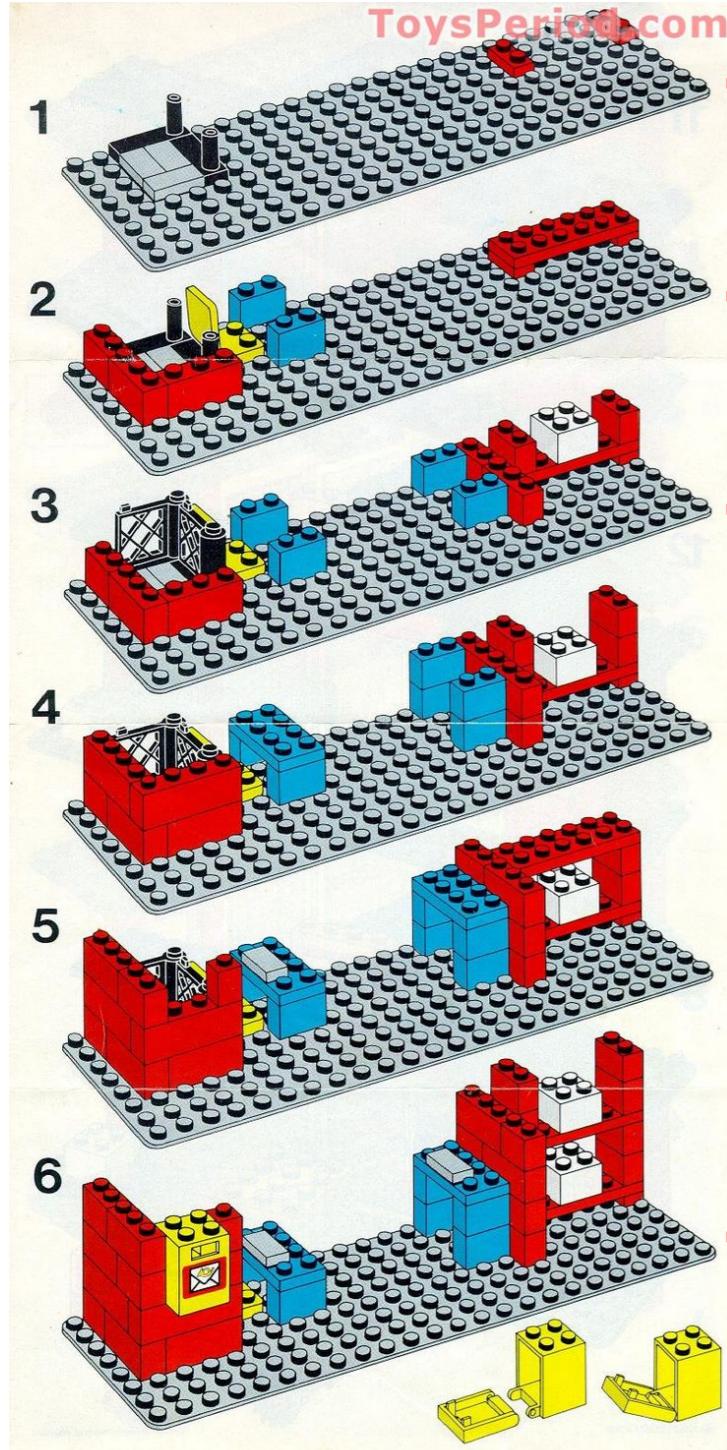


조립 판을 설치

레고블록을 각 위치에 설치

설치한 레고블록 꾸미기

레고를 조립하는 방법



조립 판을 설치

뷰 인스턴스 생성

레고블록을 각 위치에 설치

뷰 컴포넌트 설계 및 생성

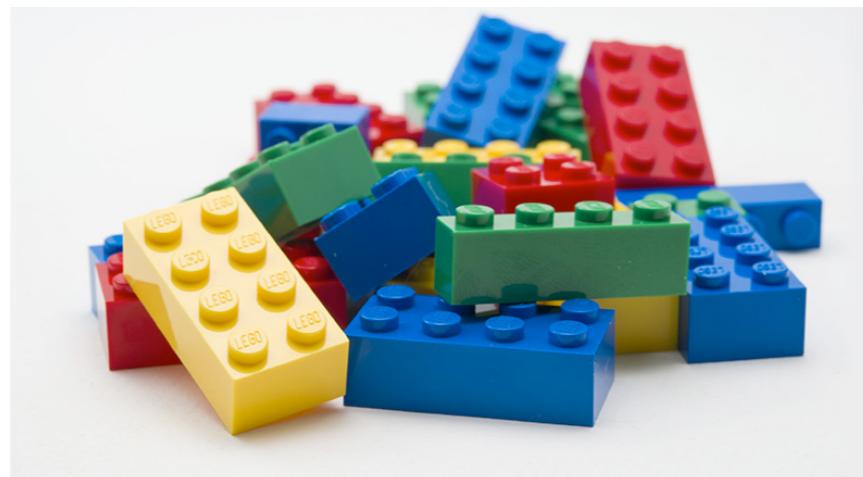
설치한 레고블록 꾸미기

뷰 컴포넌트 내용 구현

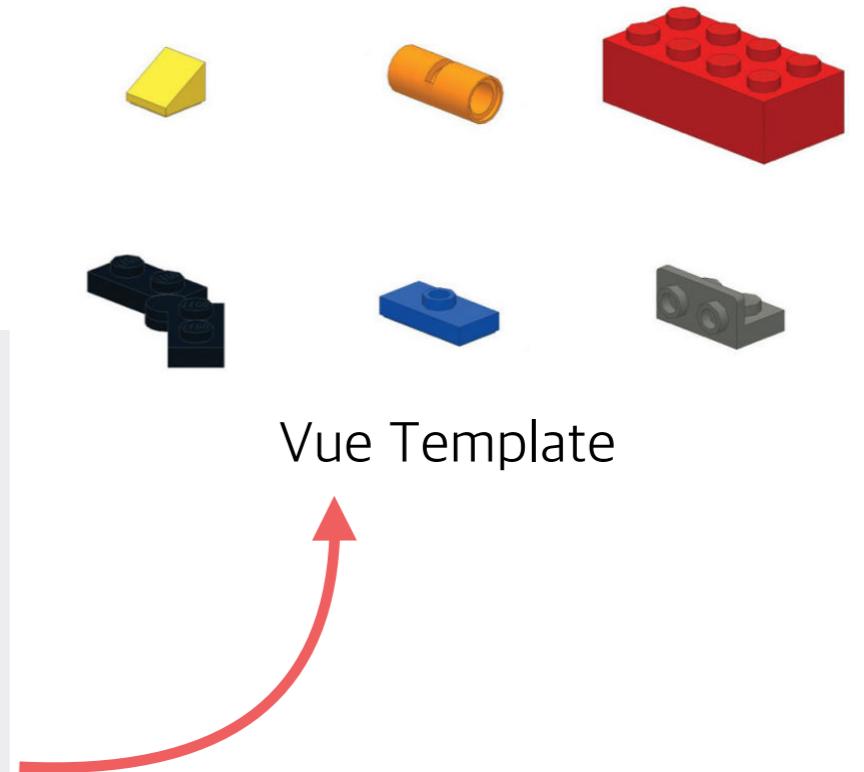
Vue.js 구현 절차



Vue Instance



Vue Components

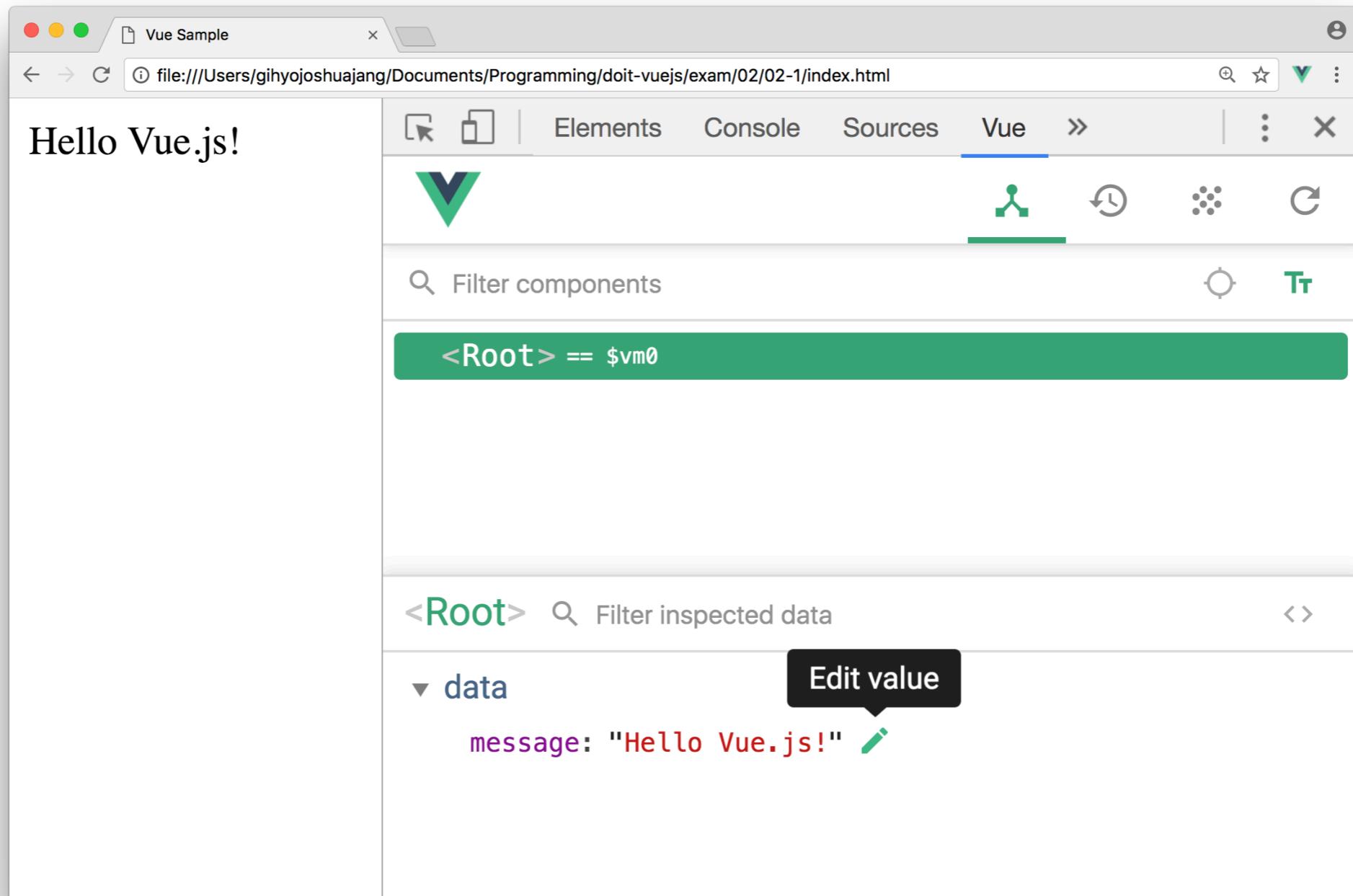


Vue Template



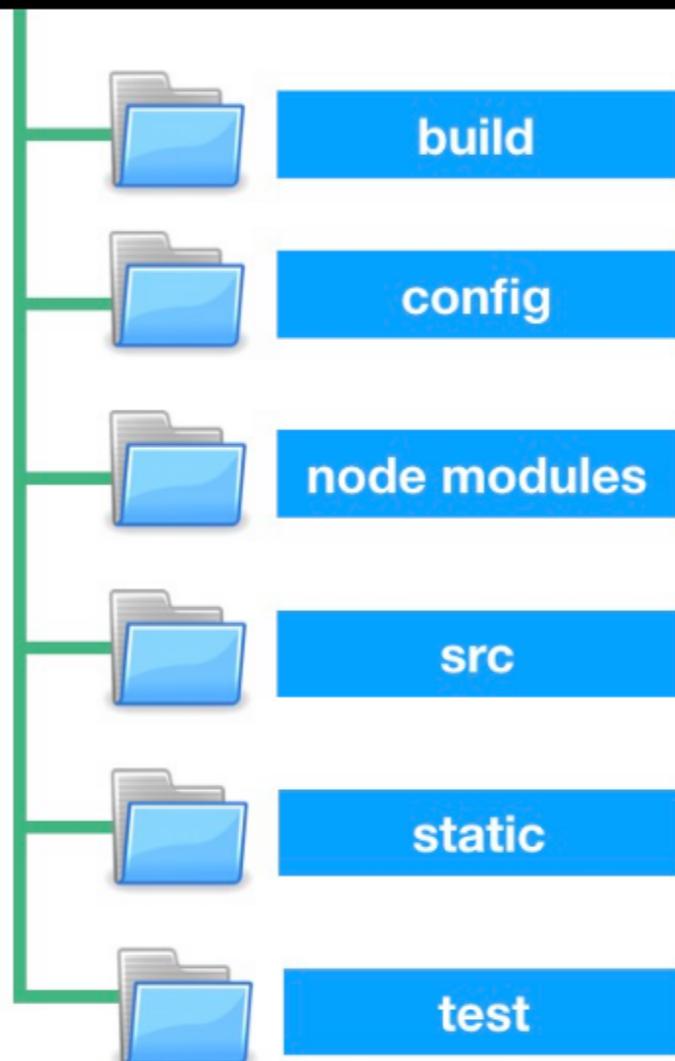
Vue.js 개발을 도와주는 도구

1. Vue.js 개발자 도구



2. Vue CLI 도구

```
$ vue init webpack my-project
```





02

Vue.js 동작원리와 자가 학습 방법

Subjects

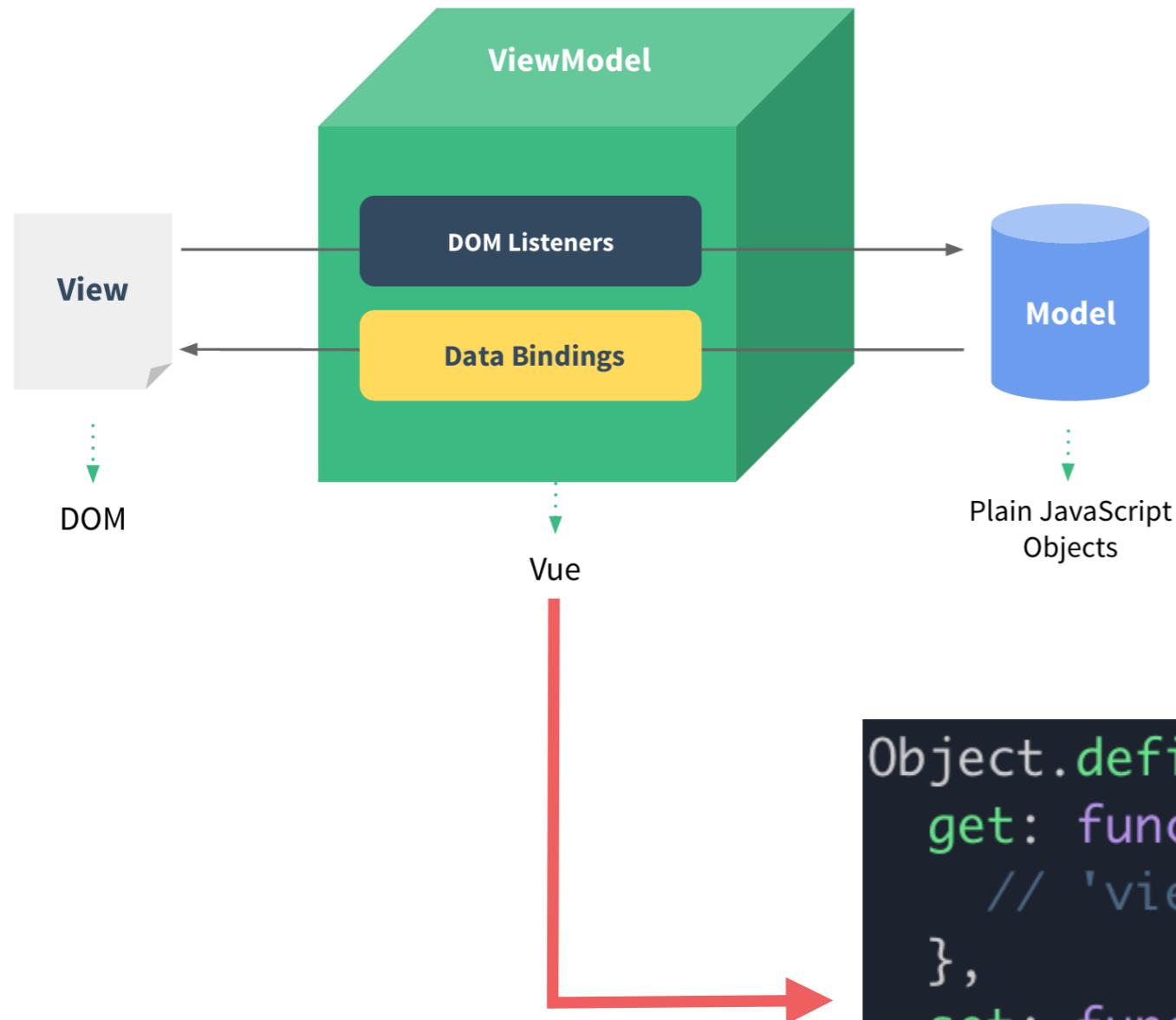
1. Vue.js 동작 원리 알아보기
2. Vue.js vs jQuery
3. Vue.js와 프런트엔드 개발 학습 방법
4. Vue.js 참고 자료



01

Vue.js는 어떻게 동작할까

Vue.js Reactivity I

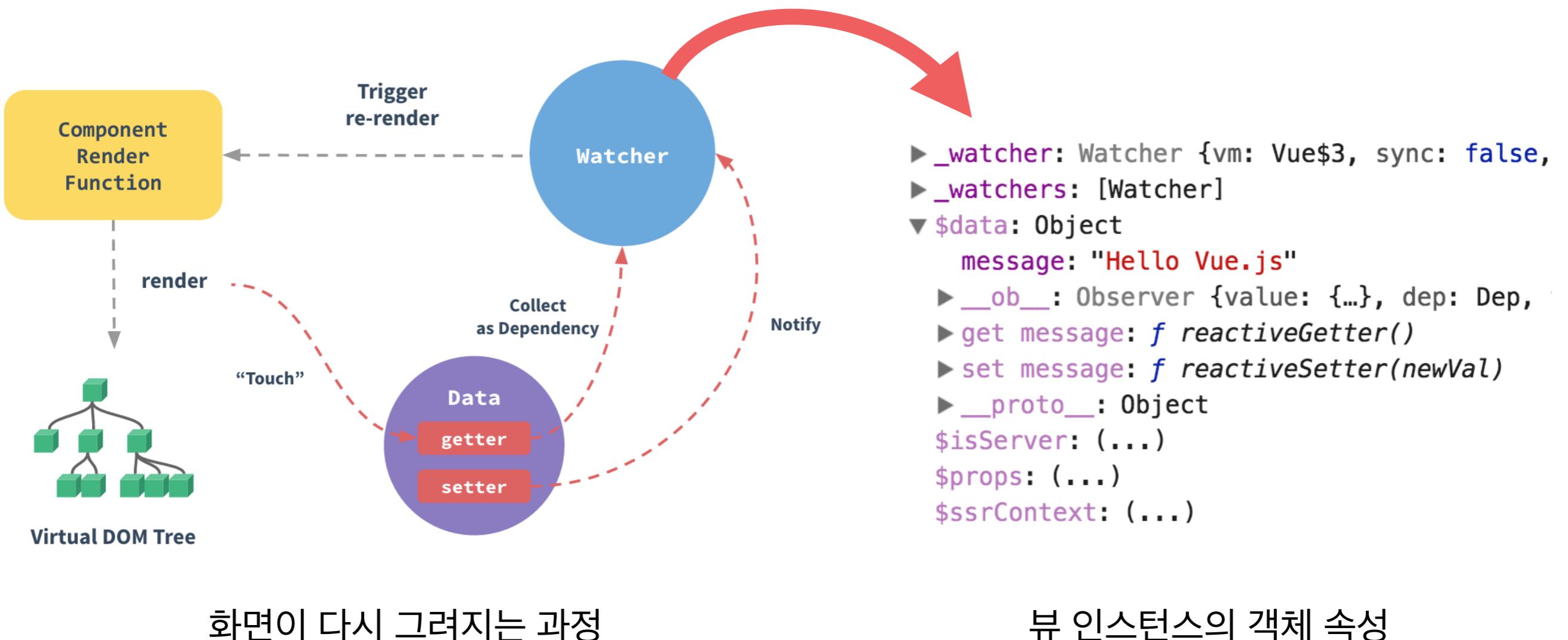


MVVM 패턴에서의 Vue.js

```
Object.defineProperty(viewModel, 'model', {  
  get: function() {  
    // 'viewModel.model'을 접근할 때의 동작 정의  
  },  
  set: function(newValue) {  
    // 'viewModel.model'에 값을 할당할 때 동작 정의  
  }  
});
```

특정 속성에 반응성을 주입하는 코드

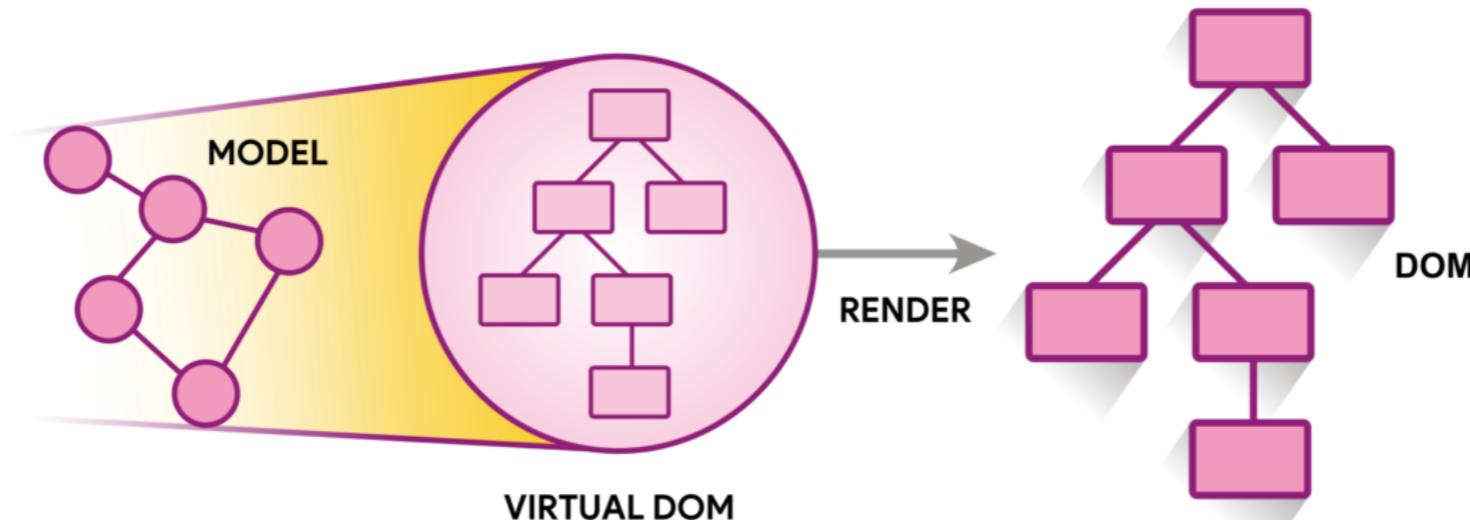
Vue.js Reactivity II



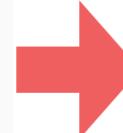
화면이 다시 그려지는 과정

뷰 인스턴스의 객체 속성

Vue.js Template Compilation

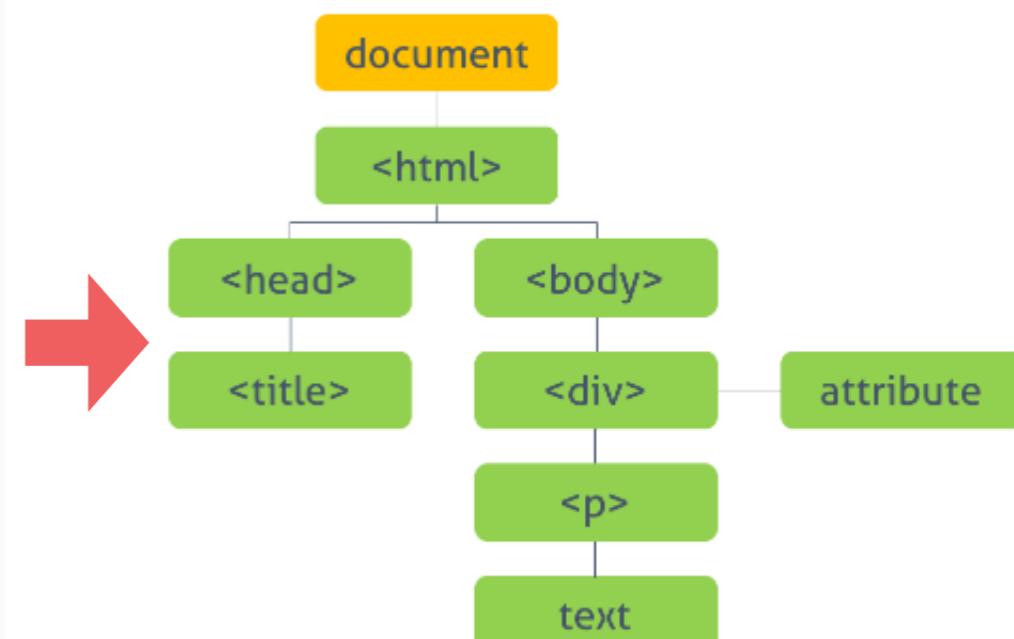


```
<template>
  <div>
    <section>
      Do it! Vue.js 입문
    </section>
    <p>
      출간 기념 공개 세미나
    </p>
  </div>
</template>
```



```
render: function (createElement)
{
  return createElement('div', [
    createElement(
      'section',
      'Do it! Vue.js 입문'),
    createElement(
      'p',
      '출간 기념 공개 세미나')
  ]);
}
```

render() 함수



Vue.js 템플릿 코드

DOM 트리에 요소 추가



02

Vue.js vs jQuery

Vue.js vs jQuery

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>
```

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>
```

jQuery로 할 수 있는 건 Vue.js로도 할 수 있다

Vue.js vs jQuery

	jQuery	Vue.js
구분	자바스크립트 라이브러리	점진적인 프레임워크 (From View Library to Framework)
동작 원리	실제 DOM Tree에 접근하여 화면 요소를 추가, 삭제, 변경	가상돔(Virtual DOM)으로 화면 요소를 변경 및 조작하고 최종 결과물을 DOM Tree에 반영
개발 생산성	렌더링 과정을 이해한 코딩 필요 기존 코드 복사 붙여넣기	뷰 개발자 도구, 뷰 CLI 도구 지원 컴포넌트 기반 코드 재사용
웹 성능 최적화	최신 웹 기술을 수용하기 위해 웹 자동화 도구 및 모듈 번들러를 별도로 구성해야 함	뷰 CLI 도구에서 제공하는 프로젝트 템플릿으로 쉽게 기타 도구 사용 및 확장 가능



03

Vue.js와 프런트엔드 학습 방법

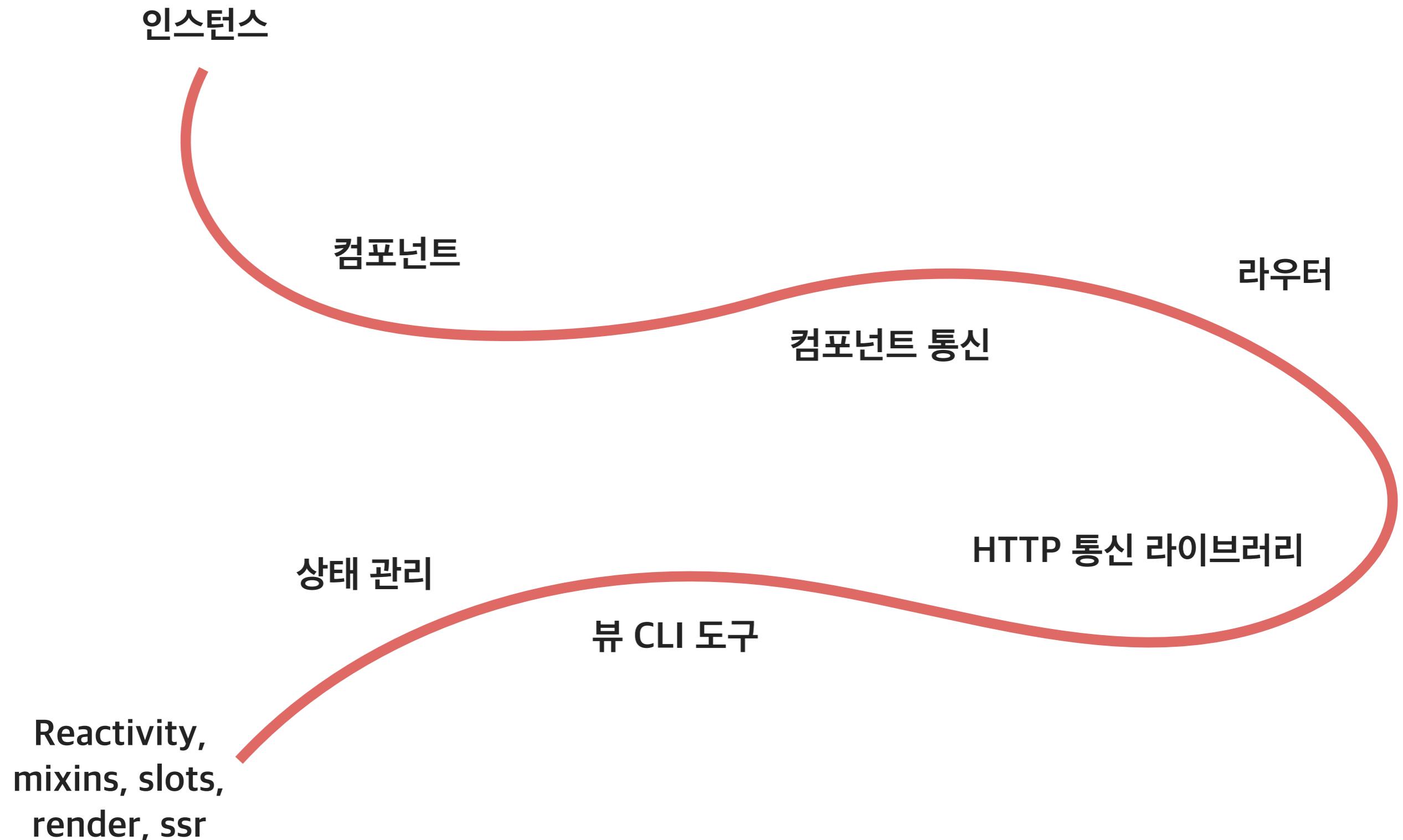


프런트엔드 Learning Path





Vue.js Learning Path





04 Vue.js 참고 자료



Vue.js 참고 자료

학습 자료

1. [Vue.js 공식 문서](#)
2. [Vue.js Feed](#)
3. [Vue.js Developers](#)

강의 목록

1. [Vue.js Complete Guide\(영어\)](#)
2. [누구나 다루기 쉬운 Vue.js\(한글\)](#)
3. [실습 UI 개발로 배워보는 순수 JS 와 VueJS 개발\(한글\)](#)

프로젝트 및 코드

1. [Awesome Vue 리포지토리](#)
2. [Vue Material](#)
3. [Scotch Vue.js 튜토리얼](#)

통계 및 기타

1. [State of Vue.js](#)
2. [Vue.js 팟캐스트](#)
3. [Vue.js Mastery](#)



03

Vue.js로 구현하는 Todo App

Subjects

1. 개발환경 설정
2. 애플리케이션 설계 - 기술 스택
3. 애플리케이션 설계 - UI

개발환경 설정

1. Node.js 설치



2. Vue CLI 설치



3. Vue 개발자 도구 설치



4. VisualStudio Code 설치



5. Git 설치

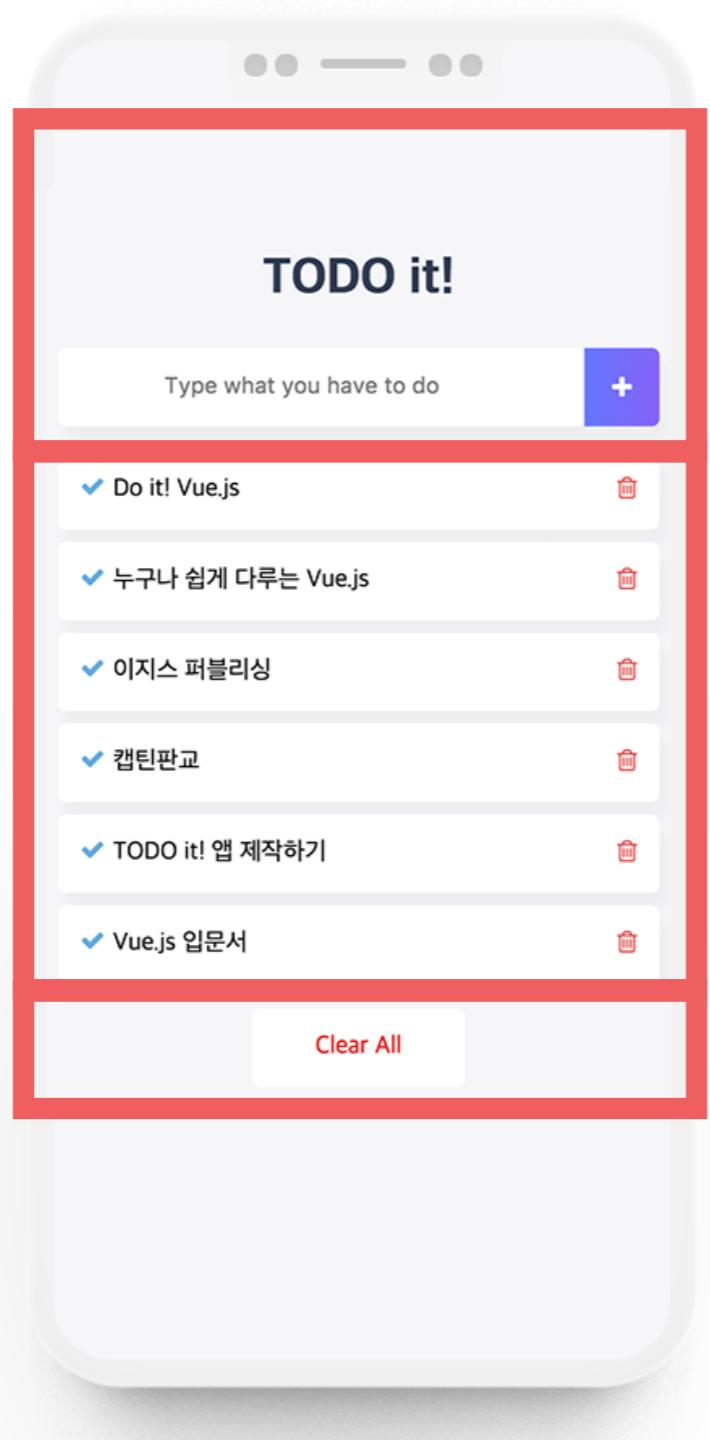




애플리케이션 설계 - 기술스택



애플리케이션 설계 - UI



할 일을 입력하는 인풋 박스

추가된 할 일이 표시되는 목록

등록된 할 일 모두 삭제



같이 만들어 볼까요?



감사합니다