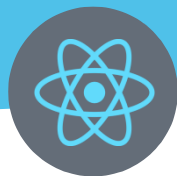




# 프론트엔드 프레임워크 비교 분석

- React, Vue, Angular, Svelte



조세나라

# 목차

1. 속성
2. 특징
3. 코딩스타일
4. 컴포넌트형태
5. DOM
6. 러닝 커브
7. 번들러
8. TypeScript 지원
9. 상태관리
10. SPA 라우터
11. SSR
12. 결론

# 속성

React	라이브러리
Vue	프로그레시브 프레임워크 <sup>[1]</sup>
Angular	프레임워크
Svelte	컴파일러

[1] 점진적으로 적용이 가능한 프레임워크

# 특징

React	라이브러리 기반의 프레임워크, 강력한 서드파티 모듈에 의존.
Vue	보다 친숙한 개발 경험을 제공하는 프레임워크.
Angular	모든 것을 제공하는 진정한 의미의 프레임워크.[1]
Svelte	컴파일러[2]

[1] 라우팅, CLI, 설정, 번들러 등의 기초 설정 뿐만 아니라 UI와 코딩스타일까지 다 프레임워크가 제공.

[2] 실질적으로 브라우저에서 작동 시 사용되는 라이브러리가 없음.

# 코딩스타일

React	HTML 마크업에 템플릿 문법을 사용하는 것이 아닌 XML 포맷 형식의 템플릿을 JS로만 작성.
Vue	HTML 마크업 기반의 문법 사용. HTML, JS, CSS 코드 영역을 분리해서 작성함, 가시성이 좋고 보다 친숙함.
Angular	타입스크립트를 통한 앵귤러 만의 문법으로 작성.
Svelte	Vue와 똑같이 HTML 마크업 기반의 문법 사용하며 HTML, JS, CSS 코드를 분리하고 마크업 방면에서 보다 더 획기적으로 간단하게 사용됨.

# 컴포넌트 형태

React	.jsx 파일의 리액트 컴포넌트 작성.
Vue	.vue 파일의 뷰 컴포넌트 작성.
Angular	데코레이터 문법 <sup>[1]</sup> 을 활용한 ts파일로 짜여진 앵귤러 컴포넌트 작성.
Svelte	.svelte 파일의 스벨트 컴포넌트 작성.

[1] Java의 어노테이션과 매우 흡사함.

# 컴포넌트란?



재사용이 가능하고 캡슐화한 HTML 엘리먼트의 단위.

HTML 엘리먼트가 작동을 하기 위한 HTML 코드와 JS를 합쳐 하나의 커스텀 엘리먼트가 구성되며  
일반적으로 프레임워크에서 독자적인 파일 확장자를  
이용하여 이를 만들어 제공한다.

사실상 프론트엔드 프레임워크의 핵심 요소 중 하나.

# DOM

React	Vitrual DOM
Vue	Vitrual DOM
Angular	Incremental DOM + Shadow DOM
Svelte	Real DOM <sup>[1]</sup>

[1] 다른 프레임워크와 달리 실제 DOM을 사용함에도 불구하고 효율적인 작동원리로 훨씬 빠르게 작동함.



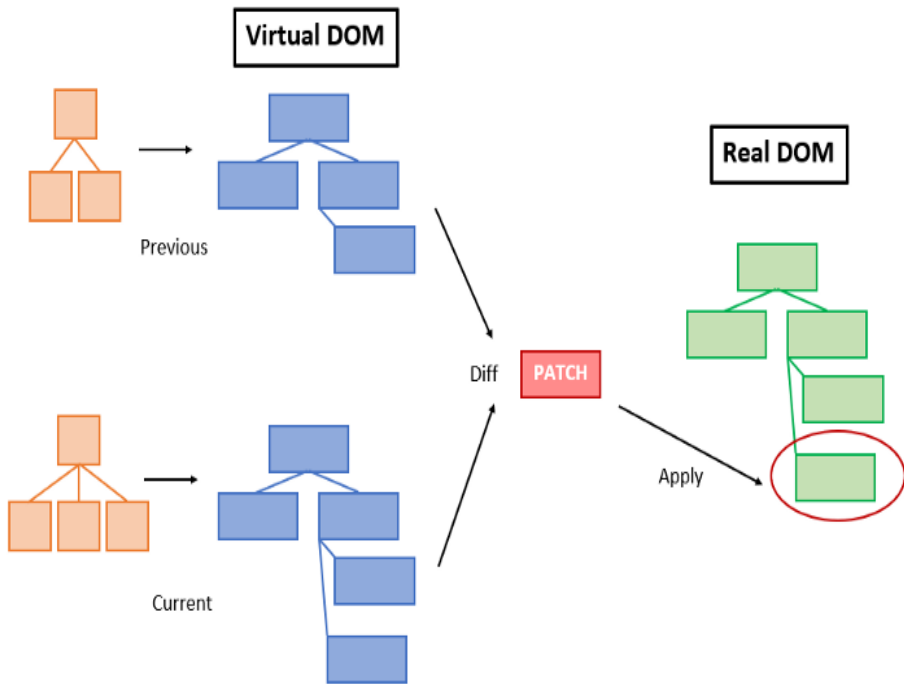
# Shadow DOM이란?



## Shadow DOM – 독립적인 DOM

DOM을 캡슐화하여 하나의 페이지에서 내부적으로 작동하며 글로벌 한 DOM 접근을 허용하지 않고 내부적으로 독립적으로 동작하는 영역을 얻는 DOM.

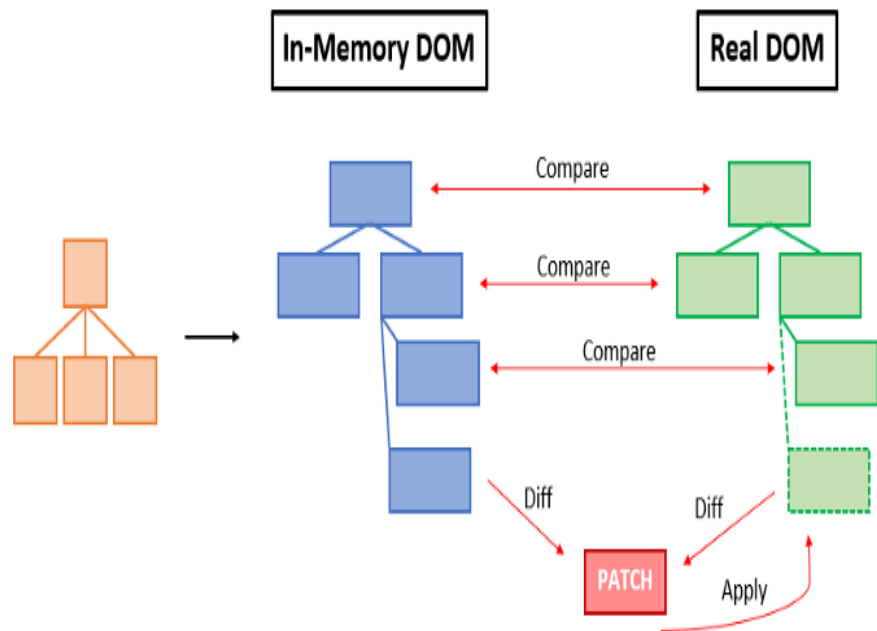
# Vitrual DOM



## Vitrual DOM의 작동 원리

1. UI가 변경되면 전체 UI를 버추얼돔으로 렌더링.
2. 현재 버추얼돔과 이전 버추얼돔을 비교해 차이를 계산한다.
3. 변경된 부분만 리얼 돔에 반영한다.

# Incremental DOM



## Incremental DOM의 작동 원리

1. 인 메모리에 리얼 돔을 올려놓음.
2. 리얼 돔과 비교하여 변경점을 찾음.
3. 변경된 부분을 즉각 리얼 돔에 반영함.

\* 속도는 버추얼 돔보단 느리지만 메모리를 효율적으로 사용함. -> 모바일에 더 적합

\* 앵귤러에서 인크리멘탈 돔을 채용한 이유는 앵귤러의 컨셉인 웹과 모바일을 앵귤러라는 프레임 워크로 다 개발하고자 하기 때문.

# 러닝 커브

React	방대한 서드파티 모듈들 덕에 높음.
Vue	친숙한 구조 덕에 낮은 편.
Angular	TypeScript의 의무화와 모든 프레임워크 구성요소를 익히기 위해선 높음. <sup>[1]</sup>
Svelte	Vue 보다 간결한 개발 경험을 제공하므로 가장 낮은 편.

[1] 단 프레임워크 설정과 관련된 부분은 제공되므로 개발 입문자에게는 타 프레임워크에 비해 비교적 낮은 편.

# 번들러

React	Webpack에서 Babel이 처리.
Vue	Vite(rollup <sup>[1]</sup> ) 혹은 Webpack에서 vue-loader를 달아서 씀.
Angular	Webpack이긴 하지만 기본 설정 스펙이므로 알 필요가 없음.
Svelte	Vite를 쓰지만 안정성으로 인해 Webpack을 사용.

[1] 최근에 떠오르는 번들러. 웹팩보다 빠른 속도를 자랑함.

# TypeScript 지원

React	오직 JS만을 사용한 코딩을 지향하므로 활용 극대화 가능.
Vue	타입스크립트와 그 모듈 사용을 위해 코드 변경이 필요함. Vue 3.x 미만일 경우 호환성이 낮음.
Angular	기본 채택 언어.
Svelte	쓸 수는 있지만 권장 사항은 아님.

# 상태관리

React	Vitrual DOM을 사용하기 때문에 상태관리가 필요함. Redux <sup>[1]</sup> 를 사용하다 최근 React hooks이 나오며 종결.
Vue	Vitrual DOM을 사용하기 때문에 필요함. vuex를 사용하여 관리하며 최근엔 TypeScript 활용이 극대화되는 Pinia로 넘어가는 중. <sup>[2]</sup>
Angular	Ngrx나 Redux를 사용함.
Svelte	Real DOM을 쓰지만 작동원리에 따른 상태관리가 적용되어 있으며 심화적인 경우 store 등 내부 기능 제공.

[1] React의 상태관리를 위해 개발된 써드파티 모듈. React 뿐만 아니라 대부분의 프레임워크에서 사용 가능.

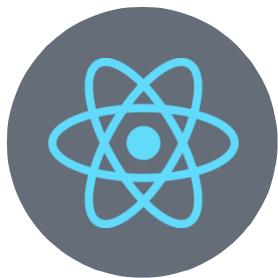
[2] vuex는 유지보수 단계로 접어들었고 pinia가 주력으로 개발이 진행되는 중.

# SPA 라우터

React	<code>react-router</code> 설치하거나 Next.js나 Gatsby와 같은 써드파티 프레임워크에서 제공.
Vue	<code>vue-router</code> 설치하거나 Nuxt.js와 같은 써드파티 프레임워크에서 제공.
Angular	프레임워크 기본 제공 스펙.
Svelte	<code>svelte-spa-router</code> 를 설치하여 사용.



# 결론



React

대규모  
웹 어플리케이션에  
적합



VUE

대규모, 소규모  
모두 적합



Angular

대규모 웹앱과  
모바일 웹앱 등에  
적합



Svelte

소규모  
웹 어플리케이션에  
적합