

브라우저 렌더링 과정 뽐내기

브라우저는 화면을 어떻게 그릴까?

왜 중요할까?

- 면접 빈출 질문이다.
- 렌더링 성능 최적화를 하기 위해서
 - 성능 개선하면 포트폴리오에 적을 만 하다.

“면접 빈출 질문이다”의 관점

- 프론트엔드 직무 지원 시 거의 **무조건 나오는 질문**이다.
 - 1분 자기소개 해주세요 급이다. 모르면 광탈하는 질문이다.
- 면접에서는 BFS방식 보다는 **DFS방식으로 대답**하는 것이 중요하다.
- 우리가 가고 싶어하는 회사는 꼬리질문을 한다.
 - Ex) 네이버는 정말로 모를 때 까지 꼬리질문을 한다. (살려주세요..)
- 나올지 안 나올지 모르는 주제를 깊게 아는 것 보다,
나오는 주제를 다른 사람보다 깊게 아는 전략을 사용해보자.

“렌더링 성능 최적화를 하기 위해서”의 관점



As page load time goes from:

1s to 3s the probability of bounce **increases 32%**

1s to 5s the probability of bounce **increases 90%**

1s to 6s the probability of bounce **increases 106%**

1s to 10s the probability of bounce **increases 123%**

- 구글피셜

1초에서 3초로 증가하면,

페이지 이탈률이 32%로 증가

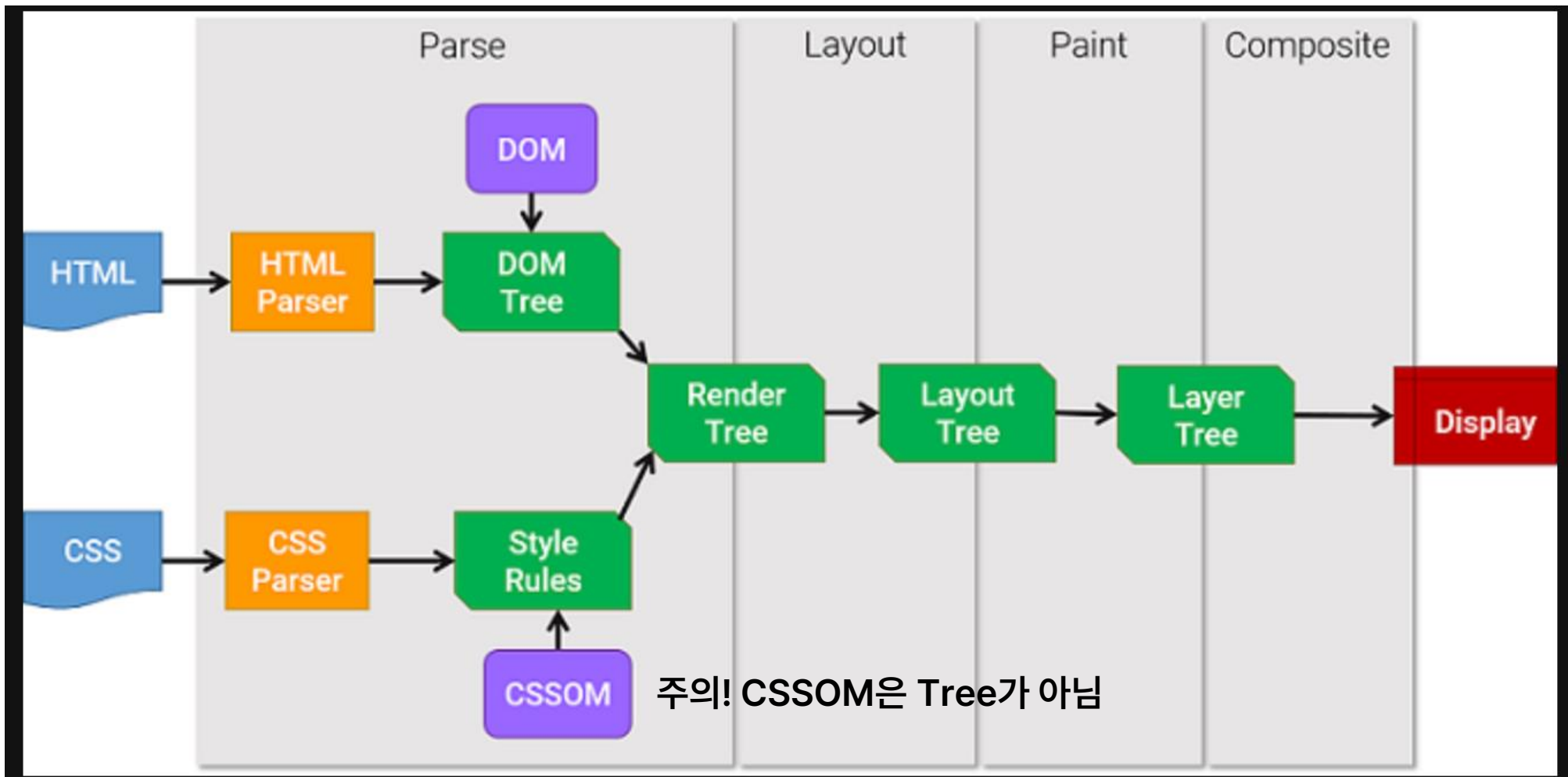
1초에서 10초가 되면,

무려 페이지 이탈률이 123%

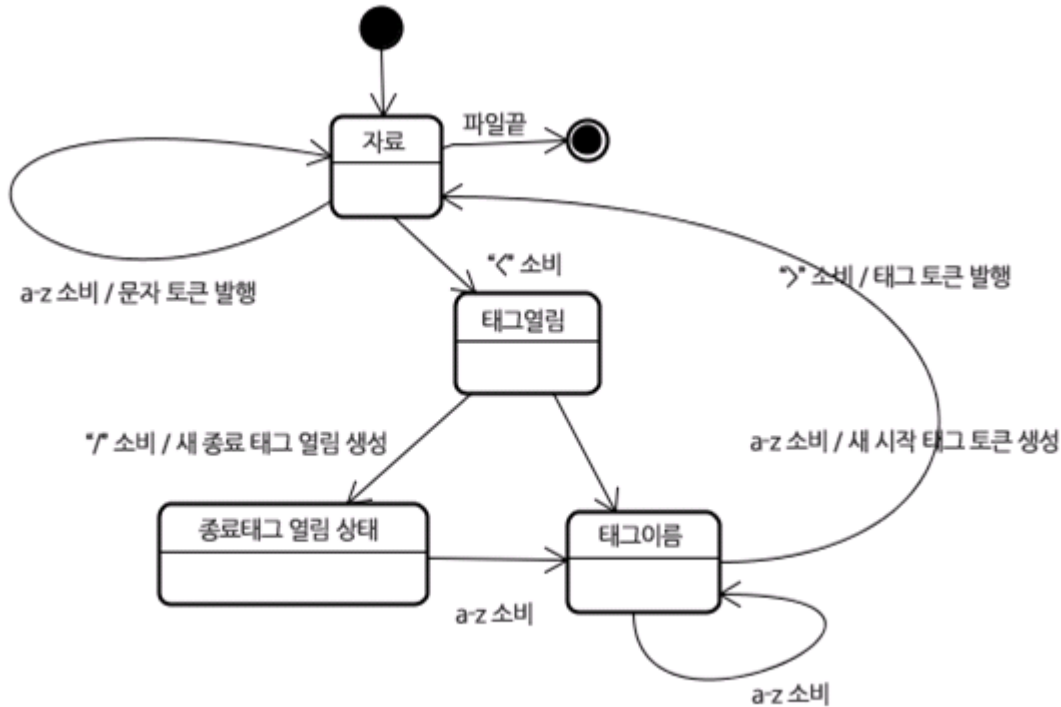


크로미움을 사용하는 브라우저를 바탕으로 제작되었습니다.
크로미움은 Blink엔진을 사용합니다.





파싱



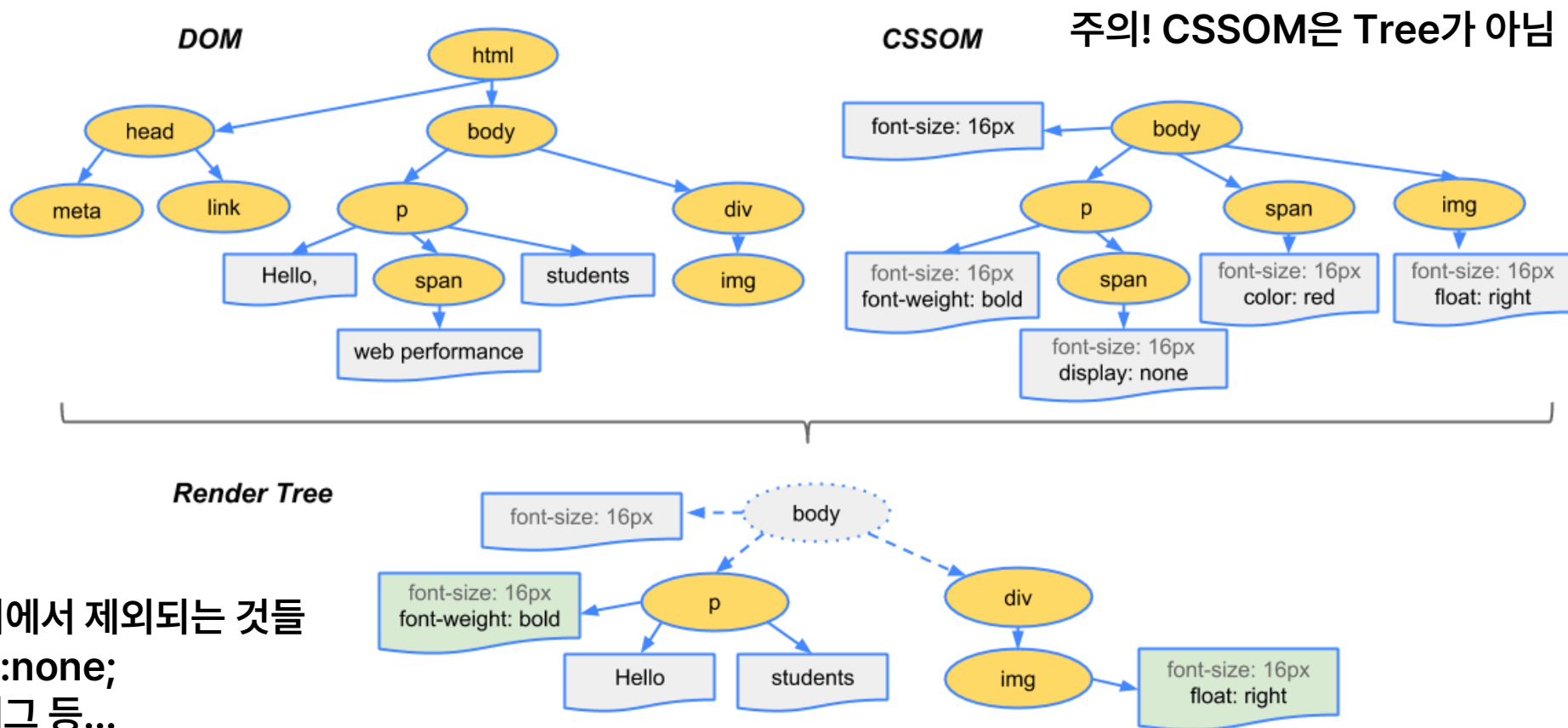
- 왜 브라우저가 다음과 같은 형식을 이해할 수 있을까?
인공지능이 알아서 읽어주는 것이 아니다.

<div>

<button></button>

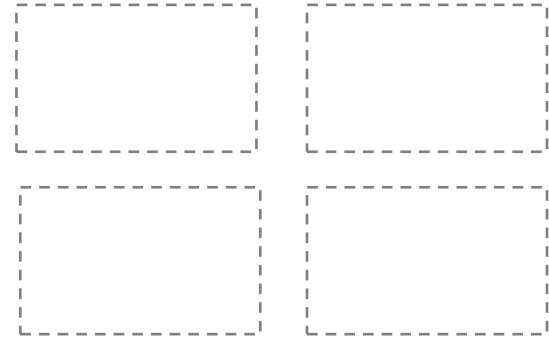
</div>

파싱



레이아웃(리플로우)과 페인트(리페인트)

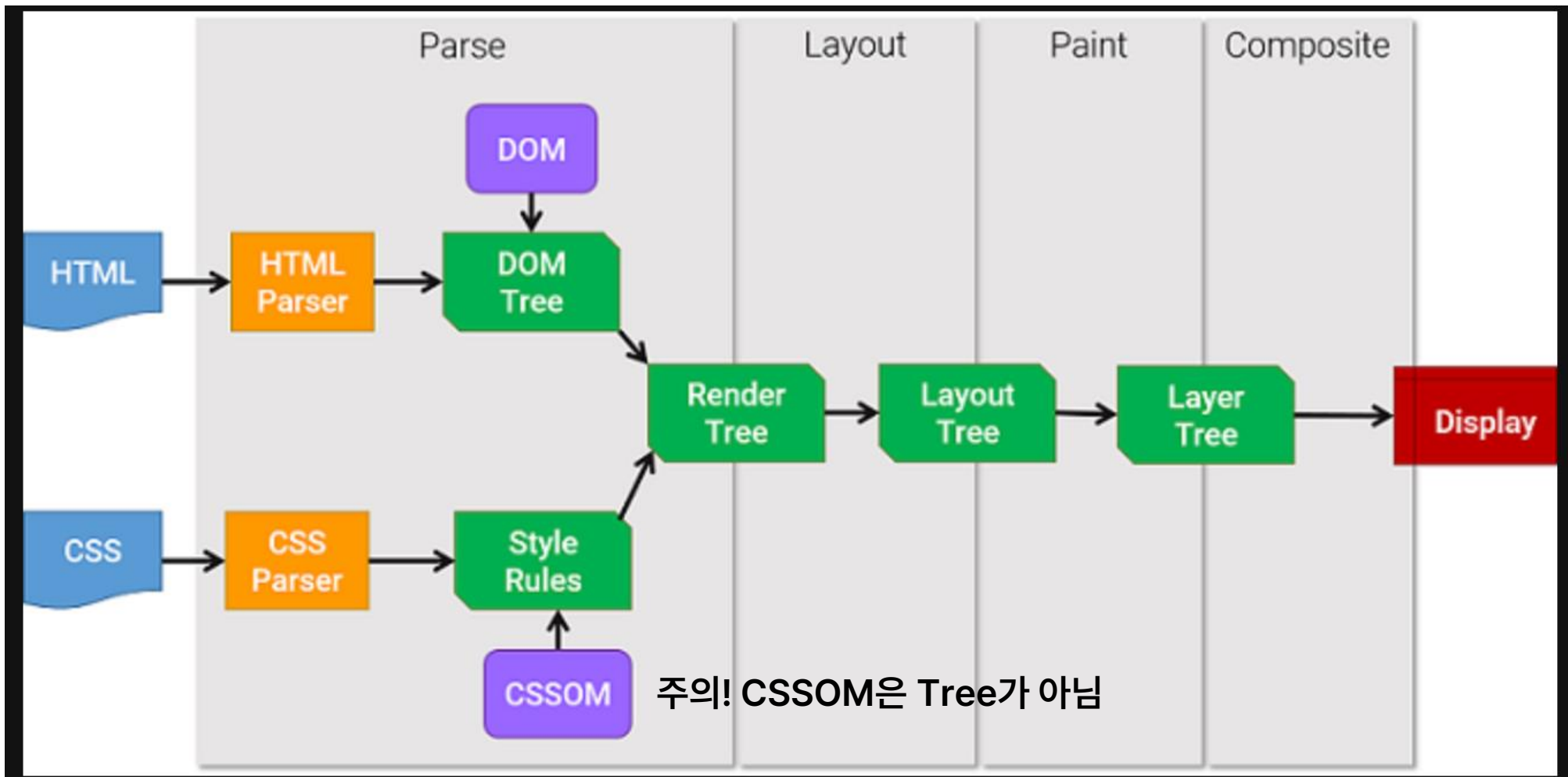
- 레이아웃은 표시될 위치와 크기를 정하는 일



- 레이아웃을 통해 렌더링 트리의 각 노드가 실질적으로 차지하는 값을 계산하고 노드들의 기하학적 형태를 구했다.

이를 화면에 그려서 사용자에게 표시하는 작업을
페인팅 또는 **래스터화** 라고 부른다.





리플로우 일으키는 CSS 속성

- Position
- Left, top, right, bottom
- Width, height, margin, padding
- Border, border-width
- Display, float, overflow
- Font-family, font-size, font-weight
- Text-align, vertical-align ...

리페인트 일으키는 CSS 속성

- Background, box-shadow
- Border-radius, border-style
- Color, outline, text-decoration

주의

- 리플로우 일으키면 리페인트도 해야한다
- 리플로우가 안 일어나고 리페인트만 일어날 수 있다.

리플로우 리페인트 둘다 생략하는 CSS 속성

- Transform
- Opacity
- (GPU가 관여할 수 있는 속성/ DOM 트리를 변경하지 않는다)

Composite만 발생하는 CSS 속성

- Composite는 여러 레이어를 하나의 레이어로 만드는 과정이다.
- Transform
- Opacity
- (GPU가 관여할 수 있는 속성/ DOM 트리를 변경하지 않는다)

렌더링 최적화 방법

1. 레이아웃과 페인트를 최소한으로 발생시키고
2. 같은 기능을 한다면 composite으로 대체한다.
3. 레이어를 잘 분리하고 최적의 레이어를 구성하는것

자바스크립트 파싱에 의한 HTML 파싱 중단

Script 태그를 앞에 넣으면 dom 조작이 안될 수 있다.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <script src="app.js"></script>
5   </head>
6   <body>
7     <ul>
8       <li>apple</li>
9       <li>banana</li>
10    </ul>
11  </body>
12 </html>
```

HTML 파싱 중단

Javascript 파싱
javascript 실행

HTML 파싱 재시작

async/defer

async vs defer



어떤 인터렉션에 의해서 DOM에 변화가 발생하게 되면 Render Tree가 그 때마다 재생성 된다.

즉, 인터렉션에 의한 변화가 발생하면 모든 요소들의 스타일을 다시 계산하고 Layout 과정과 repaint 과정을 반복하게 되는 것이다.

여기서 인터렉션이란 "Like" 버튼 클릭, 장바구니 목록에서 상품 삭제, 댓글 달기 등을 말한다.

매번 이럴 때마다 DOM을 조작하는 불필요한 비용이 발생되는데, 이 때문에 인터렉션이 잦은 페이지에서는 SPA(Single Page Application)가 유용하게 쓰인다.

인터렉션이 없는 정보만 전달하는 페이지라면 굳이 SPA를 사용할 필요가 없는 것이다.

<https://patrick-f.tistory.com/9>

면접 질문 예시

- www.naver.com을 브라우저 주소창에 입력했을 때 일어나는 일을 설명해주세요 – 네이버
- CSS의 will-change 속성을 아시나요? – 오늘의집
- 브라우저 렌더링 과정을 설명해주세요

참고자료

- <https://d2.naver.com/helloworld/59361>