



Browser Rendering Process

브라우저 작동 원리

2022.08.02

항해99 8기 A반 나유진

웹 브라우저

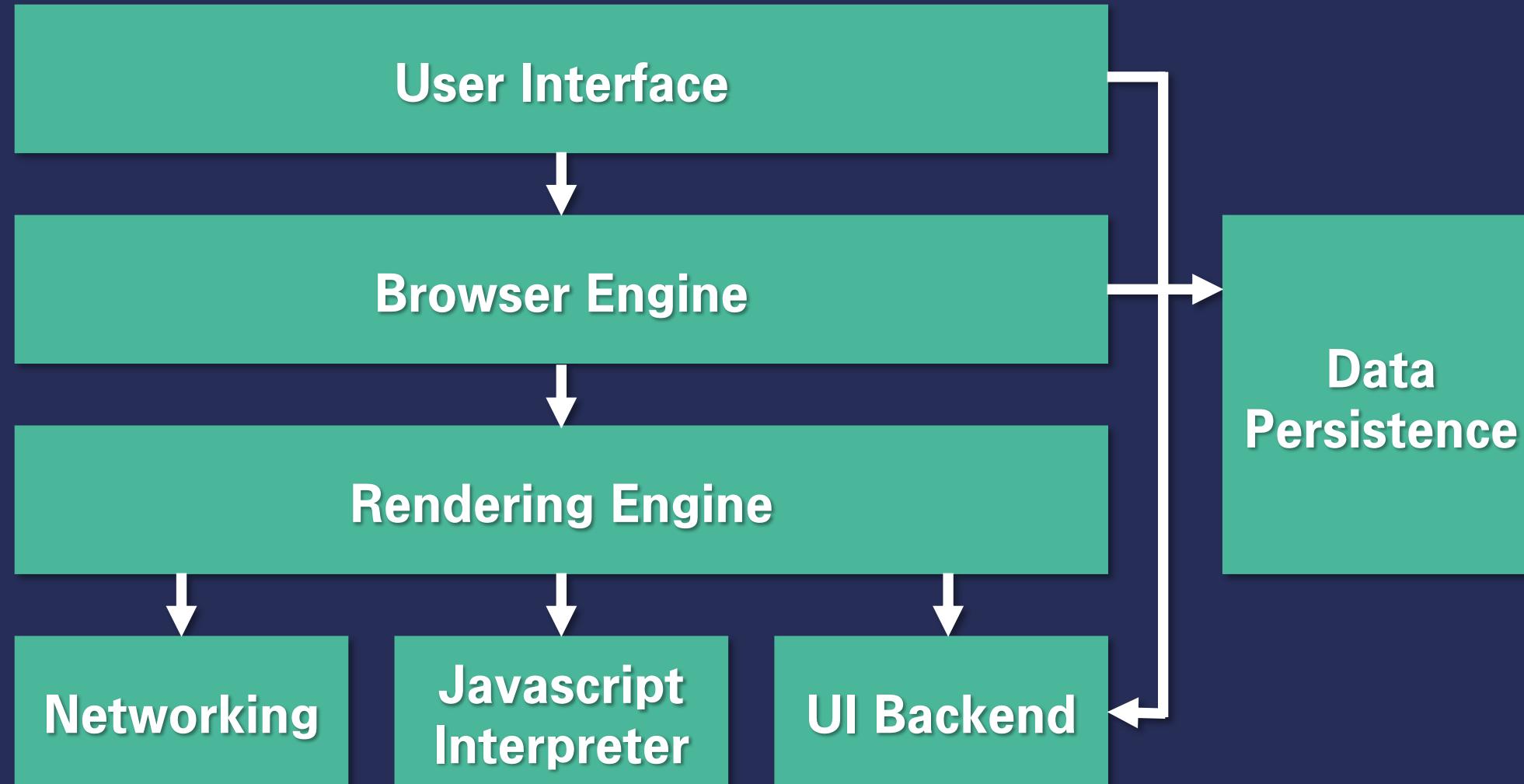
웹을 보여주는 프로그램

웹 리소스를 받아 사용자에게 보여주는

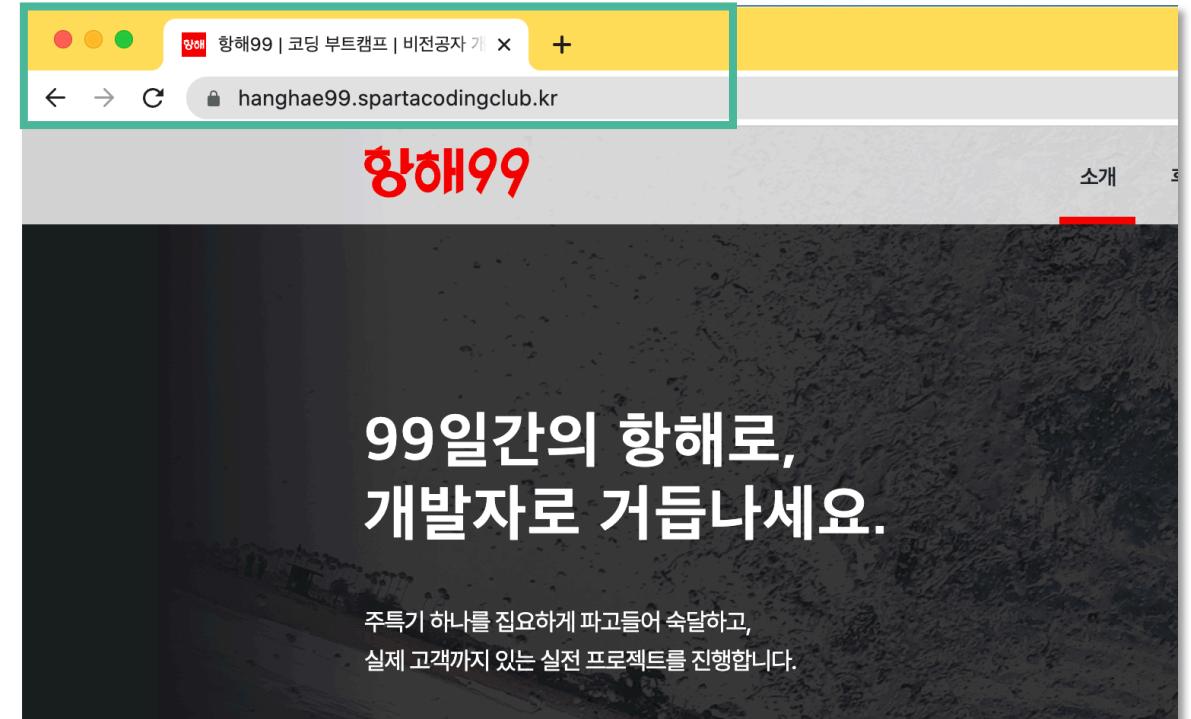
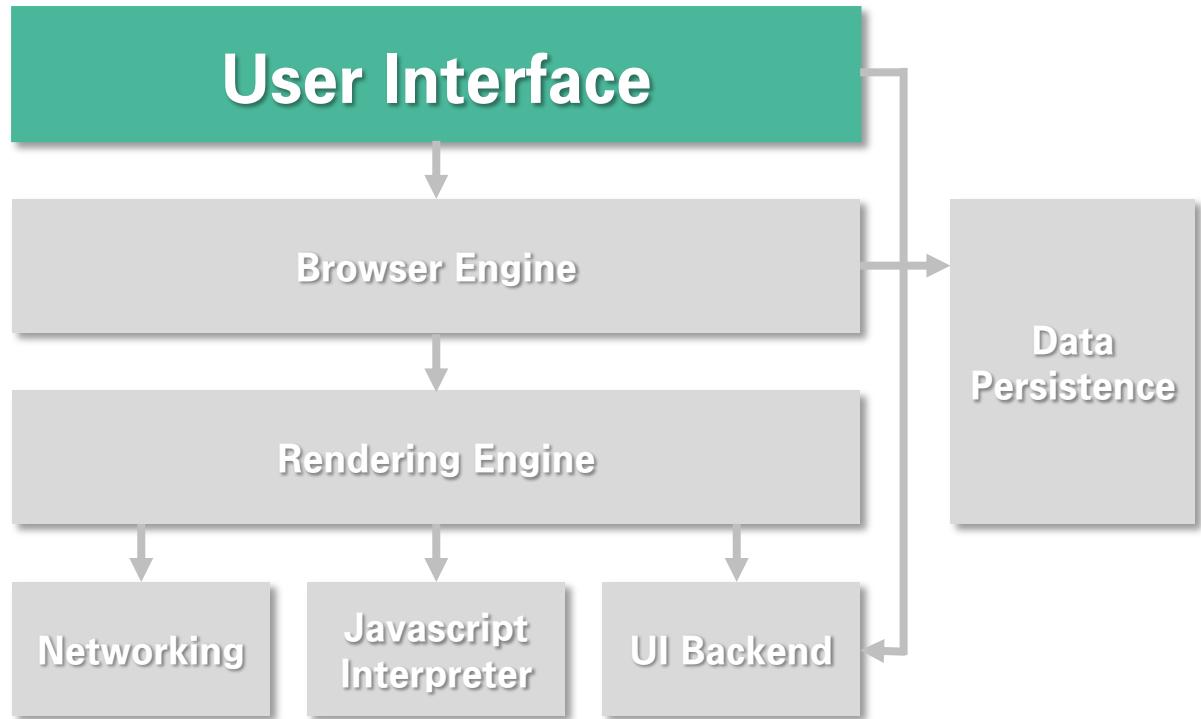
그래픽 사용자 인터페이스(GUI) 기반의 소프트웨어



웹 브라우저의 구조

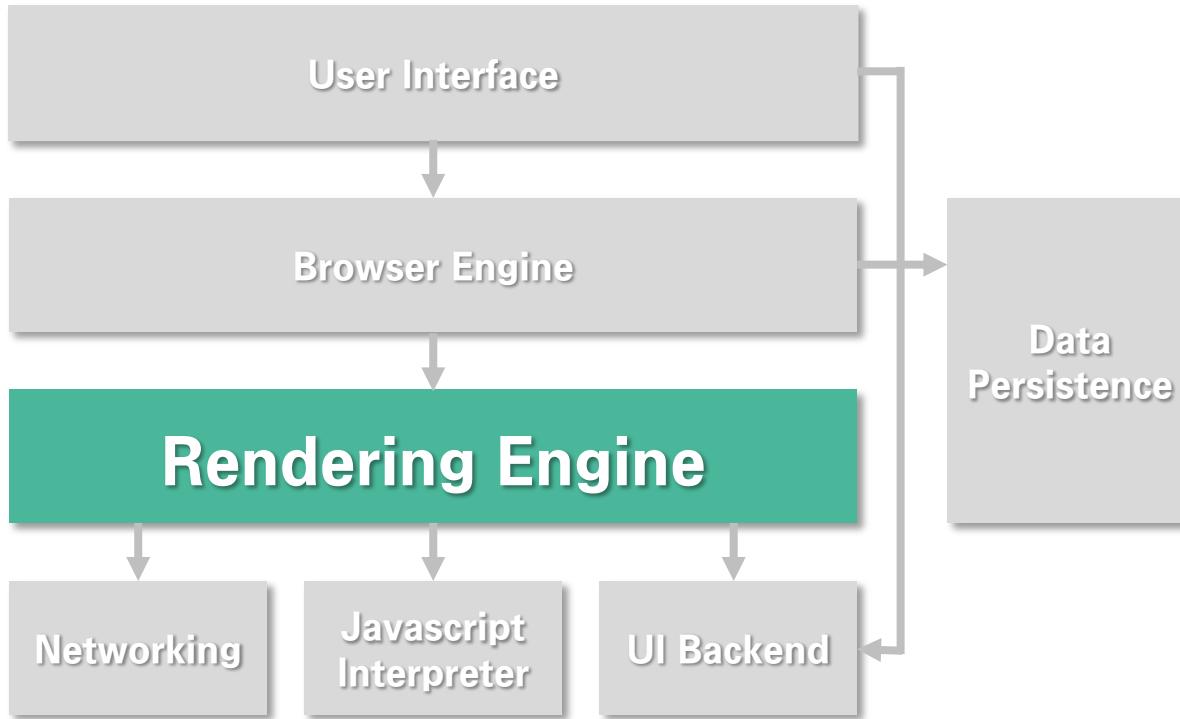


웹 브라우저의 구조 “User Interface”



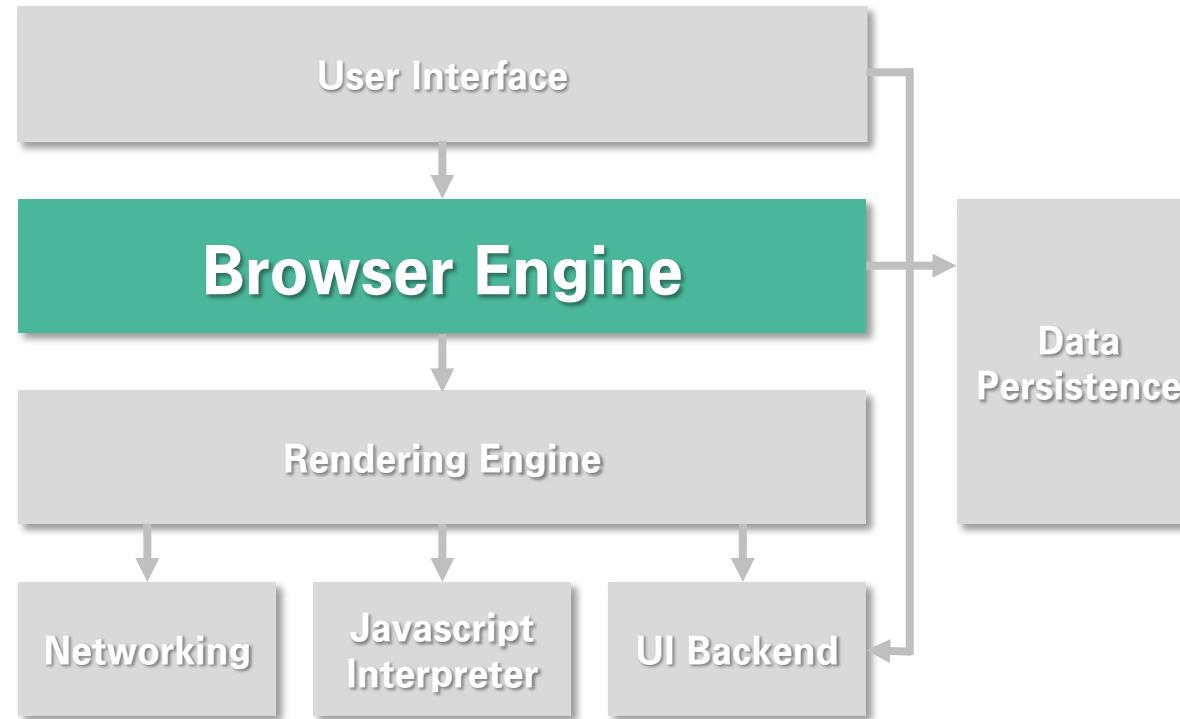
주소표시줄, 이전/다음 버튼, 홈버튼, 새로고침 등
웹 페이지를 제외하고 사용자가 컨트롤 할 수 있는 부분

웹 브라우저의 구조 “Rendering Engine”



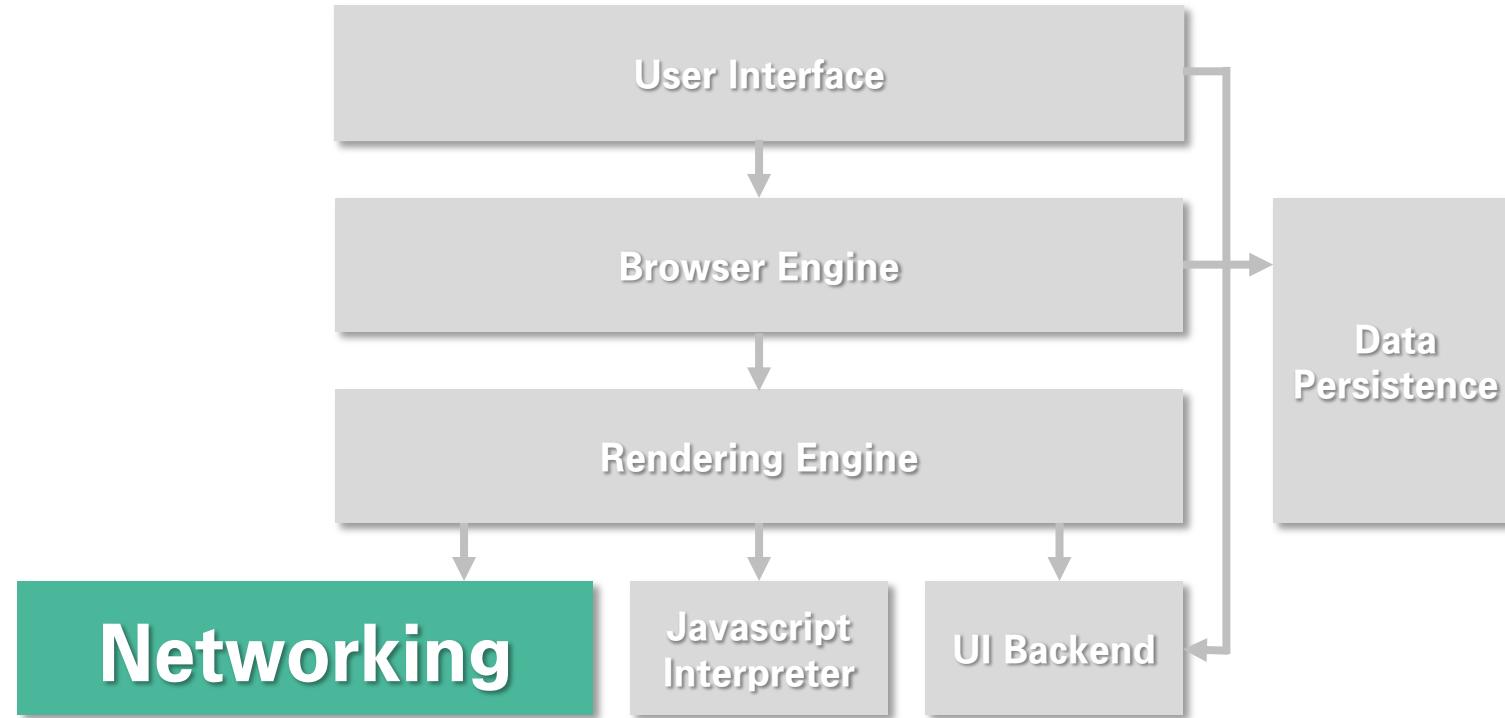
요청한 URI를 브라우저 엔진에게 받아서 Server에게 요청(통신) →
Server로 부터 URI에 해당하는 데이터(HTML과 CSS)를 받아서 파싱 Parsing한 후 렌더링 Rendering

웹 브라우저의 구조 “Browser Engine”



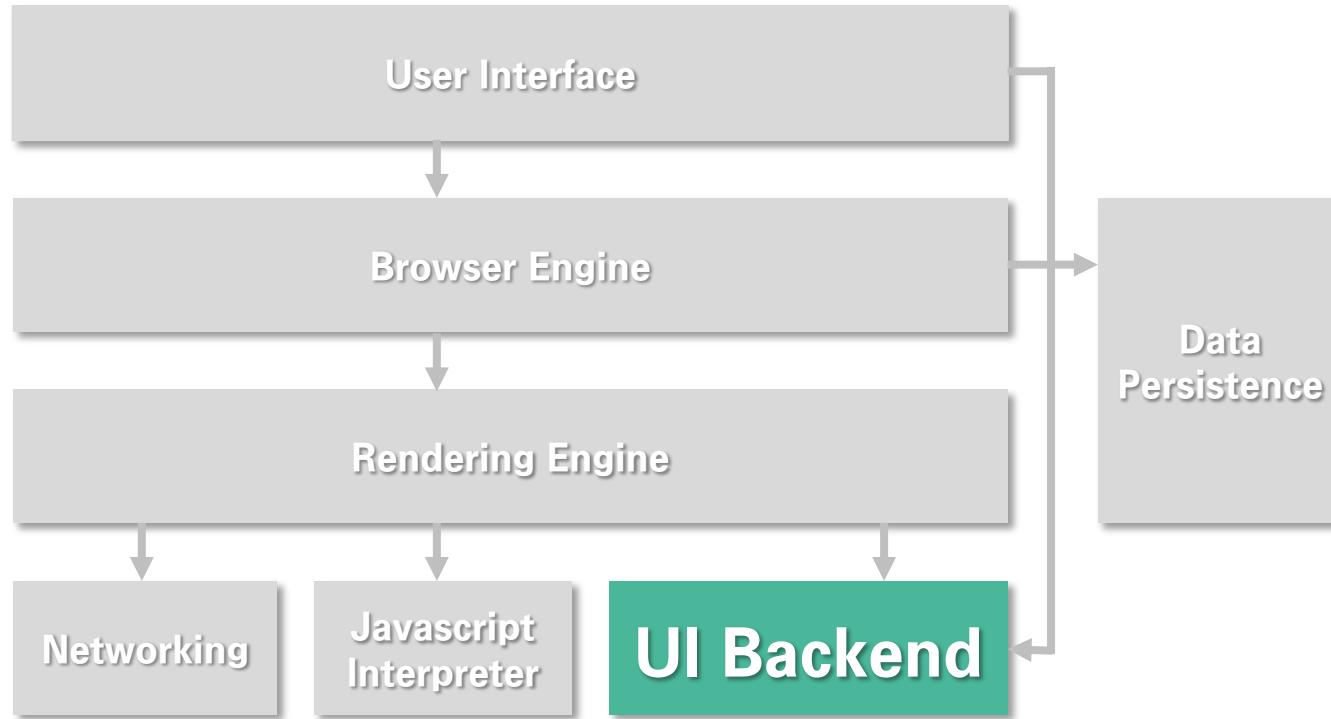
유저 인터페이스와 렌더링 엔진을 연결

웹 브라우저의 구조 “Networking”



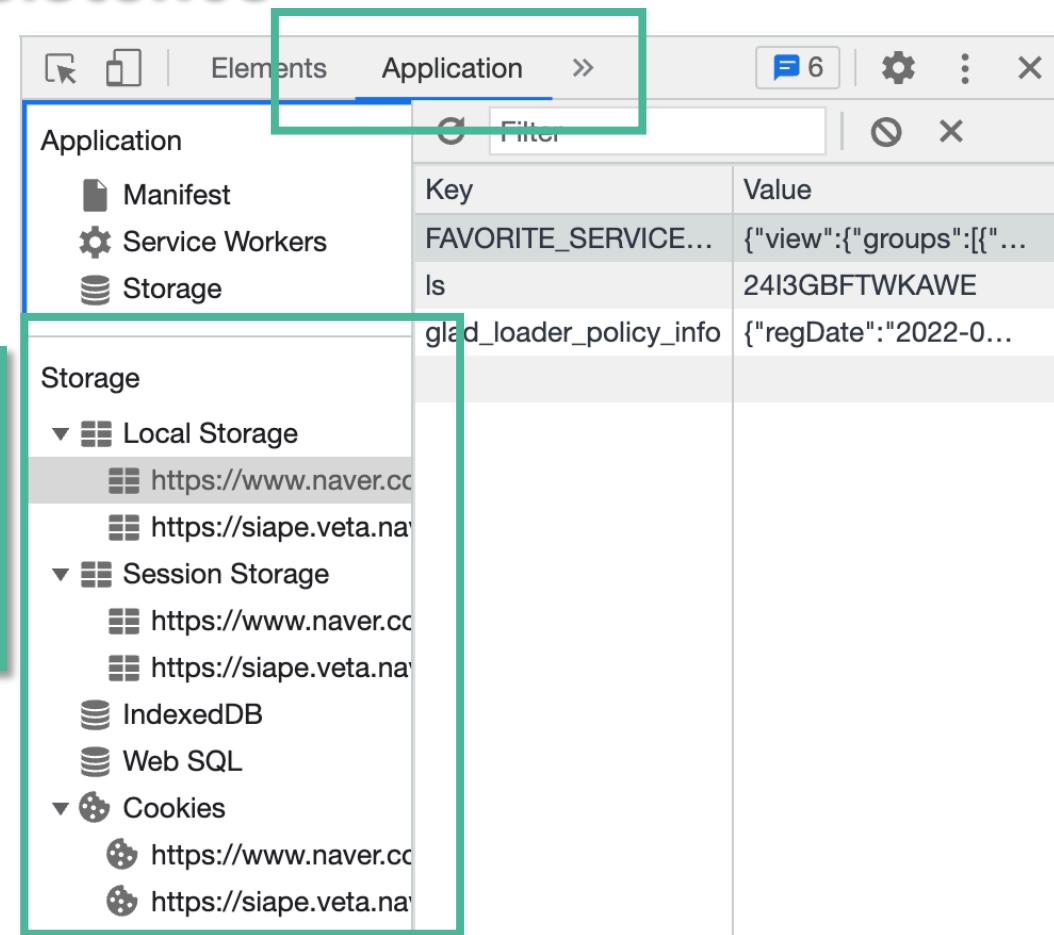
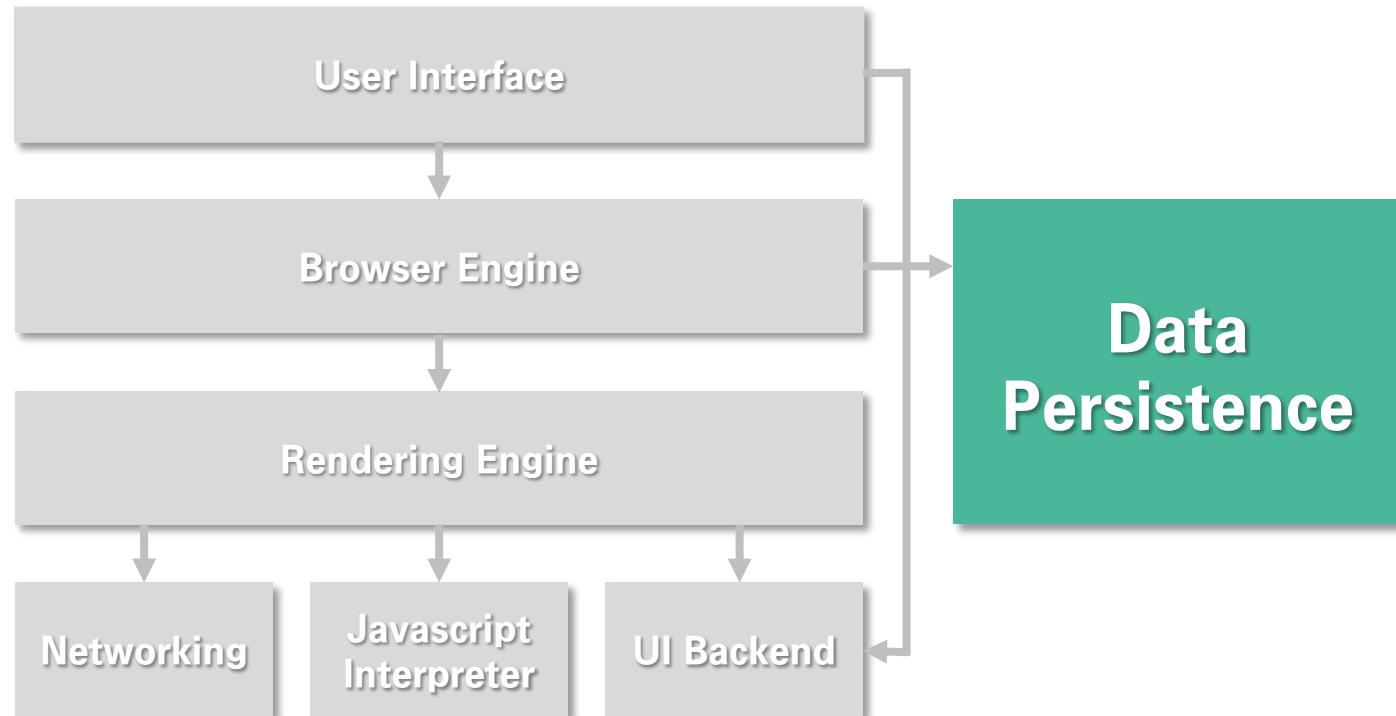
각종 네트워크 요청을 수행하는 네트워킹 파트

웹 브라우저의 구조 “UI Backend”



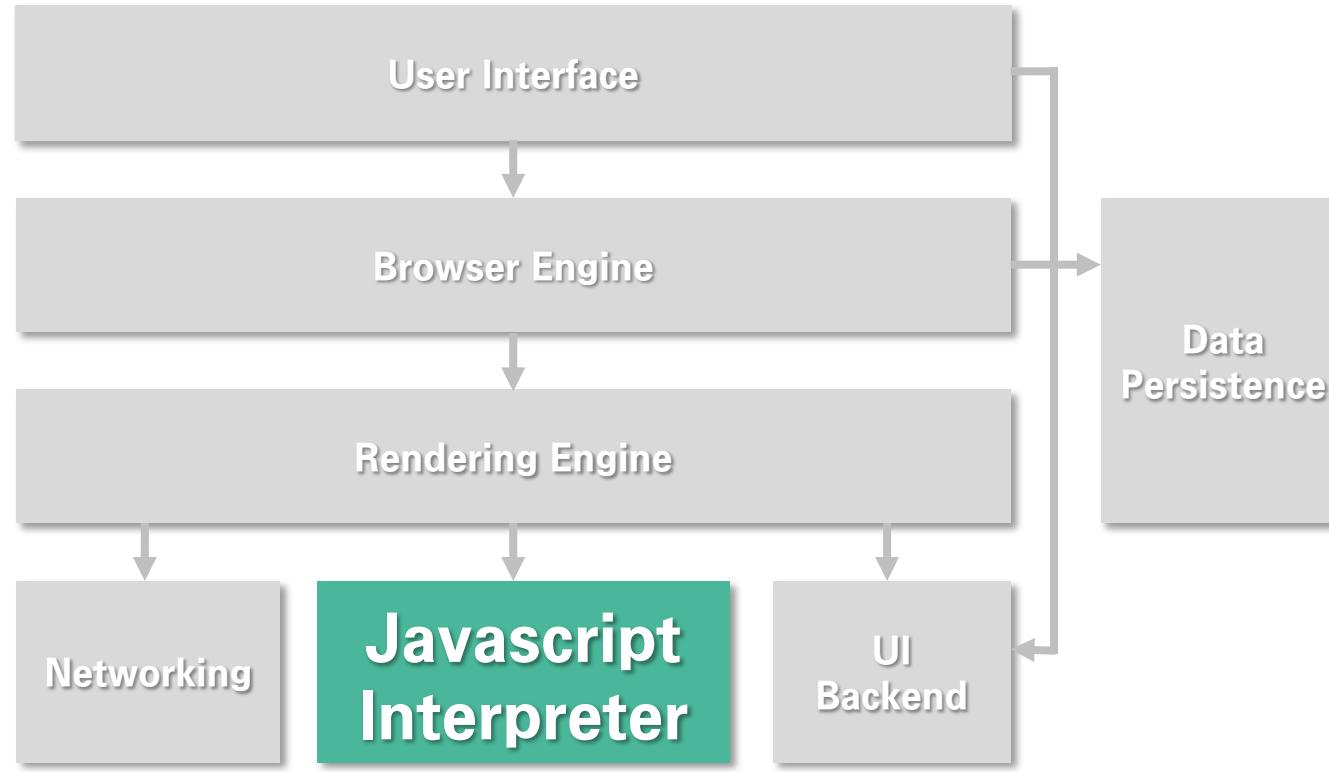
체크박스, 버튼 등 기본적인 위젯을 그려주는 파트

웹 브라우저의 구조 “Data Persistence”



Local Storage나 Cookie와 같은 보조기억장치에 데이터를 저장하는 파트

웹 브라우저의 구조 “Javascript Interpreter”



자바스크립트 코드를 실행하는 인터프리터

웹 브라우저 별 엔진



Blink

Chrome
Microsoft Edge
비발디
브레이브
삼성 인터넷
오페라
...



Gecko

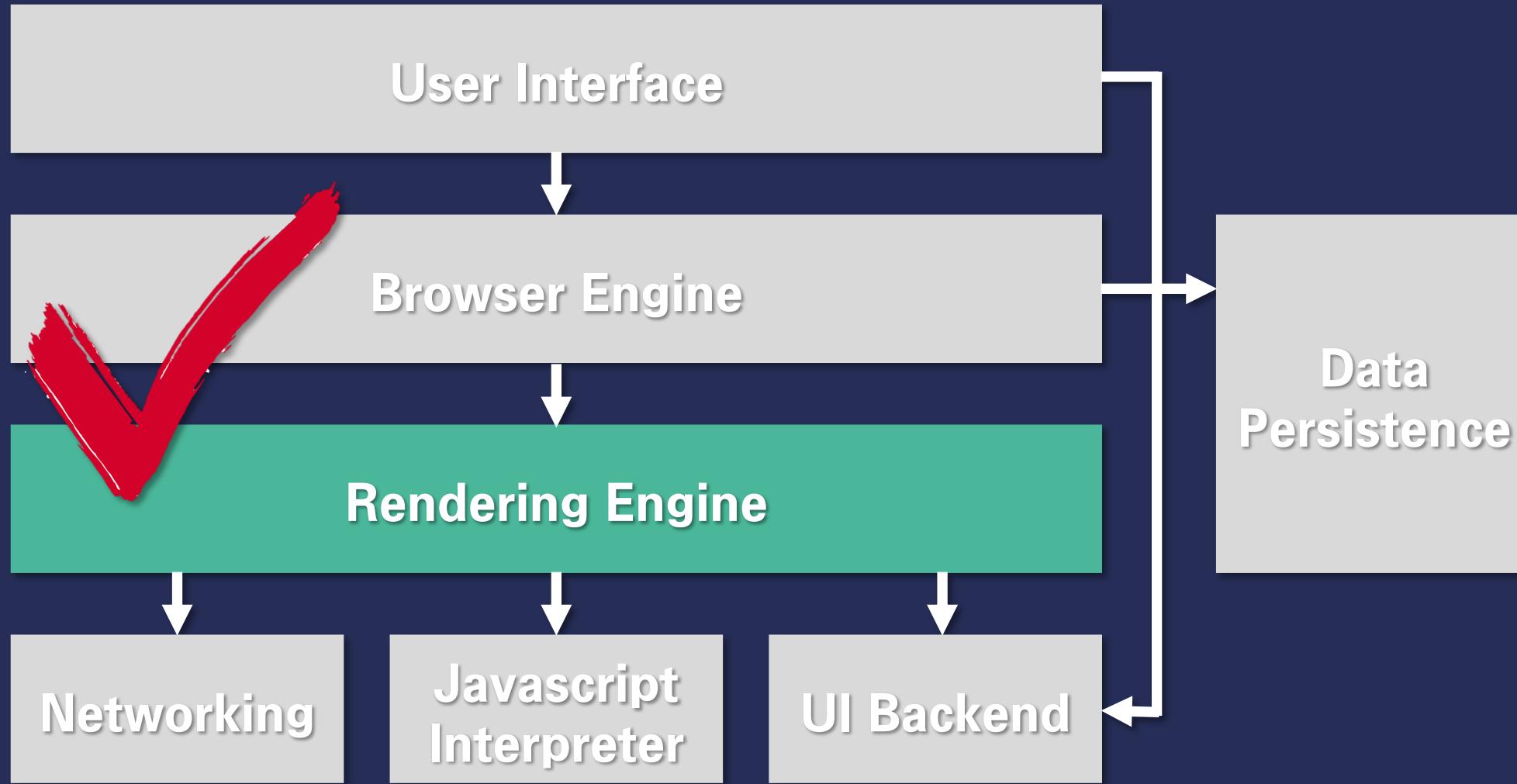
Firefox
씨몽키
K-Meleon
토어



WebKit

Safari
미도리

웹 브라우저의 구조



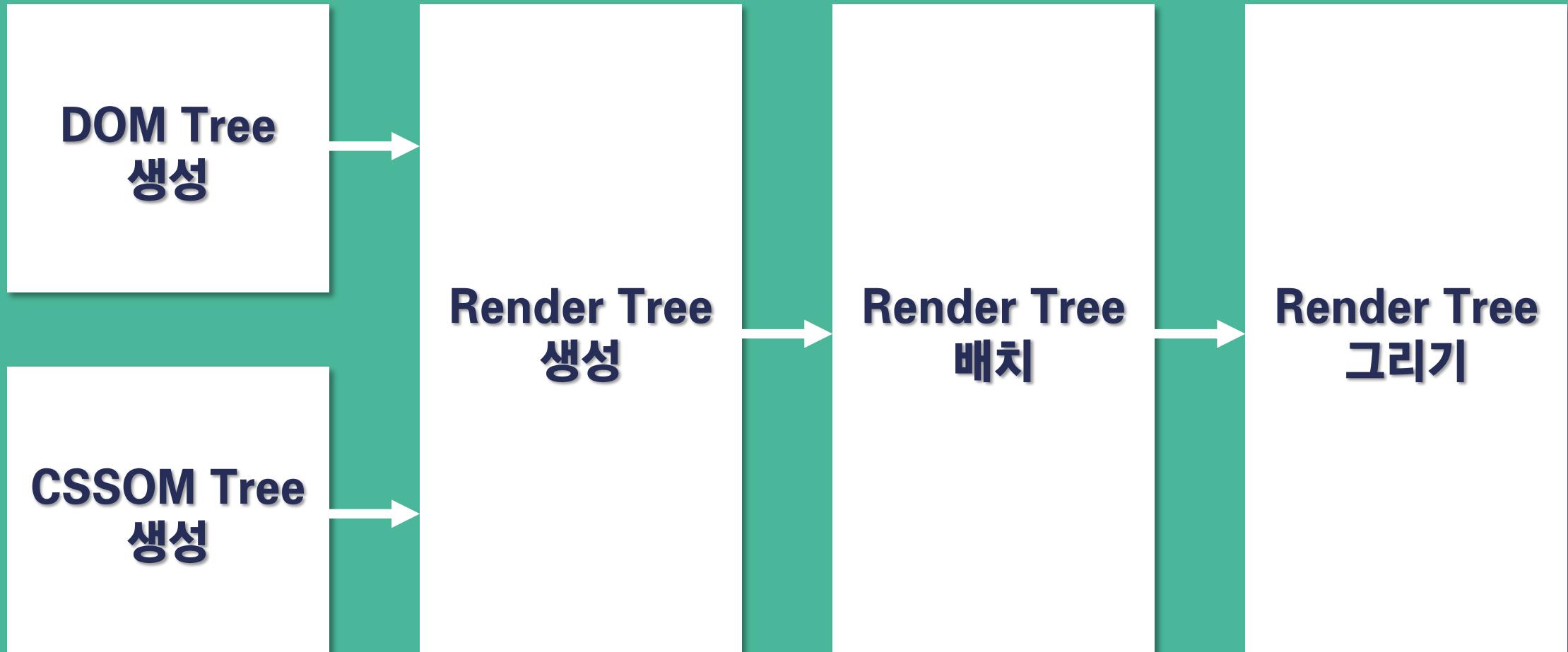
렌더링 엔진의 목표

HTML CSS JS 이미지 등
웹 페이지에 포함된 모든 요소들을
화면에 보여주기!

업데이트가 필요할 때
효율적으로 렌더링을 할 수 있도록
자료구조를 생성하기!

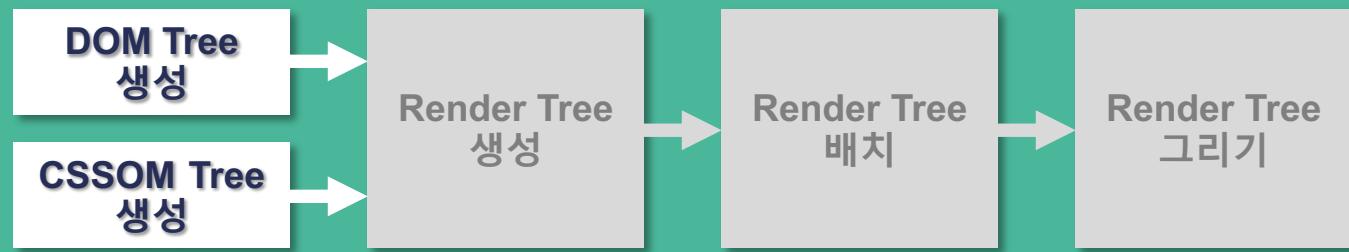
렌더링 엔진 동작 과정

Critical Rendering Path



렌더링 엔진 동작 과정

Critical Rendering Path



```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>항해99 8기 항해톡 :: 브라우저작동원리</title>
  <link rel="stylesheet" href="style.css" />
</head>
<body>
  <!-- 주석! -->
  <h1>브라우저 작동원리</h1>
  <p>브라우저는 ~~~
    <span>오늘은 22년 8월 2일!</span>
  </p>
</body>
</html>
```

토큰화

StartTag
html

StartTag
head

...

EndTag
head

StartTag
body

StartTag
h1

Text
항해99 ~

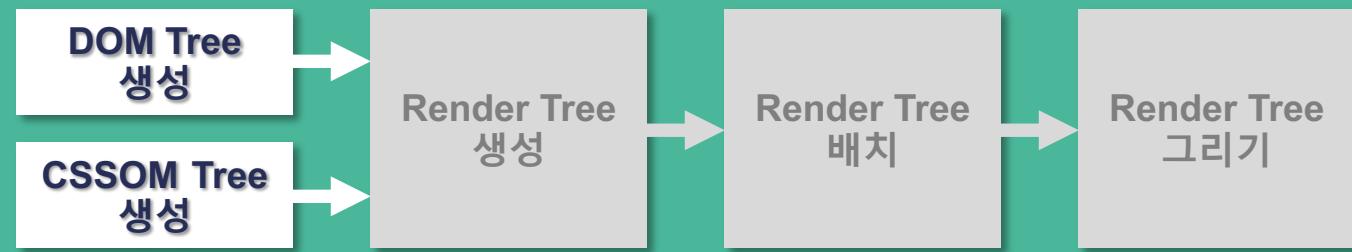
EndTag
h1

EndTag
body

EndTag
html

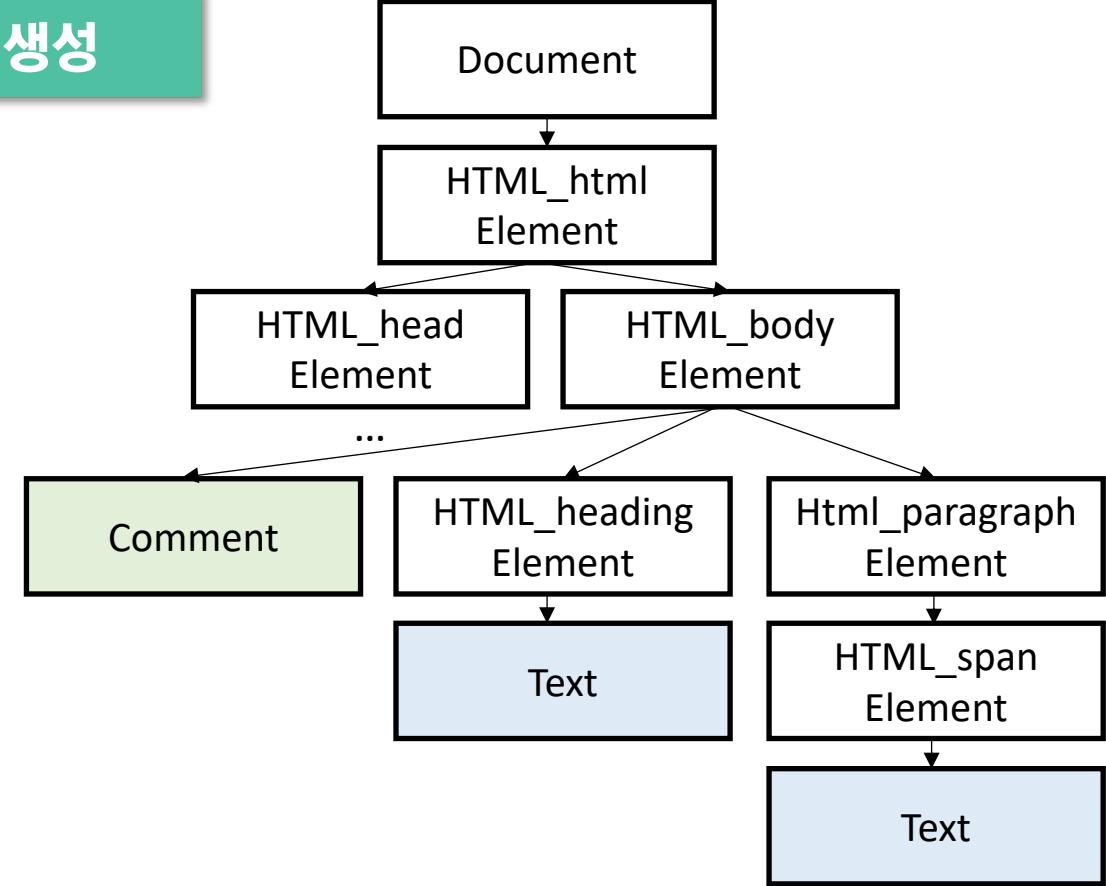
렌더링 엔진 동작 과정

Critical Rendering Path



```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>향해99 8기 향해톡 :: 브라우저작동원리</title>
  <link rel="stylesheet" href="style.css" />
</head>
<body>
  <!-- 주석! -->
  <h1>브라우저 작동원리</h1>
  <p>브라우저는 ~~~
    <span>오늘은 22년 8월 2일!</span>
  </p>
</body>
</html>
```

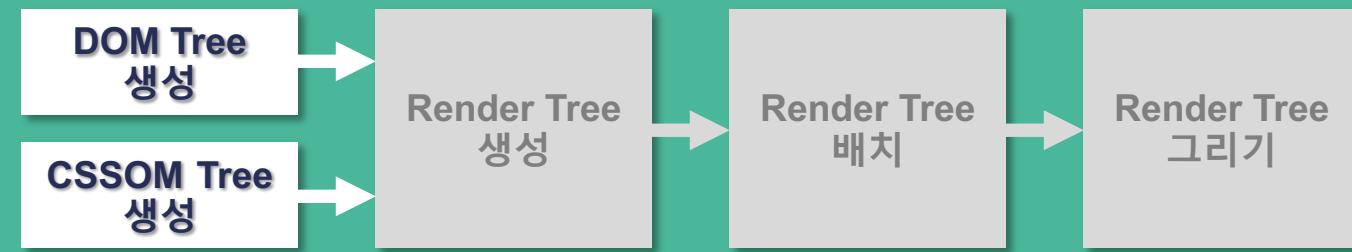
DOM Tree 생성



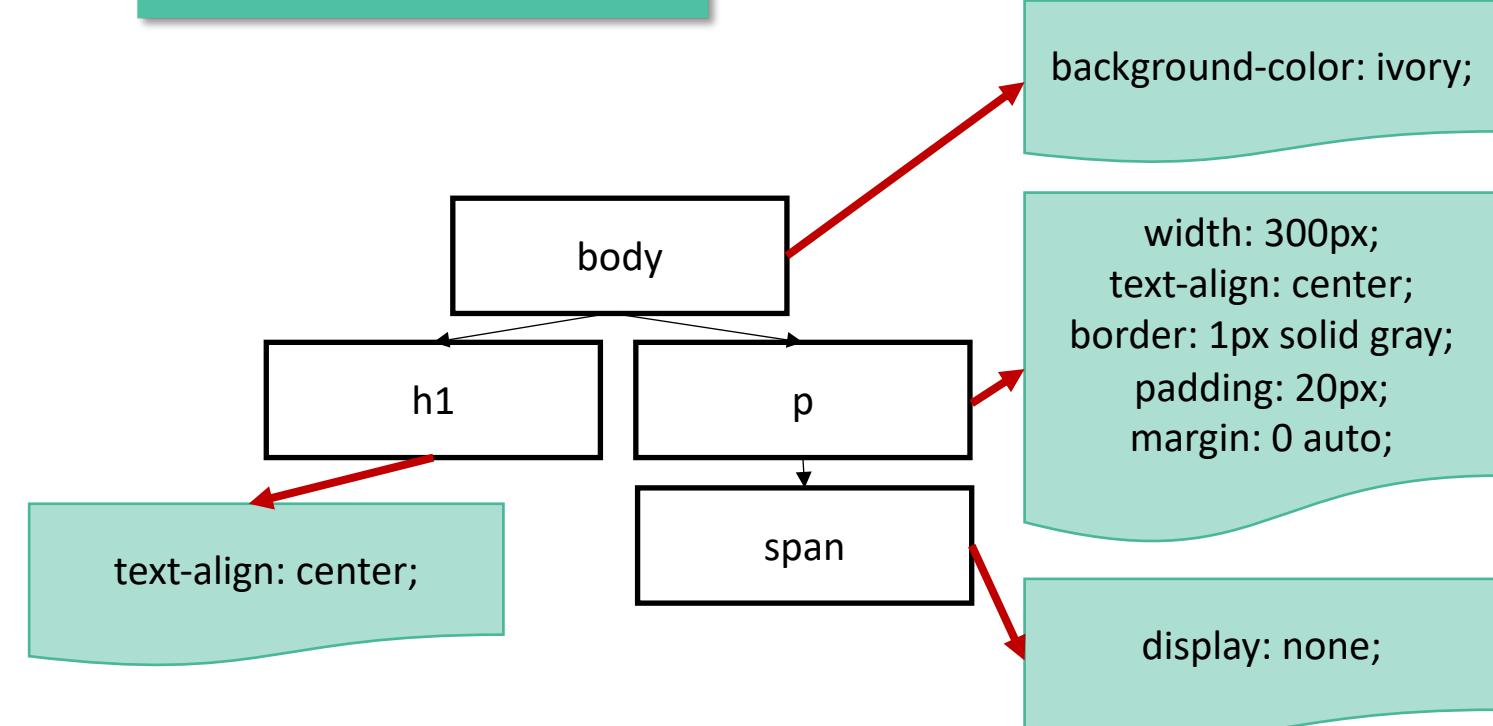
렌더링 엔진 동작 과정

Critical Rendering Path

```
body {  
    background-color: #ivory;  
}  
  
h1 {  
    text-align:center;  
}  
  
p {  
    width:300px;  
    text-align: center;  
    border: 1px solid #gray;  
    padding: 20px;  
    margin:0 auto;  
}  
  
span {  
    display: none;  
}
```

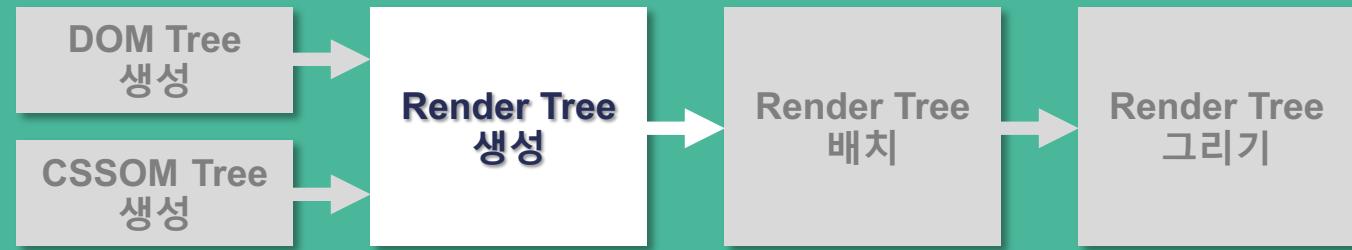


CSSOM Tree 생성



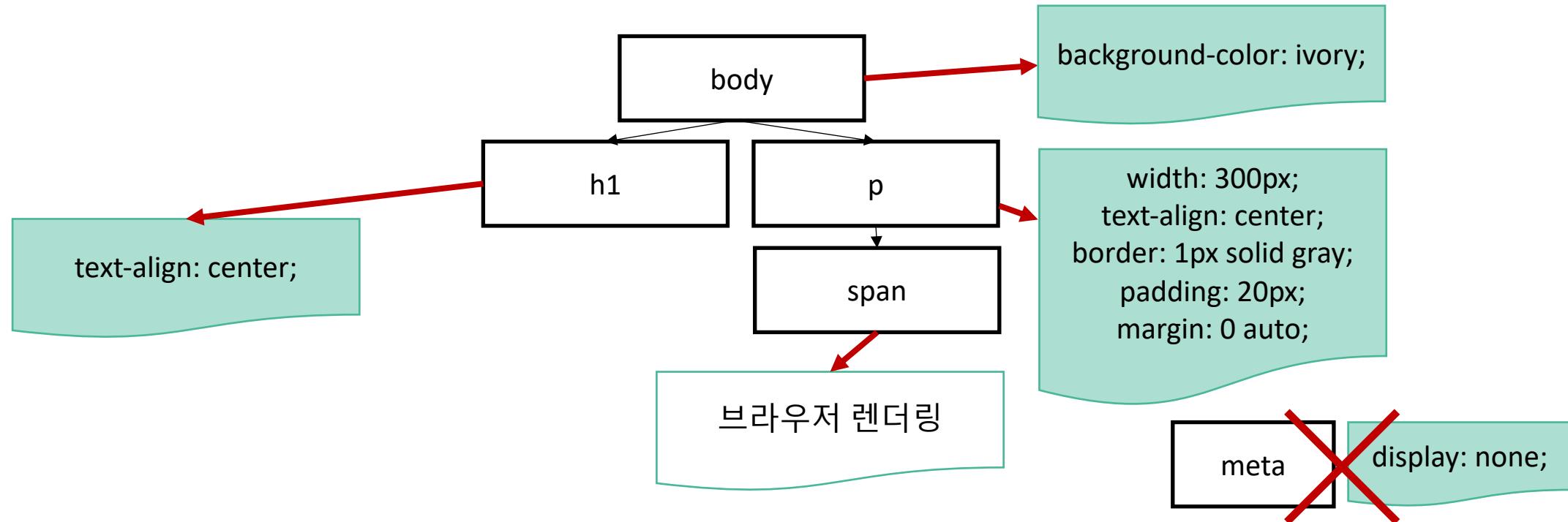
렌더링 엔진 동작 과정

Critical Rendering Path



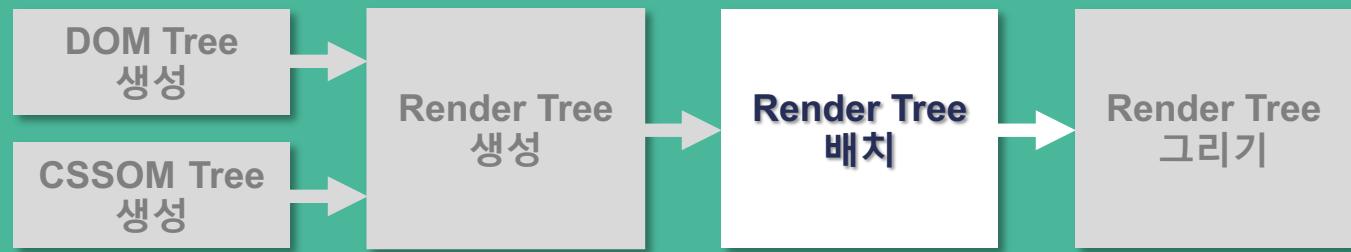
Render Tree 생성

화면에 표시되어야 할 모든 노드의 컨텐츠, 스타일 정보를 포함하는 트리

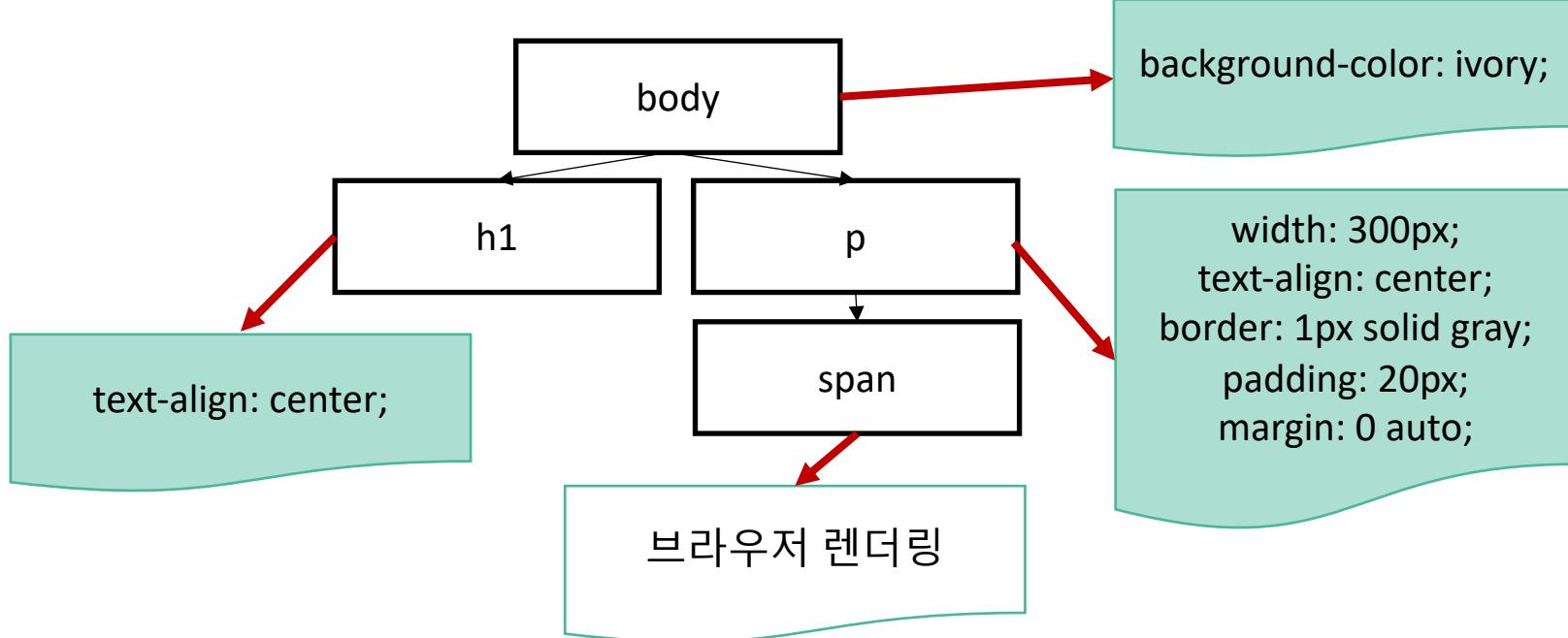


렌더링 엔진 동작 과정

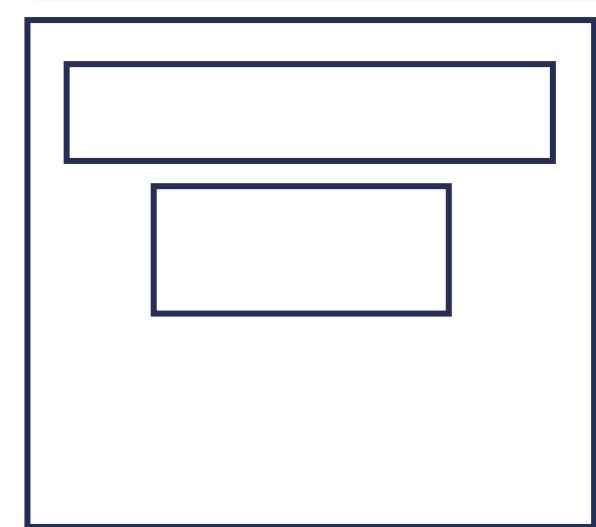
Critical Rendering Path



Render Tree 배치

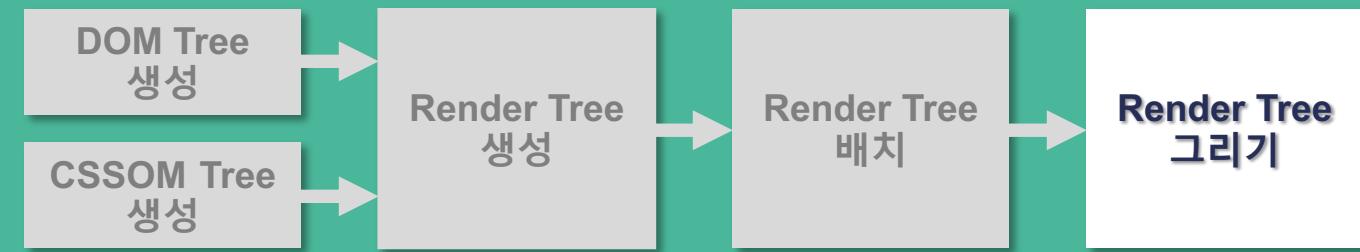


Layout

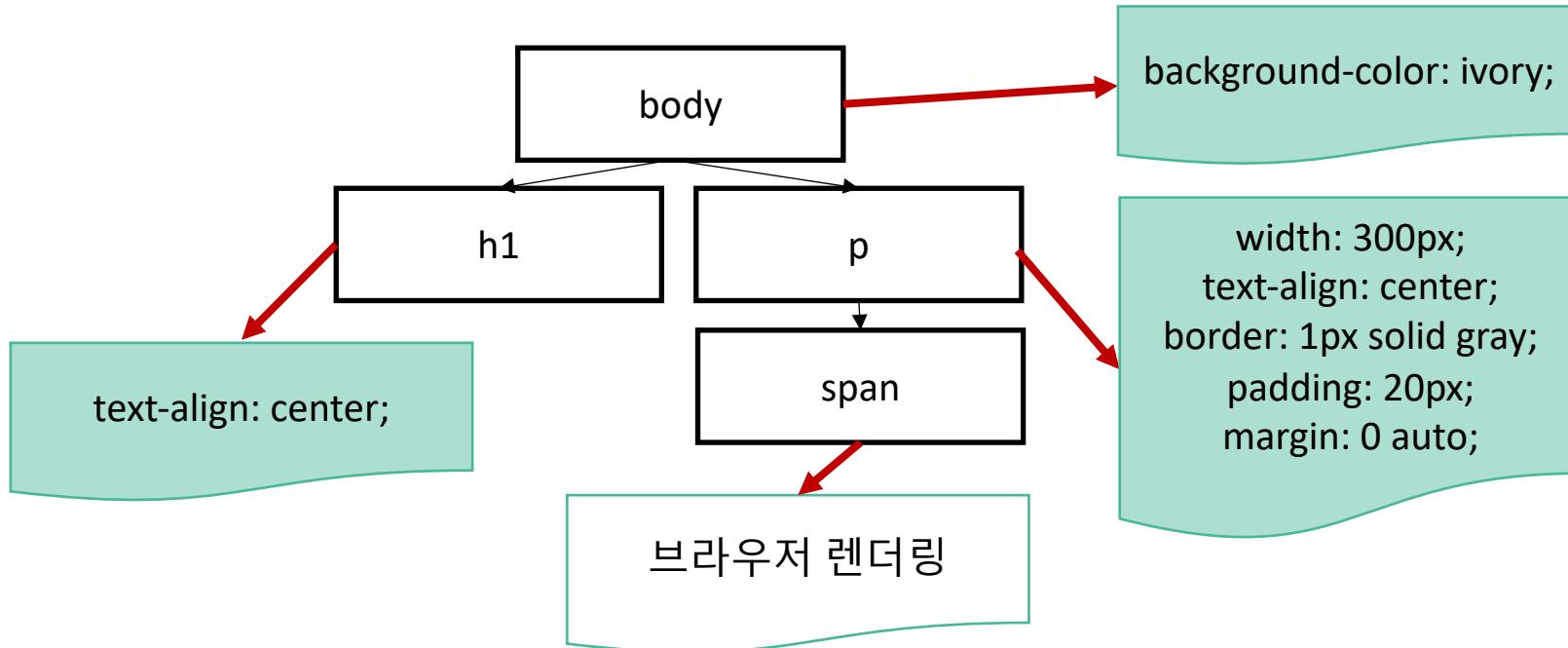


렌더링 엔진 동작 과정

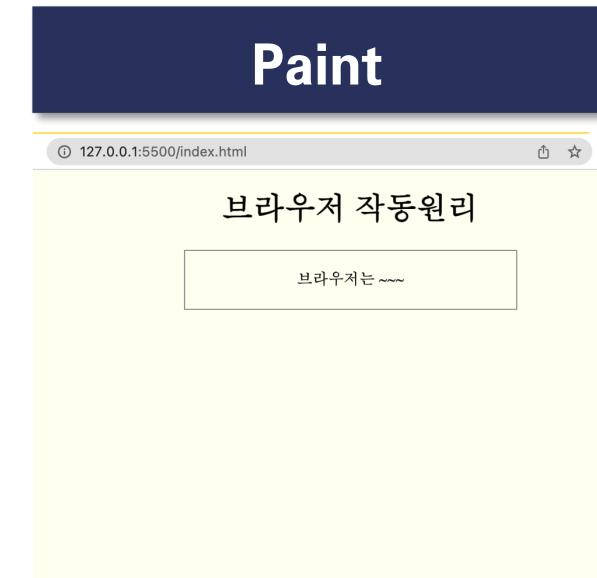
Critical Rendering Path



Render Tree 그리기



Paint



**Critical Rendering Path의
시간을 줄이면,
브라우저가 웹 페이지를 보여주는데
시간을 줄일 수 있다!**

BUT!
**사용자 동작으로 JS 실행되어
CSS 변경이나 애니메이션이 일어났을 때는?**

UI 업데이트 상황

다시 Layout이 발생하는 경우

요소의 크기나 위치가 바뀔 때

브라우저 창의 크기가 바뀌었을 때

Javascript

Style

Layout

Paint

Composite

UI 업데이트 상황

Paint부터 다시 발생하는 경우

배경 이미지나 텍스트 색상, 그림자 등

레이아웃의 수치를 변화시키지 않는 스타일의 변경이 일어났을 때

Javascript

Style

Layout

Paint

Composite

UI 업데이트 상황

레이어 합성만 다시 발생하는 경우

Layout과 Paint를 수행하지 않고 레이어 합성만 발생하기 때문에
성능상으로 가장 큰 이점을 가짐

Javascript

Style

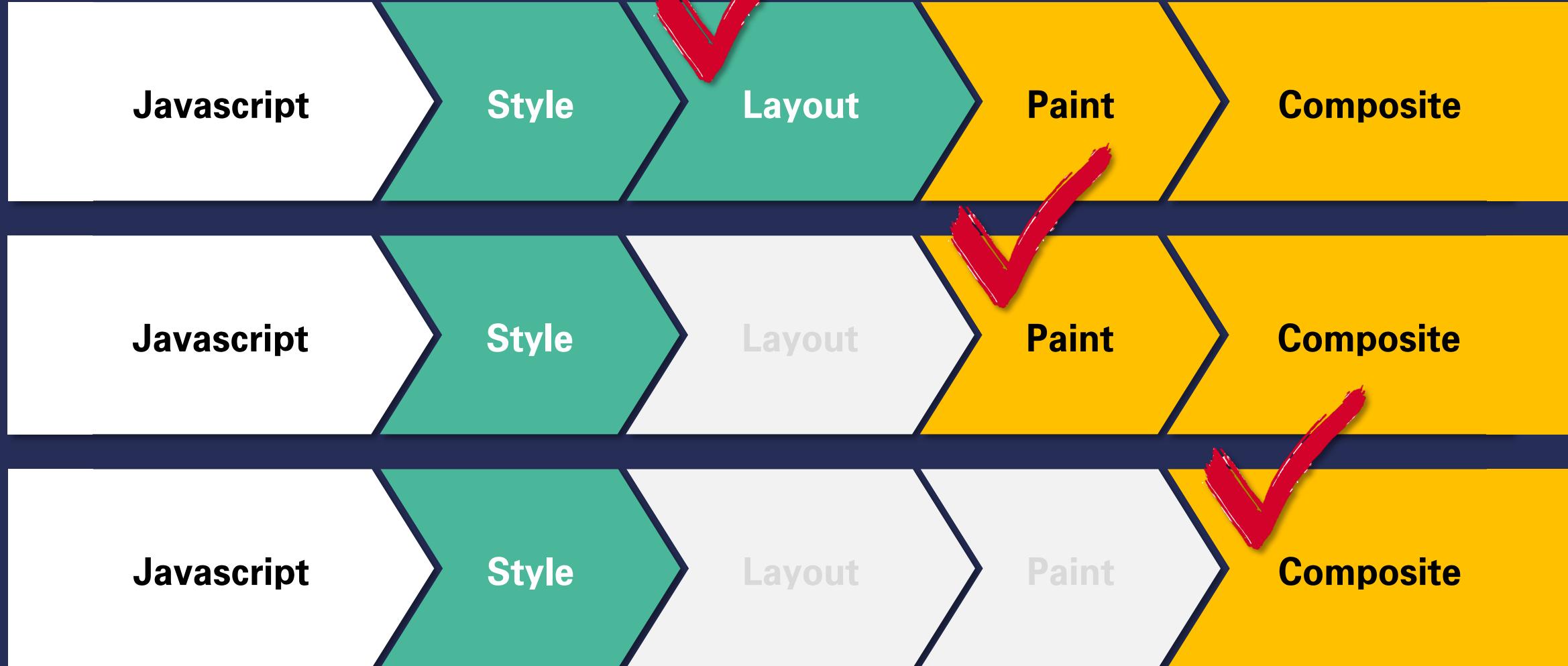
Layout

Paint

Composite

UI 업데이트 상황

** CSS 속성별 렌더링 일어나는 경우 확인 : <https://csstriggers.com/>



CSS Rendering 확인 방법

Google 항해99

전체 이미지 뉴스 동영상 지도 더보기 도구

검색결과 약 533,000개 (0.28초)

광고 · https://www.spartacodingclub.kr/ ▾

부트캠프 항해99 - 개발자 부트캠프

항해는 쉽지 않습니다. 하지만 포기하지 않도록 케어해주는 매니저가 있어 이겨낼 수 있습니다. IT기업이 대부분 사용하는 스프린트 형태로 개발하며, 현업과 같은 개발 사이클을 경험합니다.

실전 프로젝트

예비 개발자와 현업 디자이너의 협업 항해99의 실전 프로젝트 결과물

개발자 취업 무료강의

개발자 취업준비의 모든 것! 매일 선착순 100명, 한정 3일간 수강

가장 많이 하는 질문

지원자들이 가장 많이하는질문 다른 사람들은 어떨지 궁금하다면?

카카오로 1초 가입하기

탄탄한 기초부터 나만의 주특기까지, 1초만에 합류하세요

광고 · https://www.f-lab.kr/ ▾

F-Lab - 상위 1% 개발자 멘토 - 깊이 있게 제대로 배웁니다

고연봉 백엔드 개발자로 성장하고 싶다면? 고연봉 백엔드 개발자에게 직접 배우세요! 네카라쿠배 출신의 상위 1% IT 출신 개발자 멘토를 만나보세요. 대용량 트래픽 처리. 모의면접 등 취업 준비. 기술&인성 면접 지원. 고성능 튜닝.

Elements Console Sources Network Memory Performance Screenshots Memory Web Vitals

92 ms 108 ms 308 ms 508 ms 708 ms 908 ms 1108 ms 1308 ms 1508 ms 1708 ms

▶ Network search (www.google.com) 533.3 ms

▶ Frames: 166.7 ms 316.7 ms 216.7 ms

▶ Timings FP FCP LCP DCL L

Experience

▶ Main — https://www.google.com/webhp?hl=ko&sa=X&ved=0ahUKEwjQ3avrj6X5AhV_mFYBdfkAWwQPAgl

▼ Main — https://www.google.com/search?q=%ED%95%AD%ED%95%B499&hl=ko&sxsrf=ALiCzsZtN_LI86ypm6o0nmymjgd

Task T... Task Task Task Task

Pars...TML Eva...pt Pa...L Ev...t Ev...t R... s...



The End.

감사합니다.