

브라우저 렌더링 원리

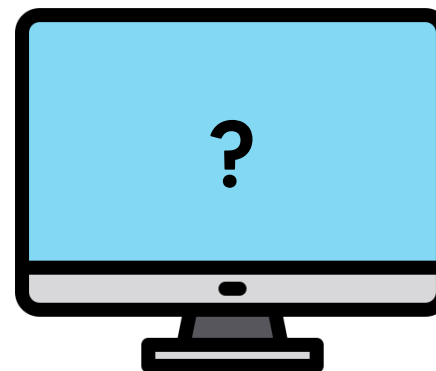


항해99 14기 프론트 기술면접 스터디 1주차
김한솔

브라우저 렌더링 원리를 알아야 하는 이유



대부분의 프로그래밍 언어는
운영체제(OS)나 가상머신(VM) 위에서 실행



그렇다면 클라이언트 사이드에서의
자바스크립트는?

브라우저 렌더링 원리를 알아야 하는 이유



브라우저에서 HTML, CSS와 함께 실행된다.

브라우저 렌더링 원리를 알아야 하는 이유

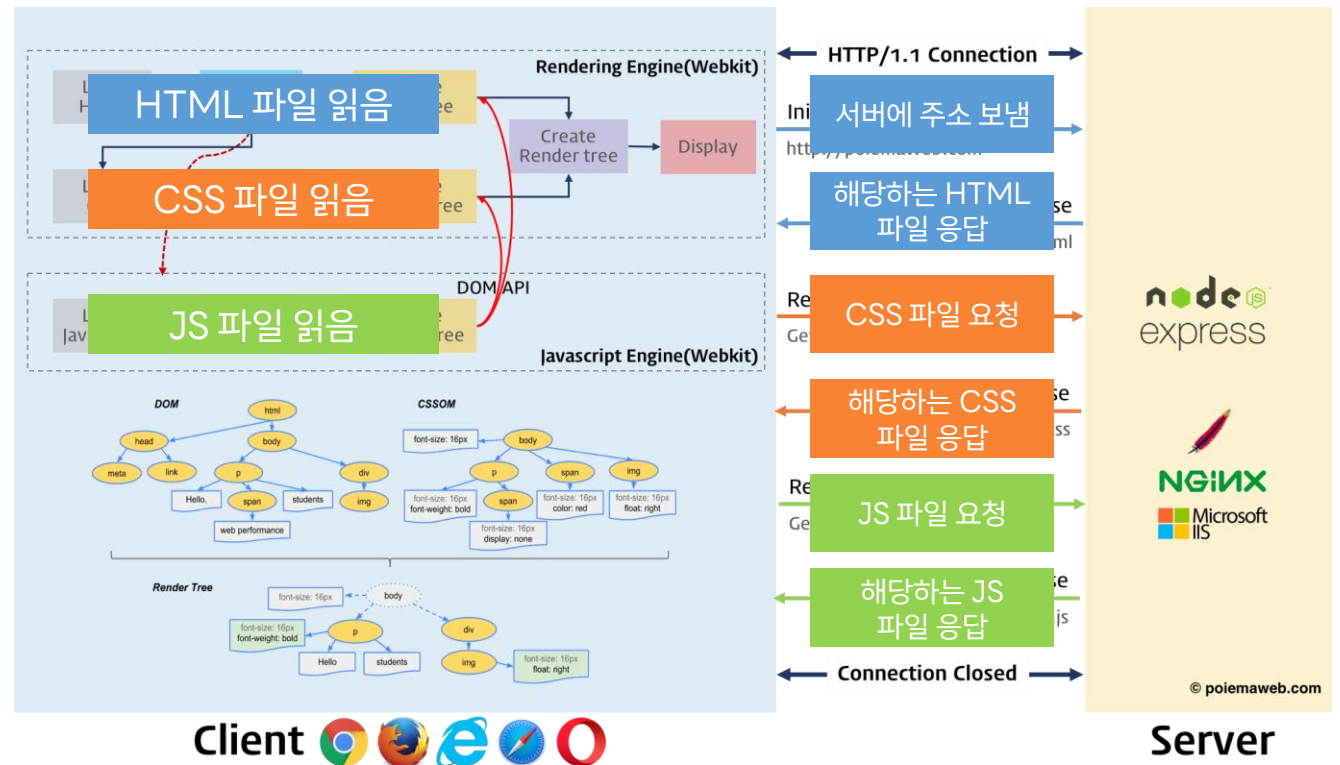


내가 의도한대로 코드를 실행시키려면
브라우저가 어떻게 렌더링되는지 파악해야
더 **효율적인 코드**를 작성할 수 있다.

브라우저의 렌더링 과정 - 요청과 응답

```
<!DOCTYPE html>
<html Lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible"
content="IE=edge">
  <meta name="viewport" content="
width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="
./style.css">
  <title>Document</title>
</head>
<body>
  <!-- 어쩌고 저쩌고 html 내용 -->
  <script src="./script.js"></script>
</body>
</html>
```

<https://www.항해14기.com/>



브라우저의 렌더링 과정 - 파싱과 렌더링?

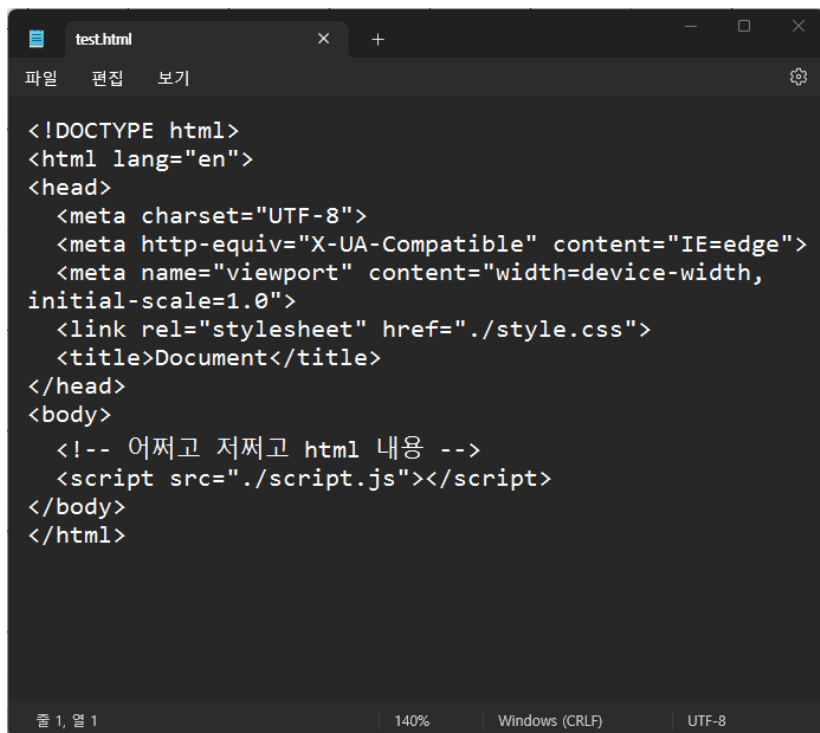
파싱

프로그래밍 언어의 문법에 맞게 작성된
텍스트 문서를 읽고 실행하기 위해
텍스트 문자의 문자열을 **분해**하고 **구조를 생성**하는 일련의 과정

렌더링

HTML, CSS, JS로 작성된 문서를 파싱하여
브라우저에 **시각적으로 출력**하는 것

HTML 파싱과 DOM 생성



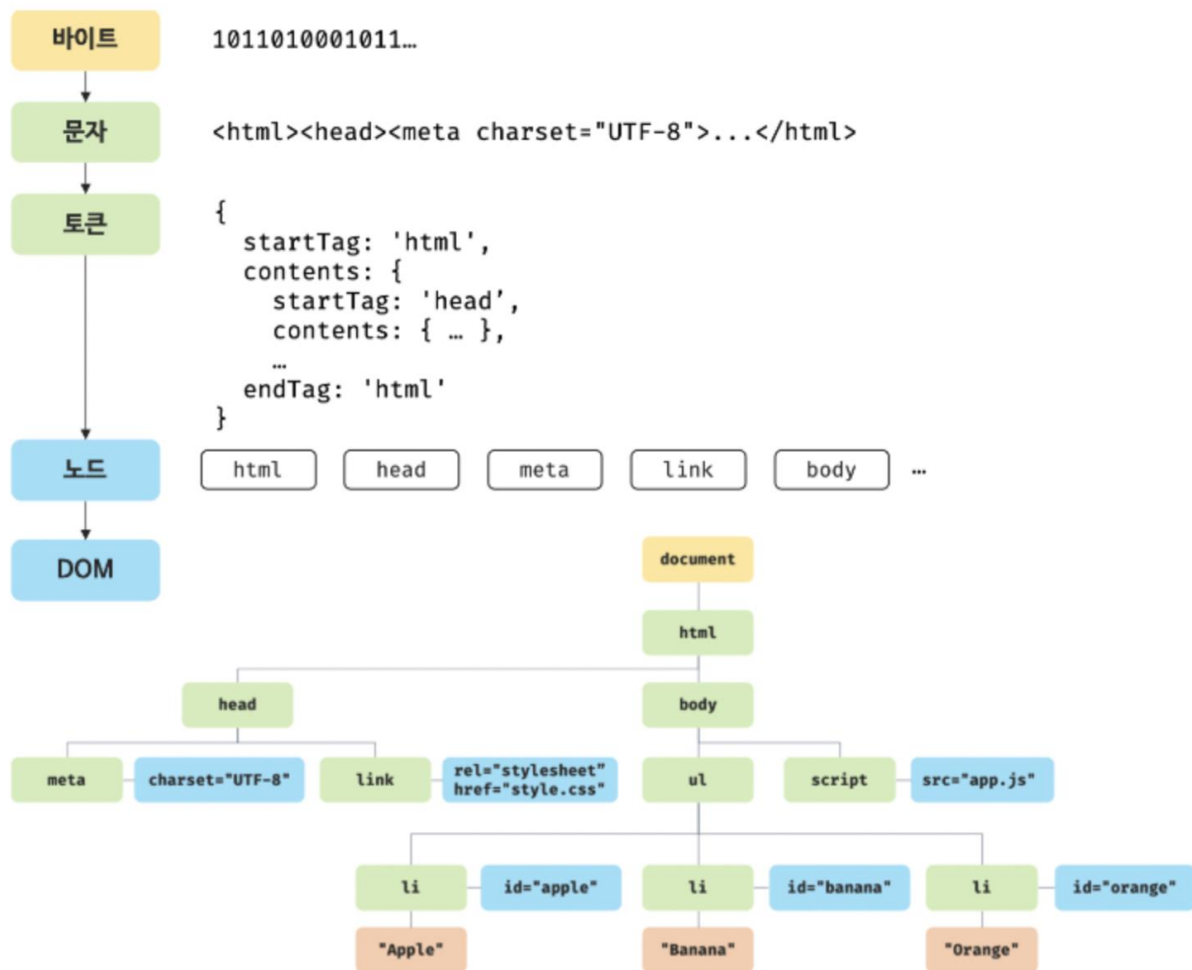
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <link rel="stylesheet" href="./style.css">
  <title>Document</title>
</head>
<body>
  <!-- 어쩌고 저쩌고 html 내용 -->
  <script src="./script.js"></script>
</body>
</html>
```

브라우저의 요청에 의해
서버가 응답한 HTML 문서는
문자열로 이루어진 순수한 텍스트이다.



따라서 브라우저가 이해하기 위해
번역이 필요하다!

HTML 파싱과 DOM 생성



→ 서버는 HTML 파일을 메모리에 저장한 다음 바이트로 응답한다.

→ HTML 파일에 적혀진 인코딩 방식을 확인하고 바이트를 문자열로 변환한다.

→ 문자열로 변환된 HTML 문서를 토큰 단위로 분해한다.
(토큰: 문법적 의미를 갖는 코드의 최소 단위)

→ 각 토큰들을 객체로 변환해 노드를 생성한다.
(노드: 컴퓨터 과학에 쓰이는 기초적인 단위)

→ HTML 문서는 HTML 요소들의 집합으로 이루어지며 HTML 요소는 중첩관계에 의해 부자 관계가 형성된다. 이 부자관계들을 반영해 모든 노드들을 트리 자료구조로 구성한다. 이러한 자료구조를 '**DOM**'이라고 부른다.

HTML 파싱과 DOM 생성

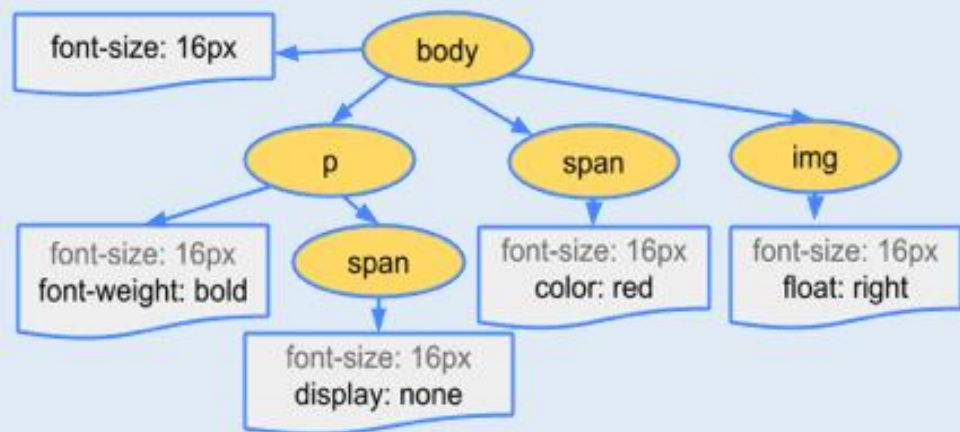
즉, DOM은 HTML 문서를
파싱(분해하고 구조 생성)한
자료구조 형태의 결과물이다.

CSS 파싱과 CSSOM 생성

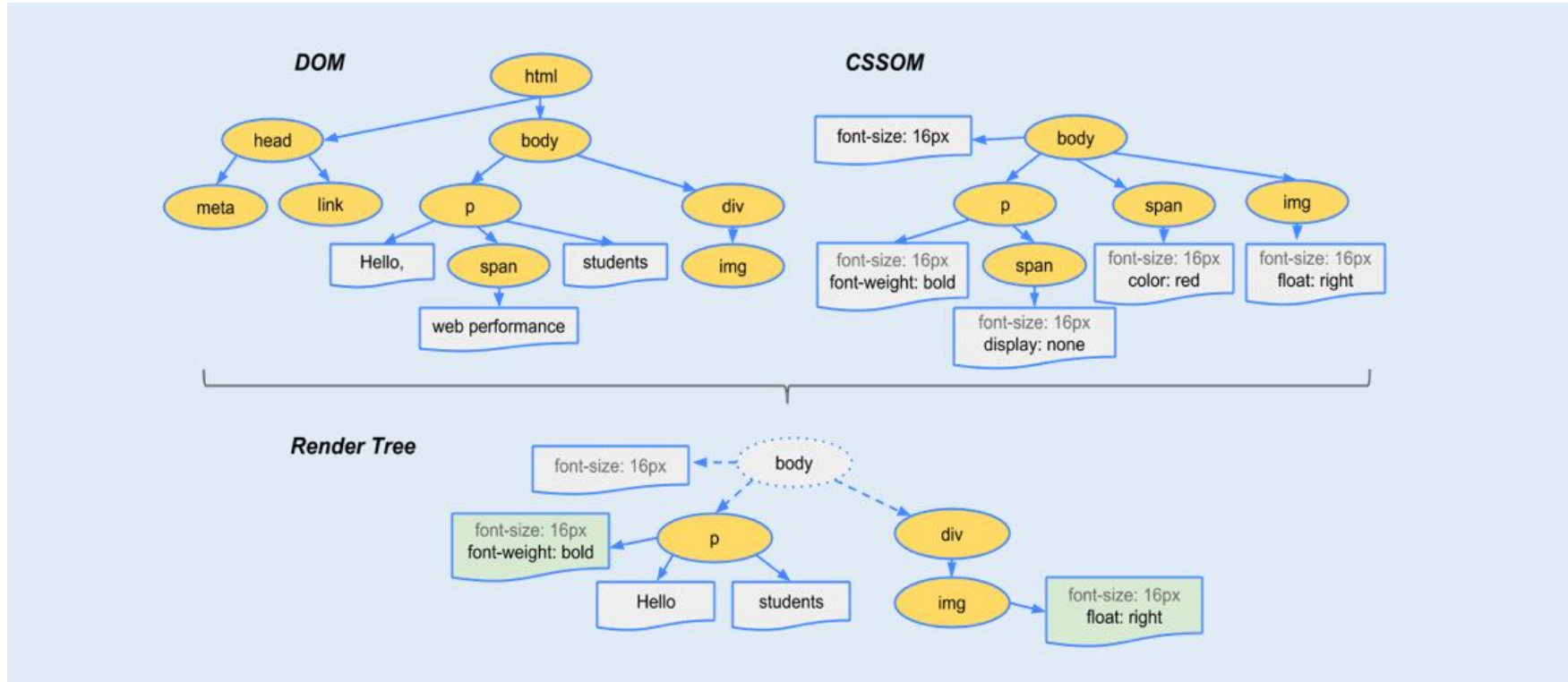
```
<!DOCTYPE html>
<html Lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible"
content="IE=edge">
  <meta name="viewport" content="
width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="
./style.css">
</head>
<body>
  <!-- 어쨌고 이쁘고 이쁘고 -->
  <script src="..."></script>
</body>
</html>
```

DOM 생성
일시 중단

CSSOM



렌더 트리 생성



DOM과 CSSOM은 렌더링을 위해 **렌더 트리**로 결합!

브라우저 렌더링이 반복되는 경우

1

자바스크립트에 의한 노드 추가 또는 삭제

2

브라우저 창의 리사이징에 의한 뷰포트 크기 변경

3

HTML 요소의 레이아웃에 변경을 발생시키는 스타일 변경

브라우저 렌더링이 반복되는 경우

1

자바스크립트에 의한 DOM 조작 또는 삭제

2

브라우저 창의 리사이즈에 따른 뷰포트 크기 변경

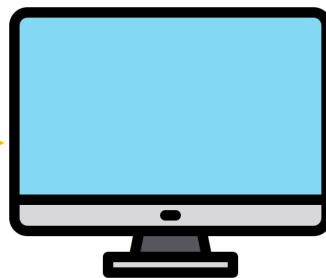


3

리렌더링은 성능에 **악영향**을 주는 작업이므로
리렌더링이 자주 일어나지 않도록 주의해야 한다!

자바스크립트 파싱과 실행

```
<!DOCTYPE html>
<html Lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible"
content="IE=edge">
  <meta name="viewport" content="
width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="
./style.css">
  <title>Document</title>
</head>
<body>
  <!-- 어쩌고 저쩌고 html 내용 -->
  <script src="./script.js"></script>
</body>
</html>
```



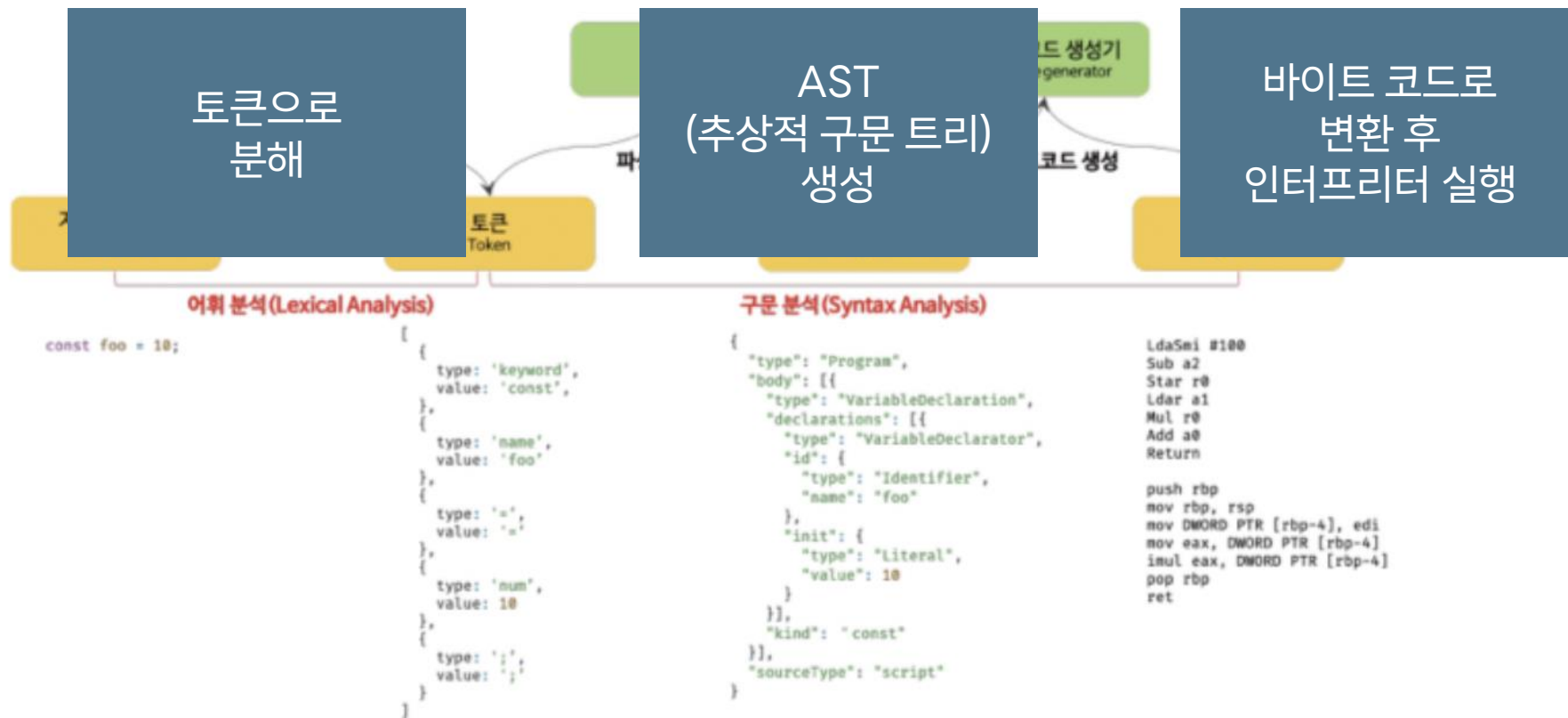
제어권을 넘김



blocking



자바스크립트 파싱과 실행



자바스크립트 파싱에 의한 HTML 파싱 중단

```
<!DOCTYPE html>
<html Lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible"
content="IE=edge">
  <meta name="viewport" content="
width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="
./style.css">
  <title>Document</title>
</head>
<body>
  <!-- 어쩌고 저쩌고 html 내용 -->
  <script src="./script.js"></script>
</body>
</html>
```

script가 body 아래에 있을 때

```
<!DOCTYPE html>
<html Lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible"
content="IE=edge">
  <meta name="viewport" content="
width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="
./style.css">
  <script src="./script.js"></script>
  <title>Document</title>
</head>
<body>
  <!-- 어쩌고 저쩌고 html 내용 -->
</body>
</html>
```

DOM 생성
일시 중단

script가 body 위에 있을 때

자바스크립트 파싱에 의한 HTML 파싱 중단

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="style.css">
    <script>
      const apple = document.getElementById('apple');
      $apple.style.color = 'red';

      // TypeError: Cannot read property 'style' of null
    </script>
  </head>
  <body>
    <ul>
      <li id="apple">Apple</li>
      <li id="banana">Banana</li>
      <li id="orange">Orange</li>
    </ul>
  </body>
</html>
```

자바스크립트 파싱에 의한 HTML 파싱 중단

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <ul>
      <li id="apple">Apple</li>
      <li id="banana">Banana</li>
      <li id="orange">Orange</li>
    </ul>
    <script>
      const $apple = document.getElementById('apple');
      $apple.style.color = 'red';
    </script>
  </body>
</html>
```

자바스크립트 파싱에 의한 HTML 파싱 중단



브라우저의 렌더링 순서 정리 - Construction

- 1 사용자가 브라우저로 웹 사이트에 접속하면 브라우저가 IP 파악
- 2 브라우저가 서버에게 HTTP Request
- 3 서버가 브라우저에게 HTTP Response
- 4 HTML을 파싱하여 DOM Tree 생성
 - 중간에 style 태그가 나오면 CSSOM Tree 생성
 - 중간에 script 태그가 나오면 AST 생성 후 실행
- 5 DOM + CSSOM = Render Tree 생성

브라우저의 렌더링 순서 정리 - Operation

- 6 Render Tree에 있는 node를 배치 (Layout)
- 7 Render Tree에 있는 node의 UI를 그림 (Paint)
- 8 Render Tree에 있는 node를 순서대로 구성 (Composition)
- 9 웹 사용자에게 결과 화면을 출력



자료 출처: 이웅모, 모던 자바스크립트 Deep Dive (위키북스, 2022), p.660 ~ 675
<https://velog.io/@whow1101>, 38. 브라우저의 렌더링 과정
https://youtu.be/hITJM_t1WWY, 웹 브라우저가 하는 일 : 렌더링 6단계 (현업에 적용하는 CS. 4편)
<https://youtu.be/FQHNg9gCWpg>, 웹 브라우저의 동작 순서(6분 순삭)