

브라우저 렌더링 과정과 함께 알아보는 DOM



DOM이 뭐야?



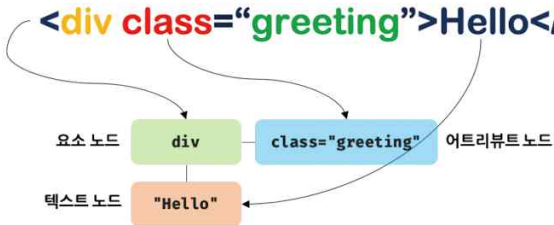
DOM은 뭘까?

Document Object Model 문서 객체 모델 이라고 불리는건 아는데...

**HTML 문서의 계층적 구조와 정보를 표현하며 이를 제어할 수
있는 API를 제공하는 트리 자료구조**

프로퍼티와 메서드 제공

HTML 요소와 **노드 요소**가 무엇인지 먼저 알아보자!



```

<main>
  <!-- section 1 -->
  <section class="section1">
    <h2 class="a11y-hidden">판매 음료</h2>
    <ul class="cola-list"></ul>

    <div class="container">
      <div class="bg-box">
        <h2 class="title">잔액</h2>
        <p></p>
      </div>
      <button class="btn" type="button">거스름돈 반환</button>

      <label for="input-money" class="a11y-hidden">금액 투입(단위:원)</label>
      <input id="input-money" min="1000" step="1000" type="number" placeholder="입금액 입력">
      <button class="btn" type="button">입금</button>

      <h2 class="a11y-hidden">장바구니</h2>
      <ul class="get-list"></ul>
      <button class="btn-get" type="button">획득</button>
    </div>
  </section>
  <!-- //section 1 -->
  <section class="section2">
    <div class="bg-box">
      <h2 class="title">소지금</h2>
      <p><strong>25,000원</strong></p>
    </div>
  </section>
  <section class="section3">
    <h2 class="get-title">획득한 음료</h2>
    <ul class="get-list"></ul>
    <p class="total-price">총금액 : 0 원</p>
  </section>
</main>

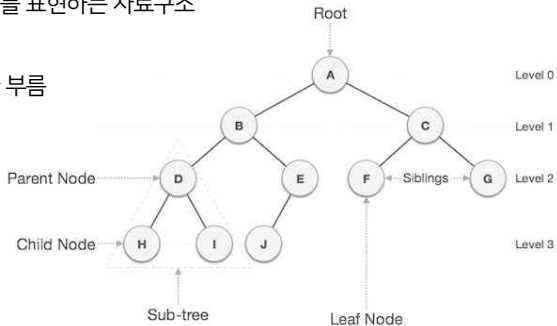
```

HTML 요소 간에는 중첩 관계에 의해
계층적인 부자 관계가 형성

이러한 부자 관계를 반영하여 HTML 요소를
객체화한 모든 노드 객체들을 트리 자료 구조로 구성

트리 자료 구조

- 노드들의 계층 구조로 이루어짐
- 부모 노드와 자식 노드로 구성되어 계층적 구조를 표현하는 자료구조
(비선형 자료구조라는데 어려워서 뽀)
- 하나의 최상위 노드에서 시작하며, 루트 노드라 부름
- 루트 노드는 0개 이상의 자식 노드를 가짐



1. 노드 객체들로 구성된 트리 자료구조를 DOM이라 한다.

=> 트리로 구조화 되어 있기 때문에
DOM 트리라고 부르기도 한다.

노드 객체의 종류

Attribute node

Text node

~~DocumentFragment node~~

~~Entity reference node~~

Document node

Element node

~~CDATASection node~~

~~Processing instruction node~~

~~Comment node~~

~~DocumentType node~~

~~Notation node~~

~~Entity node~~

노드 객체 종류 - 문서 노드 Document node

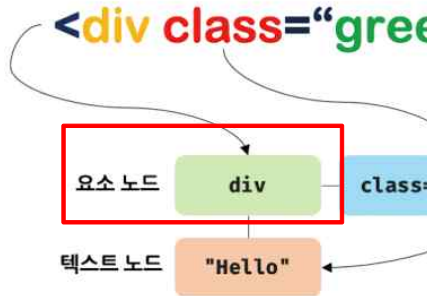
DOM 트리의 최상위에 존재하는 루트 노드로 document 객체를 가리킨다.

브라우저가 렌더링한 HTML 문서 전체를 가리키기 때문에
window.document 나 document로 참조할 수 있다.

HTML 문서 당 document 객체는 유일하다.

DOM 트리의 루트 노드 이므로 다른 노드들의 진입점 역할을 한다.

노드 객체 종류 - 요소 노드 Element node



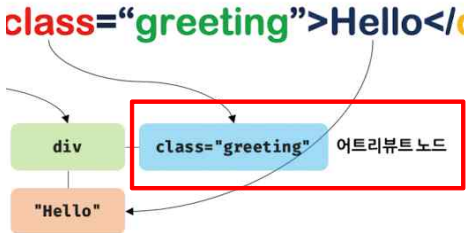
HTML 요소를 가리키는 객체다.

HTML 요소 간의 중첩에 의해
부자 관계를 가진다.

부자 관계를 통해 정보를 구조화한다.

문서의 구조를 표현한다.

노드 객체 종류 - 어트리뷰트 노드 Attribute node



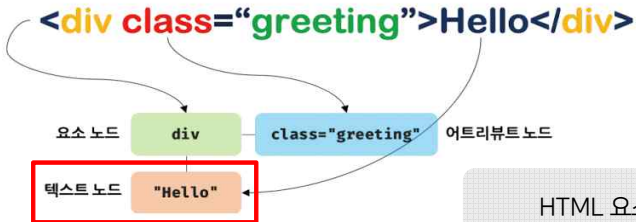
HTML 요소의 어트리뷰트를 가리키는 객체다.

어트리뷰트가 지정된 요소 노드와 연결되어 있다.

부모 노드가 없으므로 요소 노드의 형제 노드가 아니다.

어트리뷰트 노드에 접근하려면 요소 노드에
먼저 접근해야 한다.

노드 객체 종류 - 텍스트 노드 Text node



HTML 요소의 텍스트를 가리키는 객체다.

요소 노드의 자식 노드이며, 리프 노드이다.

DOM 트리의 최종단이다.

텍스트 노드에 접근하려면 요소 노드에
먼저 접근해야 한다.

- `document.getElementById(id)`
- `document.getElementsByTagName(name)`
- `document.createElement(name)`
- `parentNode.appendChild(node)`
- `element.innerHTML`
- `element.style.left`
- `element.setAttribute`
- `element.getAttribute`
- `element.addEventListener`
- [window.content_\(en-US\)](#)
- [window.onload_\(en-US\)](#)
- [window.dump_\(en-US\)](#)
- [window.scrollTo_\(en-US\)](#)

DOM은 HTML 문서의 계층적 구조와 정보를 표현하면서,
노드 객체에 따라 필요한 기능을 프로퍼티와 메서드의 집합인
DOM API로 제공한다.

우리는 이 DOM API를 통해 HTML 구조나 내용
또는 스타일 등을 동적으로 조작할 수 있다.

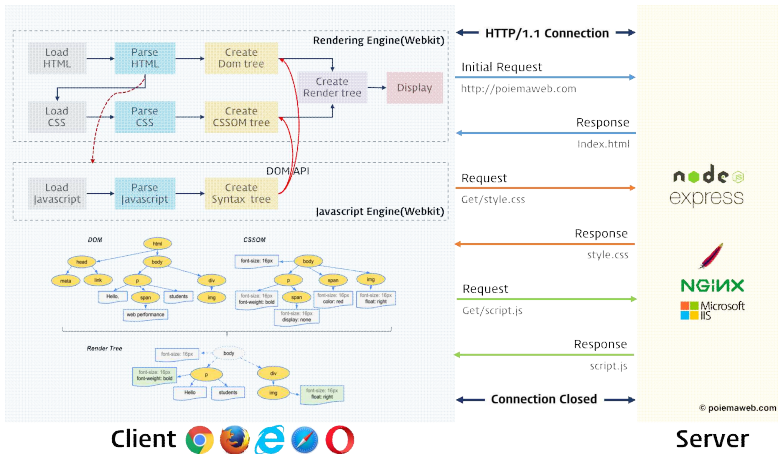
브라우저는 사용자에게 어떻게 보여주는걸까?

파싱하여 브라우저에 시각적으로 출력하는 것

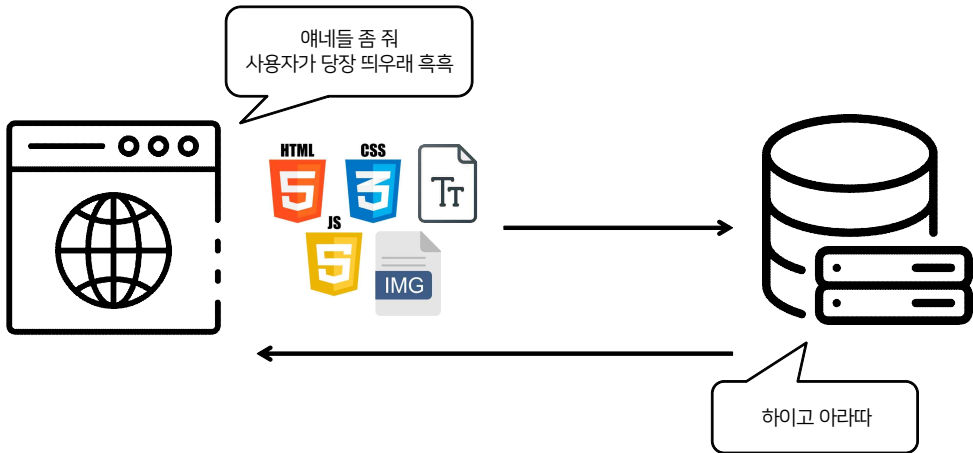
브라우저가 HTML, CSS, 자바스크립트로 작성된 텍스트 문서를
파싱하여 브라우저에 렌더링 한다.



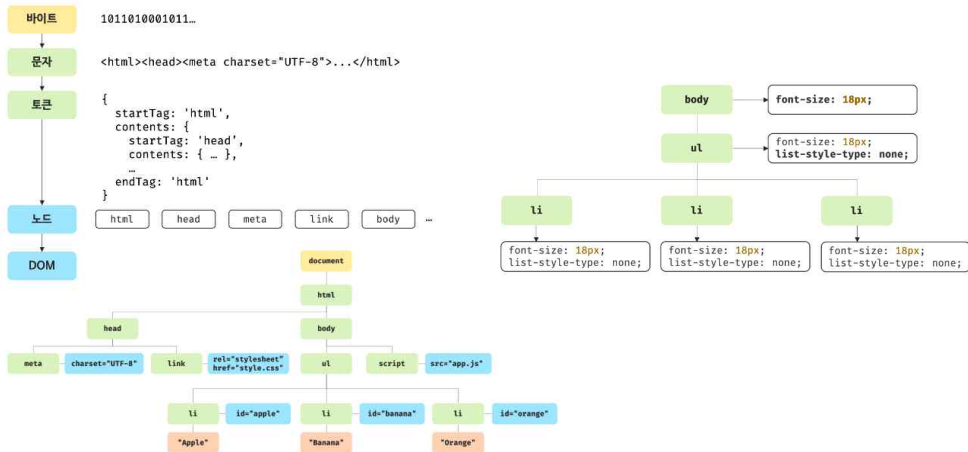
렌더링 과정을 알아보자!



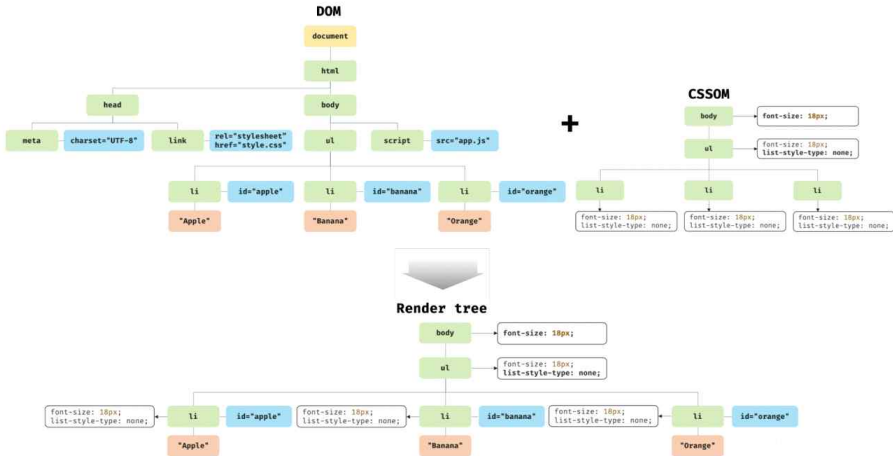
브라우저는 렌더링에 필요한 리소스를 요청하고 서버로부터 응답을 받는다.



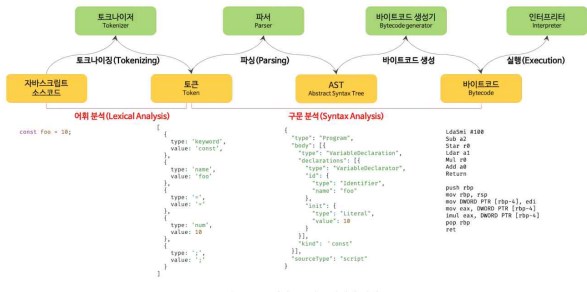
서버로부터 응답된 HTML과 CSS를 파싱하여 DOM과 CSSOM을 생성한다.



DOM과 CSSOM을 결합하여 렌더 트리를 생성한다.



자바스크립트 엔진이 자바스크립트를 파싱하고 실행한다.



뭐야 왜요

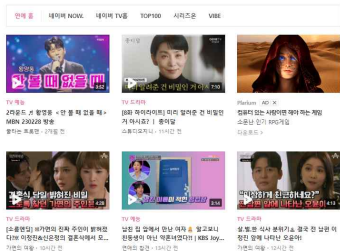
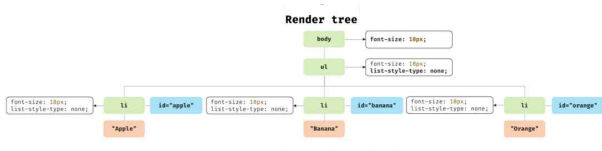
패스

DOM API로
DOM이나 CSSOM 변경

변경된 DOM, CSSOM

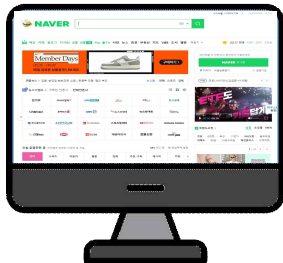
다시 렌더트리에 결합

렌더트리를 기반으로 HTML 요소의 레이아웃을 계산하고 화면에 페인팅한다.



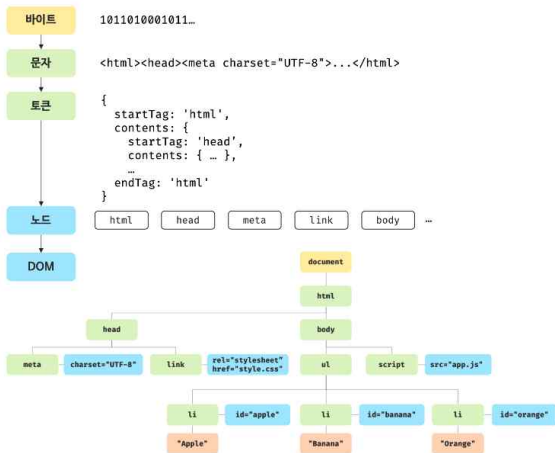
저히 들어가도 대요?

오 ㅋㅋ 딱 좋겠는데?
들어와



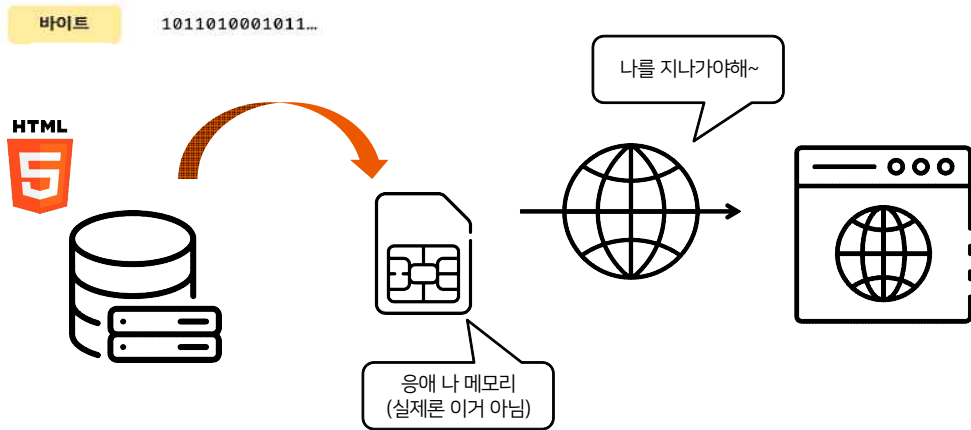
짤

HTML 파싱과 DOM 생성



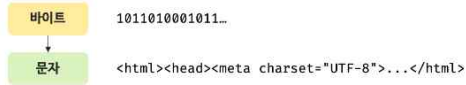
1. 서버에 존재하던 HTML 파일이 브라우저 요청에 의해 응답된다.

서버는 브라우저가 요청한 HTML 파일을 읽어서 메모리에 저장한 다음 인터넷을 경유하여 응답한다.



2. 브라우저는 서버가 응답한 HTML 문서를 바이트 형태로 응답 받는다.

응답된 바이트 형태의 HTML 문서는 meta 태그의 charset 어트리뷰트에 의해 지정된 인코딩 방식을 기준으로 문자열로 변환 된다.



```
<!DOCTYPE html>
<html lang="ko" data-dark="false" data-useragent="Mozilla/5.0 (Win
WebKit/537.36 (KHTML, like Gecko) Chrome/113.0.0.0 Safari/537.36">
  <head> ... </head>
  <body cz-shortcut-listen="true">
    <div id="u_skip">...</div> == $0
    <div id="wrap">
      <style>...</style>
      <div id="NM_TOP_BANNER" data-clk-prefix="top" class="_1hiMwem
ff7e1">...</div>
      <div id="header" role="banner">...</div>
      <div id="container" role="main">...</div>
      <div id="footer" role="contentinfo">...</div>
    </div>
    <div id="adscript" style="display:none"></div>
  </body>
</html>
```

3. 문자열로 변환된 HTML 문서를 읽어 들여 토큰들로 분해한다.

문법적 의미를 갖는 코드의 최소 단위



4. 각 토큰들을 객체로 변환하여 노드들을 생성한다.

문서 노드

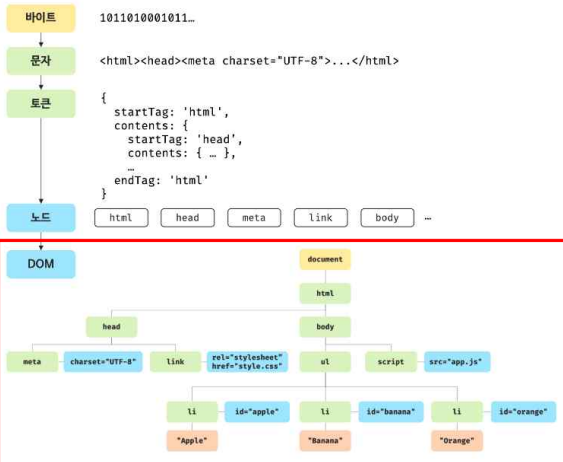
요소 노드

어트리뷰트 노드

텍스트 노드

=> 노드들은 DOM 구성의 기본 요소가 됨

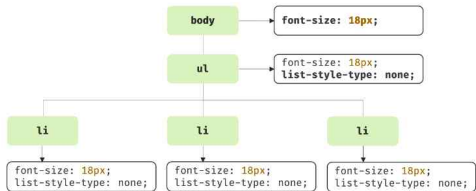
5. HTML 문서는 요소들의 집합으로 구성되어 있어 중첩 관계에 의해 부자 관계가 형성되는데 이를 반영하여 모든 노드들은 트리 자료구조로 구성한다.



=> 이게 바로 DOM!!!

2. DOM은 HTML 문서를 파싱한 결과물이다.

CSS 파싱과 CSSOM 생성



바이트 > 문자 > 토큰 > 노드 > CSSOM



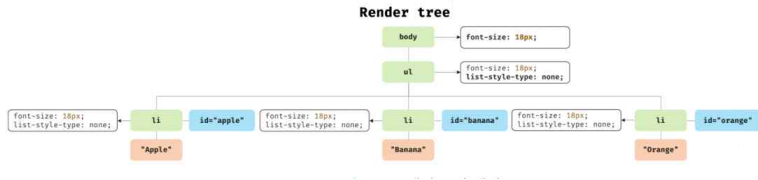
CSS도 HTML과 동일한 파싱 과정을 거치며 CSSOM을 생성

HTML을 한 줄씩 순차적으로 파싱하면서
DOM을 생성하다 CSS를 로드하는
link 태그나 style 태그를 만나면
DOM 생성을 일시 중단

CSS 파싱

완료 후 HTML 파싱 중단 지점부터
다시 HTML 파싱 시작

렌더 트리 생성



생성된 DOM과 CSSOM은 렌더링을 위한 렌더 트리로 결합

렌더 트리? 렌더링을 위한 트리 구조의 자료구조

브라우저 화면에 렌더링되지 않는 노드(meta 태그, script 태그 등), CSS에 의해 비표시(display: none)되는 노드들은 포함 하지 않음

=> 렌더 트리는 브라우저 화면에 렌더링되는 노드만으로 구성

DOM 이제 알 것 같나요? (제발)

**HTML 문서의 계층적 구조와 정보를 표현하며 이를 제어할 수
있는 API를 제공하는 트리 자료구조**

감삼당 아직 말할게 한가득인데 흑흑

reference

- 모던 자바스크립트 Deep Dive
- 잡았다 요 DOM!
- 웹페이지가 사용자에게 보여지기까지 (브라우저 렌더링 과정)
(<https://jooong.tistory.com/entry/rendering>)