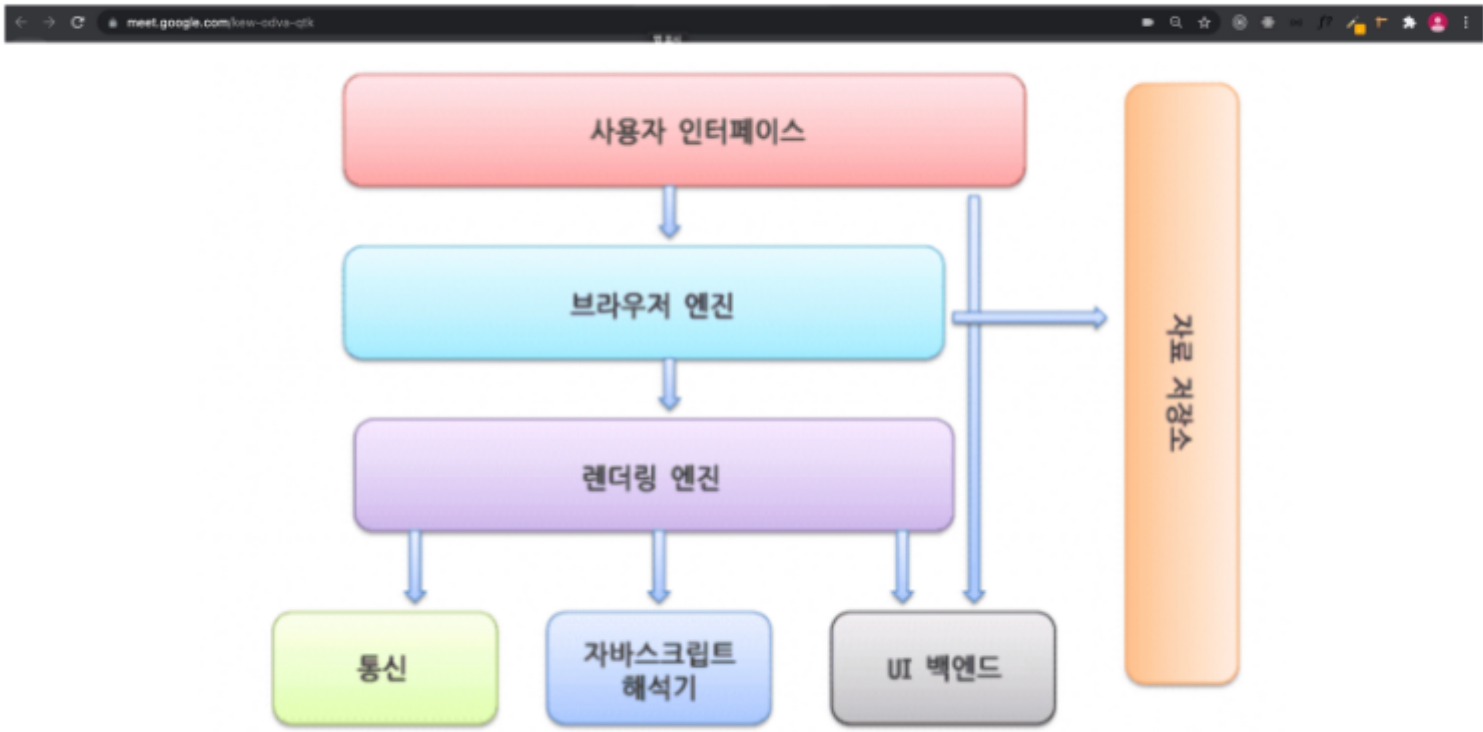


브라우저 작동 원리

| | |
|--------|-------------|
| Status | Not started |
| Assign | 웹 |

브라우저(Brouser)의 기본구조

Browser elements



사용자 인터페이스

뒤로가기, 주소창, 새로고침과 같이 페이지 창 외에서 컨트롤 할수 있는 부분을 말합니다.

브라우저 엔진

사용자 인터페이스와 렌더링 엔진 사이에서 중간다리 역할

자료저장소

웹의 데이터를 클라이언트에 저장할 수 있는 저장소이다.

☐ 쿠키(cookie)

- 웹사이트에 의해 유저의 정보를 저장하는 곳
- 서버와 데이터를 공유하는 용도로 사용되며 데이터의 유효기간을 지정이 가능

☐ 영구저장소(LocalStorage)

- 저장한 데이터를 명시적으로 지우지 않은 이상 영구적으로 보관이 가능
- 도메인마다 별도로 LocalStorage가 생성된다.

☐ 임시저장소(SessionStorage)

- 브라우저가 열려있는 한 페이지를 Reload해도 저장소 속 데이터는 계속 유지되지만, 브라우저를 닫으면 삭제된다.

렌더링 엔진

요청한 url을 브라우저 엔진에게 받아 서버에게 요청을 하고, 그 요청한 데이터를 응답받아 html,css,js를 파싱한 후 렌더링 합니다.

통신

렌더링 엔진으로부터 http 요청 등을 받아서 네트워크 처리 후 응답을 전달합니다.

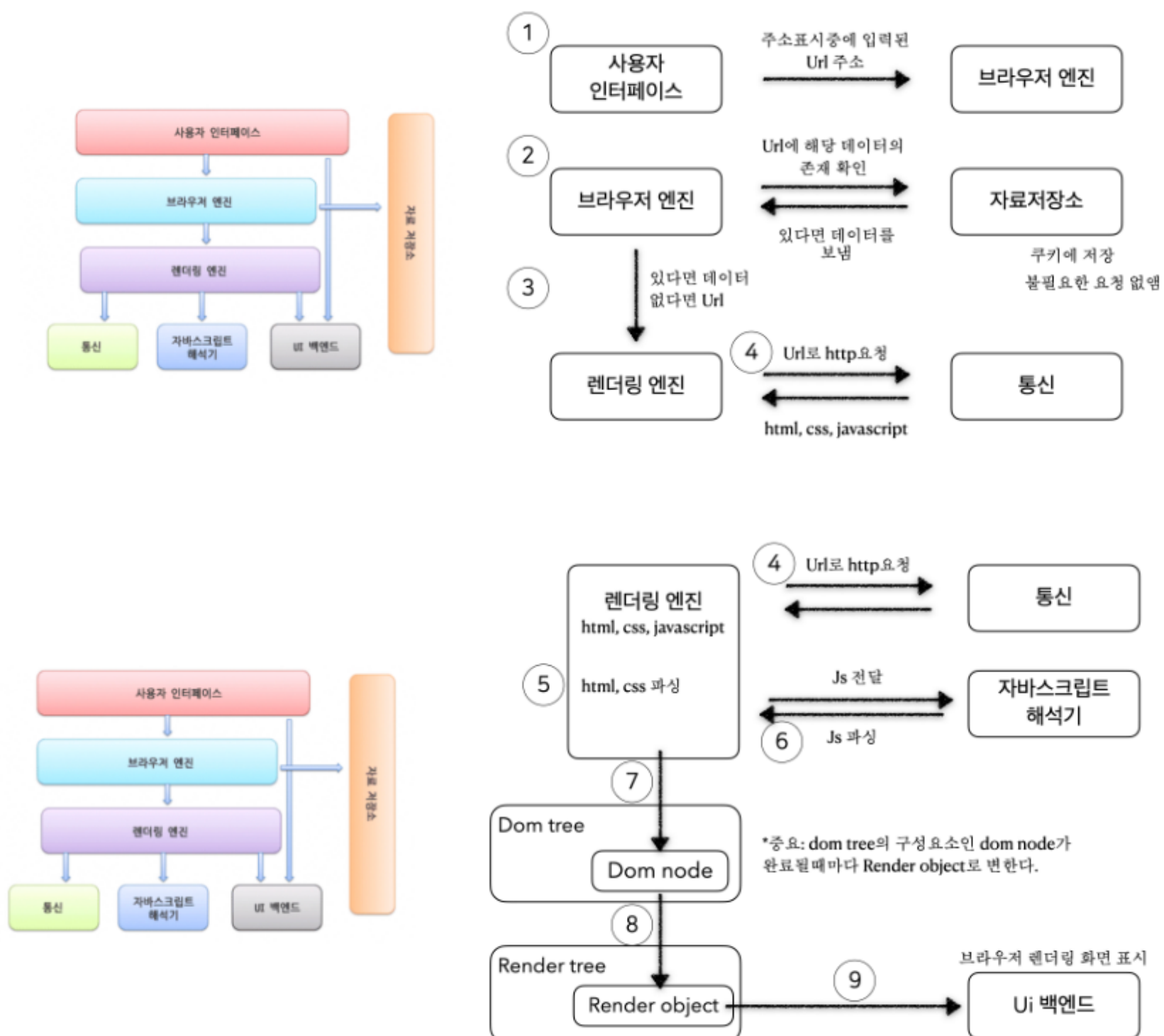
자바스크립트 해석기

js파일을 파싱합니다.

UI 백엔드

렌더 트리를 브라우저에 그리는 역할을 담당합니다

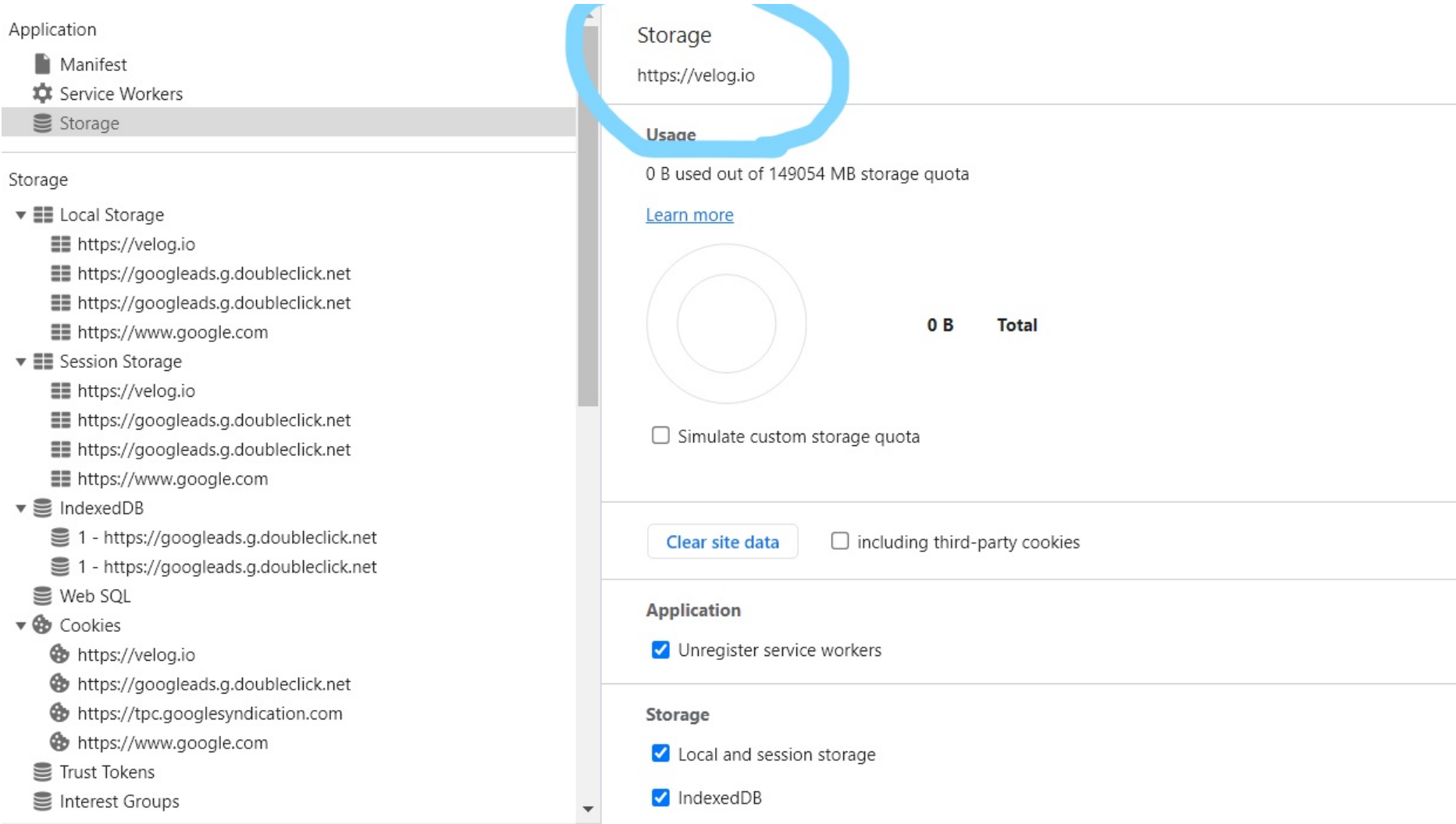
브라우저 렌더링 과정



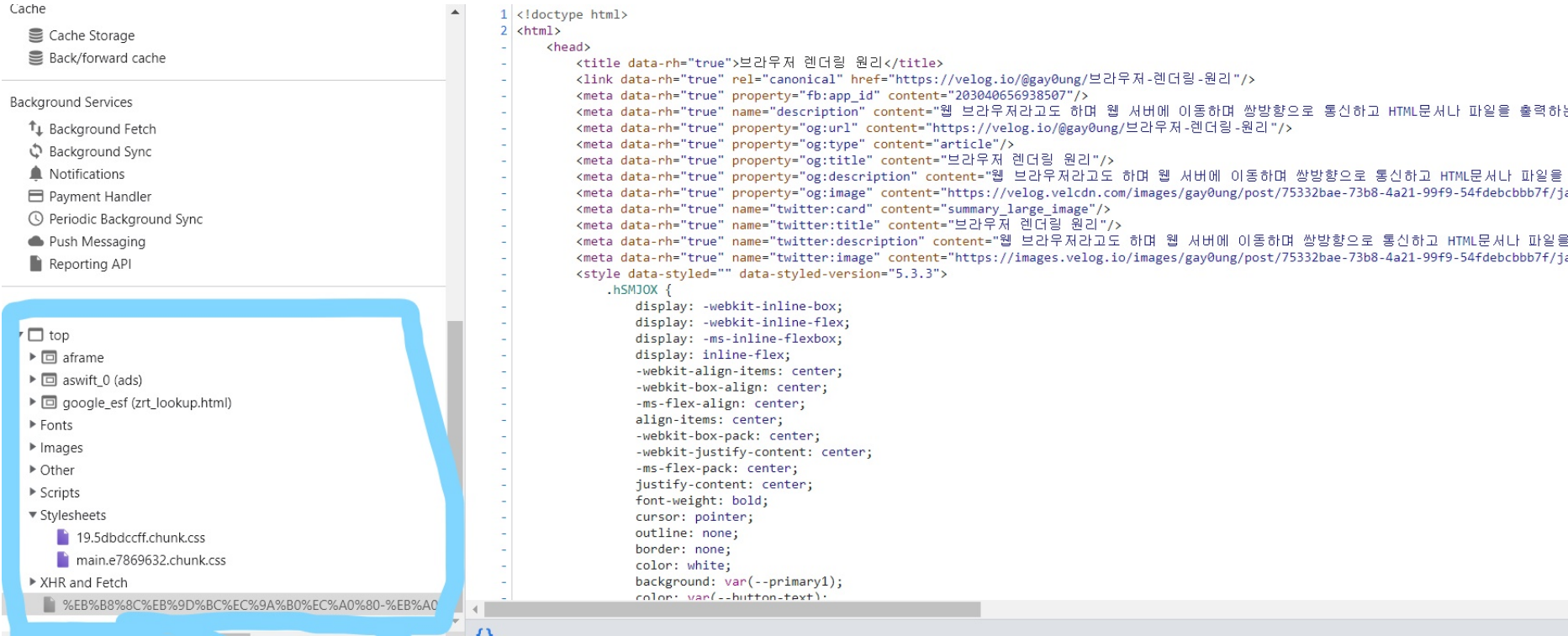
1. 브라우저 주소창에 웹사이트 주소(www.yellosis,.com/login)를 을 친다.
2. 브라우저 엔진은 자료 저장소에서 URI에 해당하는 자료를 찾고, 찾은 자료를 렌더링 엔진에 전달한다.

예) 해당URI(www.yellosis.com/login)의 영구저장소 속에 로그인 인증 토큰 데이터가 있는 확인 후 있다면 토큰의 값을 렌더링엔진으로 DNS(실제 서버가 어디에 있는지 알 수 있는 서버주소)와 토큰 데이터를 보낸다.

자료저장소 속 URI(DNS)



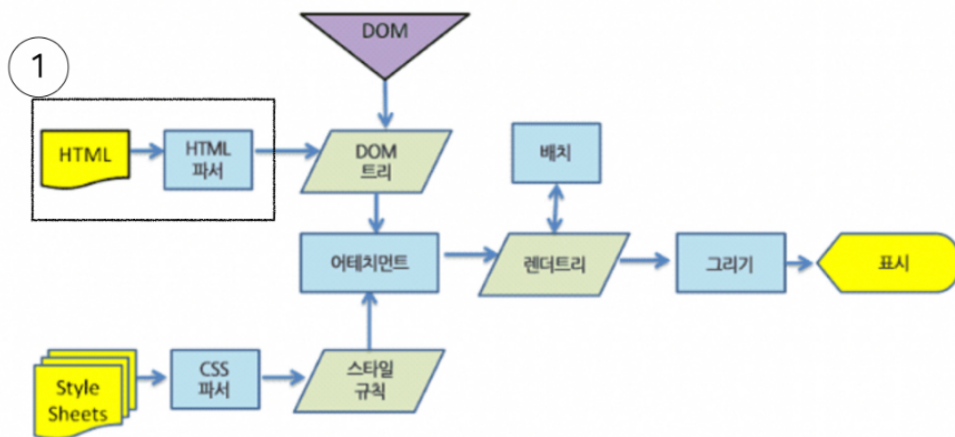
CSS, Javascript, HTML파일을 렌더링엔진으로 데이터를 보낸다.



3. 렌더링 엔진은 브라우저 엔진에서 가져온 자료(HTML, CSS, Image등)를 분석한다. 동시에 URI 데이터를 통신, 자바스크립트 해석기, UI 백엔드로 전파한다 브라우저 엔진에서 가져온 자료를 분석하여h HTML, CSS파일은 렌더링 엔진이 파싱한다. Javascript파일은 Javascript해석기를 보내며, Javascript해석기가 파싱한다. URL로 HTTP통신요청을 한다. 해석된 Javascript 결과를 다시 렌더링 엔진에 전달하여 전달한 DOM tree을 조작한다.

Rendering engine working process

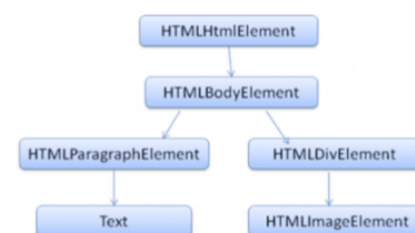
webkit 의 동작 과정



1 html을 parsing하여 DOM tree를 생성

```

<html>
<body>
  <p>Hello World</p>
  <div>
    
  </div>
  <script></script>
</body>
</html>
    
```



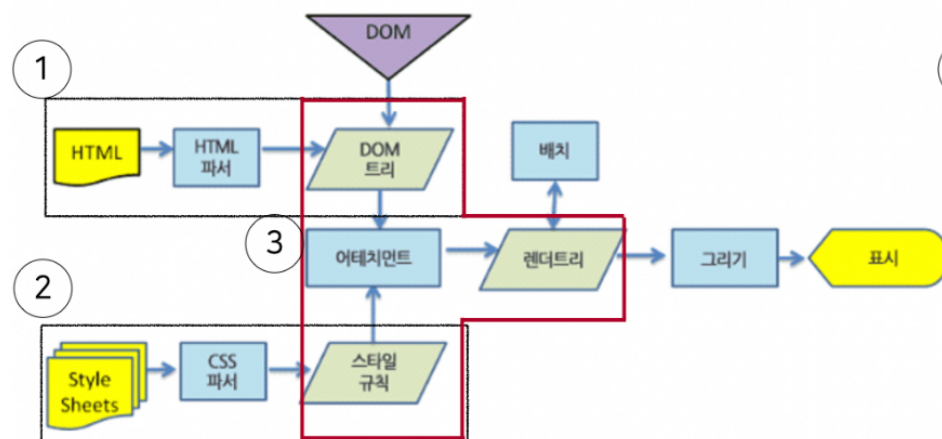
Script 태그

chrome 과 IOS 는 webkit 이라는 rendering engine 을 사용

1. 조작이 완료된 DOM node(DOM tree구성요소)는 render object(render tree 구성요소)로 변한다.

Rendering engine working process

webkit 의 동작 과정



2 CSS(style sheets)를 parsing하여 스타일 규칙을 얻음

3 Dom tree를 생성하는 동시에, dom tree와 스타일 규칙을 attachment 한다

Dom node는 attach라는 method를 가지게 됨

새로운 dom node가 추가되면, attach가 호출되어 Render object를 생성

렌더 뷰에 차근차근 쌓임

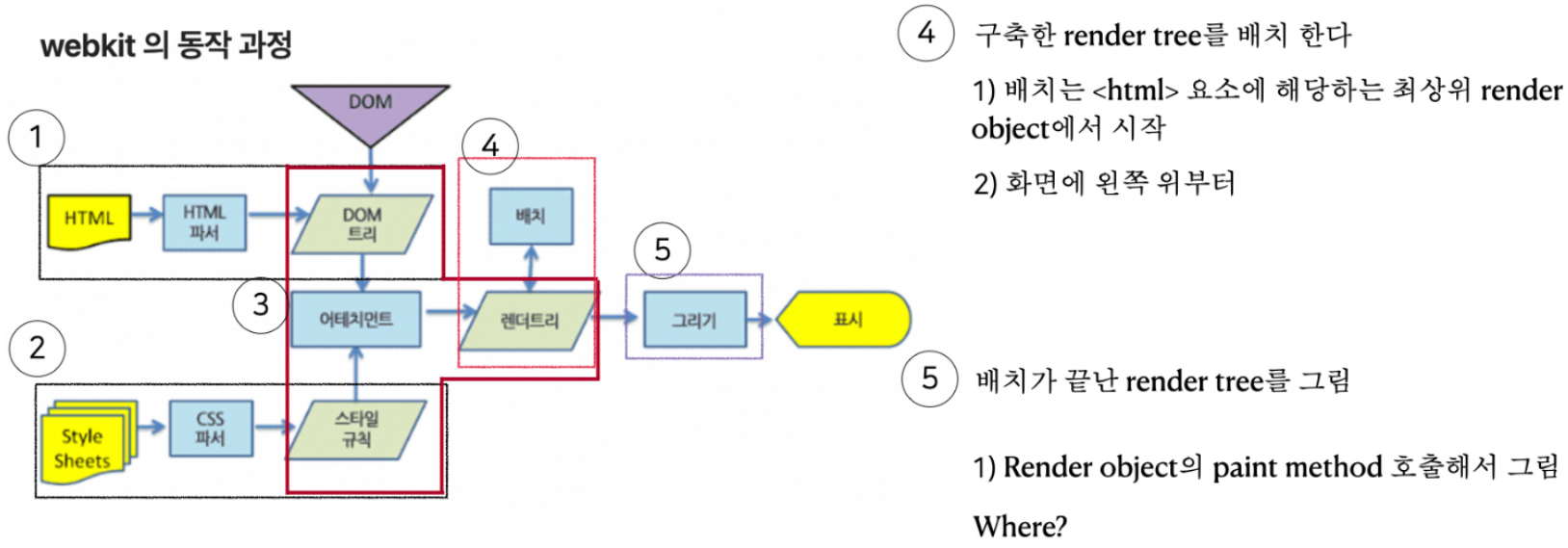
<html>, <body> dom node 또한 render object로 구성되는데 이들은 render tree root로써 render view라고 부른다. Why? 가장 위에 있기에

모든 dom node가 Render object로 변환되는 것은 아님

chrome 과 IOS 는 webkit 이라는 rendering engine 을 사용

1. UI 백엔드는 render object를 브라우저 렌더링 화면에 띄어준다.

Rendering engine working process



chrome 과 IOS 는 webkit 이라는 rendering engine 을 사용