브라우저에 대해 개발자들이 관심을 가지게 된 이유

우리의 소스코드를 어떻게 하면 더 빨리 실행시킬 수 있을까?

브라우저의 구성요소

1. User Interface - OS마다 다르다. ex) 뒤로가기 버튼, X버튼 등

2. 브라우전 엔진 - 소스코드(HTML, CSS, JAVAScript 등)를 해석해서 화면에 보여주는 역할

3. 렌더링 엔진 - 실질적으로 그리기를 담당 ex) Chrominum, Gecko, Webkit 등

4. 데이터 관리 영역 - 일부 데이터 저장

5. 네트워크 - 통신을 담당

6. JavasScript Interpreter - 자바스크립트 해석 담당

7. UI Backend - UI 영역을 처리할 수 있는 백엔드 영역 -------- 이해가 안 된다.

브라우저의 주요 흐름

1. HTML을 DOM(Document Object Model)로 변환 ---> Dom은 트리구조이다.

2. Render tree construction - 렌더 트리 생성

3. Layout of the render tree - 렌더트리를 기준으로 CSS 정보를 합친 것

4. Painting the render tree - 화면에 직접 그림을 그린다.

자세히 살펴보자

1. 뼈대를 가진 HTML을 DOM Tree로 변환한다.

2. 또한 HTML이라는 뼈대를 색칠해줄 CSS 파일을 DOM Tree와 합쳐서 Render Tree를 생성한다. ---- 각각의 DOM Tree가 어떤 스타일 정보를

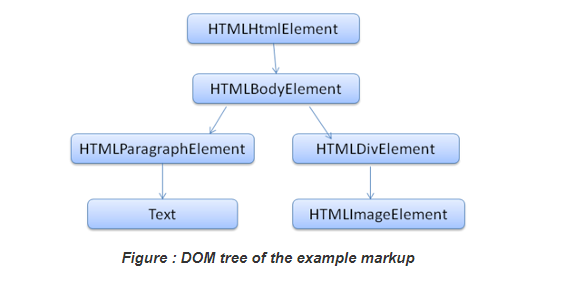
가지고 있는지 Render Tree에 저장된다.

3. 화면에 실제로 그려준다.

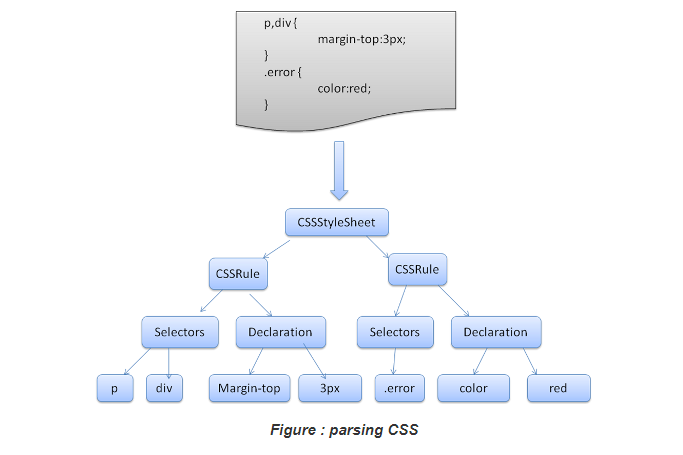
파싱은 어떻게 하는 걸까???????????????

컴퓨터가 토큰(쉽게 생각하면 의미가 있는 뭉텅이)을 기준으로 Syntax Tree를 만든다.

Syntax Tree에 따라서 값에 대한 처리가 일어난다.

HTML에서는 Tag를 Token으로 생각하면 조금 더 이해하기 쉬울 거 같다.

CSS에서는 키와 Value값으로 파싱을 하게 된다.

Selectors라는 키에 Declaration이라는 Value가 들어가게 된다고 생각하면 편한 것 같다.