
브라우저의 렌더링 원리

브라우저 렌더링은 주로 렌더링 엔진에 의해 진행됩니다.

렌더링 엔진은 브라우저에 따라 다릅니다. 구글은 blink, 사파리는 webkit이 있습니다.

렌더링 엔진은 CRP의 프로세스로 작동을 하는데요 CRP는 Critical Rendering Path의 약자입니다.

먼저, HTML을 파싱 후 DOM트리를 구축하고
CSS를 파싱 후 CSSOM트리를 구축합니다.

그 후 Javascript를 실행하고

DOM 트리와 CSSOM트리가 조합된 렌더트리를 구축합니다.

렌더트리는 레이아웃이라는 단계를 거치는데 이는 뷰포트에 맞게 각 노드들의 크기와 위치를 조정하는 단계입니다.

이 단계 후 페인팅이라는 단계가 이루어집니다. 페인팅은 레이아웃에 맞게 웹 브라우저에 해당 노드들을 그려냅니다.

Reflow vs Repaint

Reflow는 레이아웃이 재형성되는 단계입니다. 노드요소들의 크기나 위치가 변경될 때, 또는 브라우저의 창 크기가 변경될 때 발생합니다.

Repaint는 레이아웃수치가 변경되지않아도 색깔같은 것들이 변경되면 발생합니다.

주소창에 google.com을 입력하면 일어나는 일

1. 사용자가 브라우저에 URL을 입력하면 해당 URL을 DNS서버로 보냅니다.
2. 그럼 DNS에서 URL과 일치하는 IP주소를 URL과 함께 다시 보내줍니다.
3. 그럼 브라우저는 http 프로토콜을 사용하여 요청메시지를 생성하고 TCP통신을 통해 요청메세지는 서버로 전송됩니다.
4. 서버는 Response 메시지를 생성하여 다시 브라우저에게 데이터를 전송합니다.
5. 브라우저는 response를 받아 파싱하여 화면에 렌더링합니다.

호이스팅이란?

호이스팅이란 스코프 내에서 함수나 변수가 맨 위로 올라온 거 같은현상을 말합니다.

이는 자바스크립트해석기가 스코프단위로 코드를 스캔을 먼저한 후 함수와 변수를 저장 및 초기화하면서 발생합니다.

즉, 렉시컬스코프 내에서 선언이 끌어올려집니다.

렉시컬 스코프란?

렉시컬 스코프는 함수가 실행될 때 스코프가 형성되는 것이 아닌 함수가 선언될 때 형성된다는 말입니다.

클로저란?

함수와 변수는 함수 밖에서 내부로는 참조가 불가능합니다. 하지만 클로저란 이게 가능한 현상입니다.

예를 들어 내부 스코프에 a라는 변수가 있고 그 a변수가 스코프 내부에서 return이 되었을 경우 바깥 스코프에서 해당 a변수를 사용하게 됩니다.

클로저란 해당 스코프를 기억해주는 현상입니다.

REST API란?

API를 만들때 REST원칙을 지켜 만든 API를 RESTAPI라고 합니다

RESTful한 API는 한눈에 봐도 어떤 작동을 하는 API인지 알 수 있습니다.

보통 http메소드를 통해 CRUD를적용합니다.

this란?

this는 실행컨텍스트가 형성될 때 바인딩됩니다. 이를 활용해서 MVC 디자인패턴의 todolist를 만들었는데

class는분리하여 개발을 할 때 사용했습니다. View class에서 매개변수를 this에 저장해 class내부의 메서드가 모두 해당 매개변수를 사용하도록 하였습니다.

브라우저 저장소의 차이점

브라우저의 저장소는 Localstorage, sessionStorage, cookie가 있습니다.
localStorage와 sessionStorage는 web storage로 묶을 수 있습니다.

webstorage는 약 5mb의 크기를 갖습니다.

반면 cookie는 4kb의 크기를 갖고 string형태의 데이터를 저장할 수 있습니다.

localStorage는 지우지 않는 한 영구저장이 가능하고
sessionstorage는 세션이 종료되면 클라이언트에 대한 정보가 제거됩니다.

cookie는 서버에서 클라이언트로 전송되어 클라이언트에 저장되는 작은 데이터조각입니다. 주로 사용자 인증을 위해 사용됩니다. 클라이언트가 서버에 HTTP 요청을 보내면 쿠키도 함께 전송되어 서버에서 쿠키를 확인할 수 있습니다.

SPA, CSR, SSR

SPA는 single page application의 약자로 html파일이 1개이고 이를 최초 1회 로딩한 후 데이터만 변경해가며사용 할 수 있는 웹 어플리케이션 입니다.

이를 렌더링하는 방식은

csr입니다. 사용자가 웹 브라우저에 접속해 콘텐츠를 요청하면 서버에서는 빈뼈대 HTML과 연결된 js링크를넘겨줍니다. 그럼 클라이언트는 js를 다운로드 받고 동적으로 DOM을 생성하여 그려줍니다.

ssr은 사용자가 웹사이트에 요청을하면 서버는 렌더링이 가능한 html파일을 즉시 만듭니다. 그리고 바로 서버는 해당 html 파일과 js코드를 보내줍니다. 그리고 클라이언트에서 다 그려진 HTML 파일에 js를 다운받아 컴파일합니다.

이때 잠시 인터렉션이 일어나지 않을 수 있습니다.