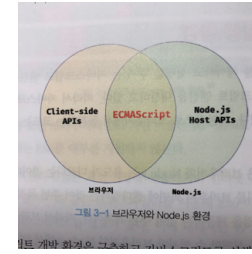


3장 자바스크립트 개발 환경과 실행 방법

3.1 자바스크립트 실행 환경



자바스크립트 엔진 (ECMAScript 실행가능)

Node.js

브라우저 외부에 JS 실행환경 제공이 목적

DOM 제공 x

파일 생성, 수정의
파일시스템(Web API) 제공 o

HTML, CSS, JS 을 실행해
렌더링하는 것이 목적

DOM API 제공 o

파일시스템(Web API) 제공 x

보안상의 이유로

클라이언트 사이드 Web API 지원 — BOM, Canvas, XMLHttpRequest, requestAnimationFrame, SVG, Web Storage, Web Component, Web Worker

** 웹크롤링 — 서버에서 웹사이트의 콘텐츠를 수집하기 위해 웹사이트 HTML을 가공해서 필요한 데이터만 추출하는 것

3.2 웹 브라우저

보통 크롬을 사용

3.2.1 개발자 도구

- Elements — DOM과 CSS 편집, 렌더링된 뷰 확인
- Console. — 에러 확인, console.log 확인
- Sources — Js 코드 디버깅 가능
- Network — 웹페이지와 관련된 네트워크 요청정보와 성능 확인
- Application — 웹 스토리지, 세션, 쿠키를 확인 및 관리

3.2.2 콘솔(Console)

- 에러가 발생했을 때 가장 먼저 봐야할 곳
- 코드를 직접 입력해서 그 결과를 확인할 수 있는 REPL(Read Eval Print Loop; 입력 수행 출력 반복)환경으로 사용할 수 있다.

3.2.3 브라우저에서 자바스크립트 실행

에러 알림, 에러 발생 위치, 에러 정보

3.2.4 디버깅

개발자 도구 Sources 패널에서 디버깅

3.3 Node.js

3.3.1 Node.js와 npm 소개

클라이언트 사이드, 즉 브라우저에서 동작하는 간단한 웹 애플리케이션은 브라우저만으로 개발할 수 있다. 하지만 프로젝트 규모가 커질 수록 여러 프레임워크(리액트, 뷰..), 라이브러리를 도입하거나 도구(Babel, Webpack, ESLint)를 사용할 필요가 있다. 이때 Node.js와 npm이 필요

3.3.2 Node.js 설치

버전 확인 : 터미널에서 node -v , npm -v 로 확인

3.3.3 Node.js REPL

- 터미널에서 REPL 기능 사용 : \$ node 입력
- 자바스크립트 파일 실행 : \$ node 파일명(.js 생략가능)
- Ctrl + C 두 번 => Node.js REPL 종료

3.4 비주얼 스튜디오 코드

3.4.1 비주얼 스튜디오 코드 설치

3.4.2 내장 터미널

3.4.3 Code Runner 확장 플러그인

입력한 코드를 터미널을 거치지않고 단축키로 바로 실행할 수 있도록 하는 확장 플러그인 (contorl + option + N)

비주얼 스튜디오 코드에서 alert 함수를 사용하면 에러가 뜬다. — 이는 브라우저에서 사용하는 Web API 이기때문에 에러가 뜨는 것이다.

3.4.4 Live Server 확장 플러그인

HTML을 바로 띄울 수 있는 확장 플러그인