



# Ch.3 자바스크립트 개발 환경과 실행 방법

## ▼ 🌟 브라우저 vs Node.js

p18(e52)

브라우저와 Node.js 모두 자바스크립트의 코어인 ECMAScript를 실행 할 수 있지만 ECMAScript 이외에 각각 추가로 제공하는 기능은 호환되지 않는다.

자바스크립트 개발환경을 제공하는 것이 주 목적인 node.js는 DOM API를 제공하지 않는다. 브라우저 외부 환경에서는 html요소를 파싱해서 객체화한 Dom을 직접 다룰 필요가 없기 때문이다.

**브라우저는 웹 페이지를 렌더링하고 사용자와 상호작용하는 데 DOM API를 사용한다.** 하지만 Node.js는 이러한 작업을 수행할 필요가 없으므로, DOM API 대신 파일 시스템 접근, 네트워크 통신, 프로세스 관리 등의 서버 측 작업에 필요한 API를 제공한다.

+웹 애플리케이션의 자바스크립트는 사용자 컴퓨터의 브라우저에서 동작한다. 만약 브라우저를 통해 다운로드되어 실행되는 자바스크립트가 사용자 컴퓨터의 로컬 파일을 삭제하거나 수정하고 생성할 수 있다면 이는 사용자 컴퓨터가 악성 코드에 그대로 노출된 것과 마찬가지다. 따라서 보안상의 이유로 브라우저 환경의 자바스크립트는 파일 시스템을 제공하지 않는다.

+브라우저 외부 환경에서 HTML 요소를 파싱하고 객체화한 DOM을 다루고자 할 때는, Node.js에서는 'jsdom' 같은 외부 라이브러리를 사용할 수 있습니다. 이러한 라이브러리를 사용하면 서버 측에서도 DOM 조작을 할 수 있습니다.

## ▼ 웹 크롤링

p17(e51)

서버에서 웹사이트의 콘텐츠를 수집하기 위해 웹사이트에서 HTML 문서를 가져온 다음, 이를 가오해서 필요한 데이터만 추출하는 경우가 있다. 이를 웹 크롤링이라 한다.

웹 크롤링(Web Crawling)은 인터넷 상의 웹 페이지를 자동으로 탐색하고 필요한 데이터를 수집하는 프로세스를 의미합니다. 웹 크롤러(Web Crawler) 또는 스파이더(Spider)라고 불리는 소프트웨어 프로그램이나 스크립트를 사용하여 수행됩니다.

웹 크롤링의 주요 과정은 다음과 같습니다:

1. **시드 URL 설정:** 크롤링을 시작할 웹 페이지의 URL을 설정합니다. 이 URL을 시드 URL(seed URL)이라고 합니다.
2. **페이지 다운로드:** 설정된 URL을 기반으로 웹 페이지의 HTML 코드를 다운로드합니다.
3. **데이터 추출:** 다운로드한 HTML 코드에서 필요한 데이터를 추출합니다. 예를 들어, 특정 태그에 포함된 텍스트, 링크, 이미지 등을 추출할 수 있습니다.
4. **링크 추출 및 추적:** 웹 페이지 내의 다른 링크를 추출하고, 이를 통해 새로운 페이지로 이동하여 동일한 과정을 반복합니다. 이를 통해 여러 페이지를 순차적으로 탐색할 수 있습니다.
5. **데이터 저장:** 추출한 데이터를 데이터베이스나 파일로 저장하여 이후 분석하거나 사용할 수 있도록 합니다.

웹 크롤링은 다양한 도구와 라이브러리를 사용하여 수행할 수 있으며, 대표적으로 파이썬의 BeautifulSoup, Scrapy, Selenium 등이 있습니다. 웹 크롤링을 수행할 때는 해당 웹사이트의 로봇 배제 표준(robots.txt)을 준수하고, 과도한 트래픽을 유발하지 않도록 주의해야 합니다.

#### ▼ npm

npm은 자바스크립트 패키지 매니저다.

Node.js에서 사용할 수 있는 모듈들을 패키지화해서 모아둔 저장소 역할과 패키지 설치 및 관리를 위한 CLI(Command line interface)를 제공한다.