

간략하게 보는 자바스크립트의 변천사

송헌용

1995년 - 표준화되기 이전

- 화면 인터랙션을 위해 탄생
- 넷스케이프 - 라이브스크립트
- 익스플로러 - 제이스크립트
- 서로 다른 회사에서 개발하다보니 파편화

1996년 - 표준화 시도

- 넷스케이프에서 표준화 기구인 Ecma International에 요청
- 1996년 11월, ECMA-262라 불리는 명세서에 대한 작업을 시작했고 그 버전은 1997년 7월에 완성
- 당시 Sun사가 'JAVA'라는 단어를 상표 등록을 해놨기에 'JavaScript'라고 부를 수 없었음
- 표준 언어의 이름은 ECMAScript이고, 대외적으로 알리기 위해서 자바스크립트
- 현재 ECMAScript의 최신 버전은 7이고, 이것은 ECMA-262, edition 7과 같은 의미

1997년 - 동적인 HTML

- HTML 문서를 DOM으로 표현
- 자바스크립트로 DOM에 접근 가능
- 동적인 웹 콘텐츠 개발 가능
- 스타일이 변하고, 요소를 숨기거나 보이게 가능
- Internet Explorer4에서 등장

1999년 - XMLHttpRequest

- HTTP 혹은 HTTPS를 이용해 서버와 통신
- 페이지 새로고침 없이 데이터 갱신
- 텍스트 형식으로 데이터를 전송 받음
- Internet Explorer5에서 등장

1999년 - JSON

- JavaScript Object Notation
- 비동기 서버 통신을 위해 사용
- XML을 대체하는 주요 데이터 포맷
- key - value의 쌍으로 이루어진 구조

2005년 - Ajax의 대중화

- 1999년에 탄생했지만, 관심이 많지 않았음
- 구글에서 Ajax를 활용해서 지도, 메일을 개발
- 기존에는 대부분의 로직을 백엔드에서 담당했지만, 프론트엔드에서도 로직을 일부 담당하게 됨
- 싱글 페이지 어플리케이션 개발의 인기

2006년 - jQuery

- Javascript + Query(원래는 jSelect를 하려고 했음)
- DOM 트리에서 원하는 것을 질의한다는 의미로 Query
- 이 당시, 웹 브라우저마다 서로 다른 API를 제공
- 순수 자바스크립트로 개발하면 크로스브라우징 이슈
 - DOM 탐색, 선택, 수정
 - 인터랙션
 - 유틸리티 메소드

2008년 - V8 엔진

- 구글에서 개발된 오픈소스, 자바스크립트 엔진
- JIT = 인터프리터의 장점 + 컴파일러의 장점
 - Just-In-Time compilation
- 크롬 브라우저와 안드로이드 브라우저에 탑재
- ECMA Script 3rd Edition 규격의 C++로 개발
- 독립적으로 실행이 가능

2009년 - Node.js

- 단일 스레드 + 이벤트 루프를 통한 높은 처리 성능
- 내장 HTTP 서버 라이브러리를 포함
- 확장성 있는 네트워크 어플리케이션 개발에 사용되는 소프트웨어 플랫폼
- 패키지 매니저인 NPM은 2011년에 등장

Node.js의 주요 활용 예

- NPM: 외부 의존성 관리
- Babel: 코드 변환(transpiler)
- Browserify / Webpack: 모듈 번들러
- Grunt / Gulp / NPM script: 빌드 자동화
- Express + etc: API 웹 서버 구축

2015년 - ES2015

- let / const에 의한 블록 스코프
- Map / Set / WeakMap / WeakSet에 의한 컬렉션
- 형을 정의하는 class
- generator / for ... of
- CommonJS 방식의 모듈
- context 자동 제어를 위한 화살표 함수
- 비동기 프로그래밍을 위한 Promise

2016년 - ES2016

- ES2015의 마이너 패치
- `Array.prototype.includes`
- 거듭제곱 연산자 `**`

2017년 - ES2017

- Object.values
- Object.entries
- Object.getOwnPropertyDescriptors
- String.prototype.padStart
- String.prototype.padEnd
- async / await

감사합니다.