

01

01-3. JavaScript Overview

Overview

자바스크립트는 인기 있는 언어일까? (TIOBE, 를 참고해보자)

자바스크립트 역사

자바스크립트는 역사가 오래된 소프트웨어 언어이다.

- 자바스크립트는 1995년 Brendan Eich(브렌던 아이크 – 자바스크립트 창시자)에 의해 처음 개발되었으며 초기에는 LiveScript라고 불리었습니다.
- 1996년 네스케이프 커뮤니케이션즈에서 LiveScript를 자바스크립트(Javascript)로 이름을 변경하면서 네스케이프 브라우저의 표준 스크립트 언어로 채택을하기 시작하면서 웹 개발자들에게 알려지기 시작했고, 이용되기 시작했습니다.

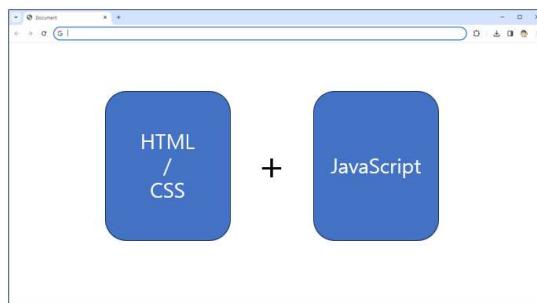
web 초창기에 전세계 100% 브라우저는 네스케이프였다.

자바스크립트 역사

자바스크립트는 대단한 애플리케이션을 만들 목적이 아니었다.

- 자바스크립트의 탄생 목적은 그렇게 대단한 애플리케이션을 만들기 위한 목적이 아니었습니다.
- 브라우저에서 간단한 언어적인 처리만을 목적으로 해서 자바스크립트가 이용된 것입니다.

연산은 백엔드가 다 처리했었다.



- HTML 문서의 화면 구성은 HTML/CSS 조합으로 하고 그곳에 자바스크립트를 추가해서 언어적인 기능이 가능하게 하겠다는 의도이며 이런 목적으로 넷스케이프 브라우저에 자바스크립트가 추가된 것입니다.
- 자바스크립트는 다른 소프트웨어 언어와 비교해 보면 문법이 약간 유연합니다.

자바스크립트 역사

원래 이런 단어는 존재하지 않았다. 백엔드에서 한 것을 render 정도만 했다.

프런트 웹 애플리케이션이 인기를 끌면서 덩달아서 인기를 얻게 된 언어이다.

- 자바스크립트가 현재처럼 이렇게 인기를 끌게 된 것은 프런트 웹 애플리케이션의 발전 때문입니다.
- 과거와 다르게 브라우저에서 실행되는 웹이 애플리케이션급으로 개발되기를 원하다 보니 그 애플리케이션을 개발할 수 있는 소프트웨어 언어가 있어야 하고 그러다 보니 자바스크립트로 작성하는 코드의 규모 및 개발하고자 하는 내용이 다양화 되었으며 이로인해 자바스크립트의 인기가 올라가게 된 것입니다.

자바스크립트와 자바의 관계는?

: 전혀 관계가 없다. 인기있는 Java의 이름을 javaScript가 따라한 것이다.

자바스크립트 역사

Node.js 때문에 더욱 인기있는 언어가 되었다.

- Node.js 가 나오기 시작하면서 더 이상 자바스크립트가 브라우저만을 위한 소프트웨어가 아닌 브라우저 밖에서 실행되는 애플리케이션을 개발하기 위한 언어로 사용이 가능해 졌습니다.
- 그로인해 백엔드 웹 애플리케이션이 자바스크립트로 개발되기 시작했고 이로인해 자바스크립트의 이용비율이 훨씬 더 증가하게 되었습니다.

프론트에서만 해도 인기가 올라가고 있었는데, 백엔드까지 사용하니 인기가 더 올라갔다.

자바스크립트 역사

자바스크립트는 ECMA 단체에서 표준을 정의하고 있다.

- 자바스크립트의 첫 시작은 넷스케이프 브라우저에서 이용하기 위한 소프트웨어 언어였습니다만 넷스케이프 브라우저 이외 다양한 브라우저에서 채택이 되어 사용되었습니다.
- 그런데 브라우저별로 자바스크립트를 독자적으로 발전을 시키다 보니 자바스크립트 호환성 문제가 발생했습니다.
- 자바스크립트의 표준을 책정하여 모든 브라우저에 동일하게 동작할 수 있게 하는 필요성이 제기되었고 이 요구를 받아 ECMA 인터내셔널(<https://ecma-international.org/>)이라는 비영리 단체에서 자바스크립트 표준을 책정하게 되었습니다.

자바스크립트 역사

ECMA2015

- 자바스크립트에 대한 표준은 ECMA 단체에서 정의합니다.
- 자바스크립트를 또 다른 용어로 ECMAScript 라고도 합니다.
- ECMAScript 의 표준이 1 버전부터 5버전까지 나왔는데 이 5버전이 나온 시점이 90년대 후반입니다.
- 표준은 계속 버전업 하면서 변화가 있기는 했지만 90년대 후반의 웹개발 환경과 그 환경에서의 자바스크립트 언어를 목적으로 하다보니 표준에서 제공되는 여러가지 기법들이 다른 소프트웨어 언어에 비해 부족했습니다.
- 즉 90년대 후반의 ECMA 5 버전은 애플리케이션을 만들 목적이 아니라 간단한 소프트웨어적인 코드 작성에만 초점을 맞춘 버전이라는 이야기입니다. 그리고 한동안 버전 변경이 없었습니다.

애플리케이션을 만들기에는 굉장히 미약

자바스크립트 역사

ECMA2015

- 2010년대로 넘어오면서 프런트 웹 애플리케이션이 개발되기 시작하다 보니 자바스크립트 표준에서 제공되는 것들이 너무 빈약해 보이기 시작한 것입니다
- 각 브라우저 별로 애플리케이션을 개발하기 위한 독자적인 방법을 추가하기 시작했습니다.
- 문제는 각 브라우저의 방식이 다르다 보니 하나의 애플리케이션을 만드는 개발자 입장에서 우리의 애플리케이션이 어느 브라우저에서 실행되는지에 따라 다른 코드를 작성해야 하는 상황이 발생했습니다.
- 이런 문제를 해결하기 위해서 ECMA 단체에서 오랜만에(ECMA5 표준이 나온 후 오랜만에) 새로운 표준을 발표합니다.
- 이 표준이 ECMA 6 버전이며 줄여서 ES6 라고 부릅니다.
- 이 ES6를 발표하면서 자바스크립트도 정식의 소프트웨어 언어처럼 많은 애플리케이션을 개발하기 위한 기법을 제공하겠다고 선언한 것입니다.

자바스크립트 역사



ECMA 6 (ES6) : 웹 애플리케이션을 만들 수 있는 표준을 지원하기 위해 발표한 것

- 계속 버전을 변경하면서 ECMA 7, 8 버전이 발표되었는데 이름을 새로운 버전이 발표된 년도를 추가해서 ECMA2015, ECMA2016, ECMA2017, ECMA2018, ECMA2019 이런식으로 부르기 시작했습니다.
- 결국 ECMA6 와 ECMA2015는 동일한 버전을 이야기하며 줄여서 ES6 혹은 ES2015 라고 부릅니다.
- ES2015가 나오면서 거의 매년 새로운 버전을 발표하고 있는데 개발자들이 흔히 각각의 세부 버전을 명시하지 않고 ES2015 이후에 나온 버전을 통칭해서 ES2015 라고 부르고 있습니다.

ES6가 나온지 오래됐기 때문에 요즘은 버전별로 구분을 딱히 하지는 않는다.

자바스크립트 특징 - 스크립트 언어

컴파일 언어와 스크립트 언어

- 소프트웨어 언어는 크게 또 두 가지 타입으로 구분
- 소스코드는 사람이 이해할 수 있는 문서이기 때문에 엄밀히 말하면 텍스트 파일

스크립트 언어

- 사람이 해석할 수 있는 소스 파일을 특정 플랫폼에서 직접 해석해 실행시켜 준다면 그 소스 파일을 개발한 소프트웨어 언어를 스크립트 언어라고 분류합니다.
- 자바스크립트가 대표적인 스크립트 언어입니다. 자바스크립트로 개발된 소스파일이 브라우저에 의해 해석되어 애플리케이션이 실행되게 됩니다.



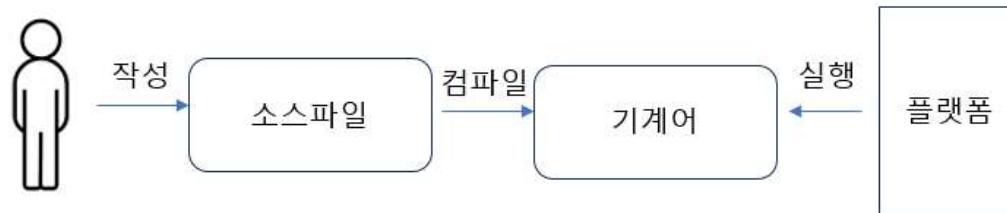
컴파일을 안 해도, 플랫폼이 우리가 작성한 걸 바로 해석 가능

자바스크립트 특징 - 스크립트 언어

컴파일 언어

우리가 작성한 코드를 플랫폼이 직접 돌려주진 못하기에

- 소스 파일을 다시 플랫폼에서 해석 가능한 형태로 변형시켜 실행
- 소스파일을 플랫폼이 이해할 수 있는 형태로 변형하는 작업을 컴파일이라고 부르며 컴파일에 의해 사람은 해석 불가능한 특정 플랫폼만 이해할 수 있는 기계어로 바뀌었다고 표현합니다.
- 이 컴파일 과정이 필요한 소프트웨어 언어를 컴파일 언어라고 부릅니다.



- 대표적으로 자바, C 등의 소프트웨어 언어들이 컴파일 언어

1. html parwer
2. css engine
3. js engine

자바스크립트 특징 - 브라우저의 자바스크립트 엔진에 의해 실행

- 자바스크립트를 실행시켜 주는 플랫폼 역할을 하는 것은 크게 2가지 입니다. 하나는 브라우저에 내장되어 있는 자바스크립트 엔진이고 또 다른 하나는 노드(Node)입니다
- 프런트 웹 애플리케이션의 자바스크립트는 브라우저의 자바스크립트 엔진에 의해 실행된다고 정리할 수 있습니다.
- 백엔드 웹 애플리케이션을 자바스크립트로 만든다면 그 애플리케이션은 브라우저에서 실행되는 것이 아님으로 자바스크립트를 실행시켜 줄 플랫폼이 따로 있어야 합니다.
- 이 역할을 하는 대표적인 것이 노드입니다. 노드는 '자바스크립트 런타임'이라고 소개되며 자바스크립트를 실행시켜 주는 역할을 합니다.

타입스크립트

자바스크립트 단점

10-100줄 정도되는 간단한 코드를 작성할 땐 문법이 전혀 문제가 안 됐었음.

- 자바스크립트의 탄생 배경이 브라우저에서 실행되는 간단한 코드를 작성하기 위한 것임으로 다른 언어와 비교해 보면 문법이 유연하고 간단합니다.
- 문법이 유연하고 간단하다는 것은 장점일 수도 있고 단점일 수도 있는데 간단한 코드를 작성하기에는 좋습니다만 복잡한 애플리케이션을 개발하기에는 여러가지 어려움을 겪을 수밖에 없습니다.
- 복잡한 애플리케이션을 개발하기 시작하면서 많은 개발자들에 의해 조금 더 정교한 언어가 필요하다는 요구가 생기게 되었고 그런 이유로 자바스크립트를 대체할 목적의 많은 언어들이 만들어 졌습니다.
- CoffeeScript 도 있었고, Kotlin, Swift , Dart 등으로도 자바스크립트를 대체할 수 있습니다.
- 그런데 이런 다양한 시도 중 유일하게 협업에서 자바스크립트를 대체하기 위해서 사용하는 소프트웨어 언어가 타입스크립트(TypeScript)입니다.

타입스크립트

타입스크립트

- 타입스크립트(Typescript) 는 MS에서 만든 언어예요.
- 타입스크립트는 확장자가 ts로 만들어 집니다.
- 그런데 타입스크립트로 만든 소스는 브라우저의 자바스크립트 엔진이나 노드에 의해 실행되지 못합니다.
- 결국 타입스크립트로 작성한 소스가 실행되기 위해서는 자바스크립트로 변형작업을 해주어야 합니다. 물론 이 변형작업을 자동으로 해주는 툴이 제공이 됩니다.

변형작업 : transpile (complie)

Q. 그럼 자바스크립트가 아니라 타입스크립트를 학습해야 하나?

A. 타입스크립트는 완벽하게 자바스크립트를 대체할 수 있다. 하지만 여전히 자바스크립트의 점유율이 높다.

자바스크립트를 이해하면, 타입스크립트를 정리하는 것은 시간이 오래걸리지 않는다. 자바스크립트를 먼저 제대로 학습해야 한다.



감사합니다

단단히 마음먹고 떠난 사람은
산꼭대기에 도착할 수 있다.
산은 올라가는 사람에게만 정복된다.



윌리엄 셰익스피어

William Shakespeare