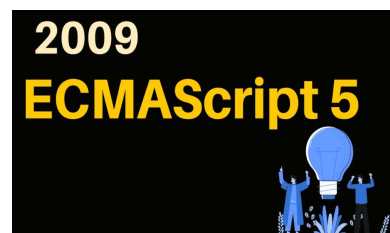
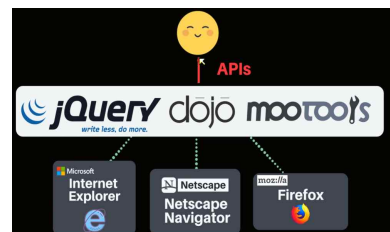
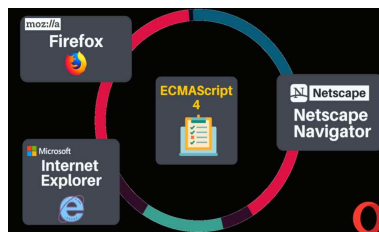
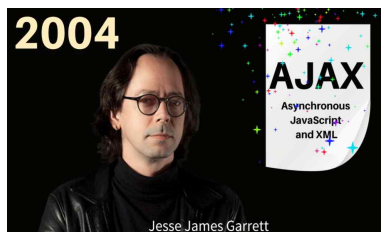
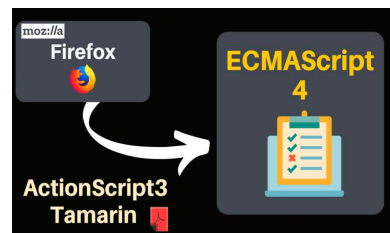
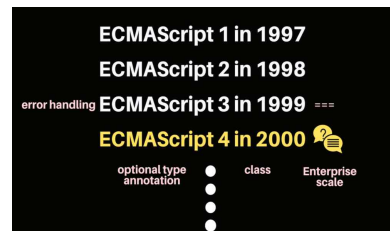
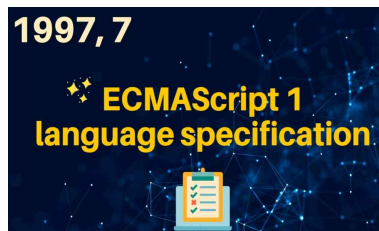
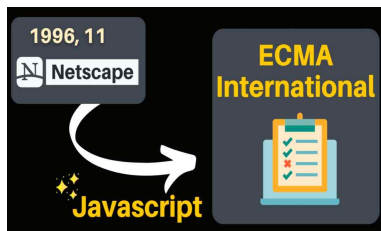
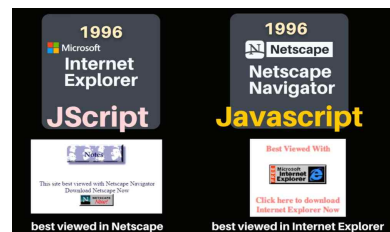
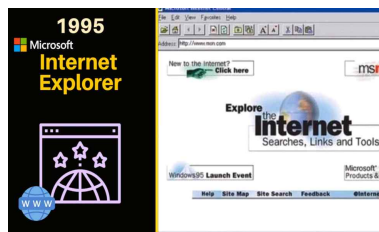
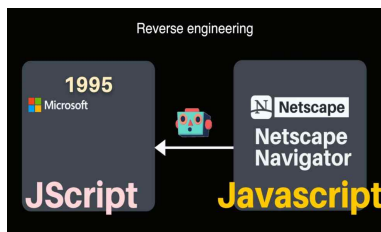


JavaScript의 역사


- ✓ 1993 NCSA Mosaic Web Browser, Marc Andreessen → Netscape
- ✓ 1994 Netscape Navigator → HTML/CSS, 80%
Brendan Eich → Scheme Scripting Language
- ✓ 1994.9 Mocha Language 탄생 → LiveScript → LiveScript Interpreter
LiveScript → JavaScript 로 이름 변경
- ✓ 1995 Netscape Navigator → JavaScript, Interpreter Engine 포함
- ✓ 1995 NetScape Navigator → Reverse engineering으로 JScript 출시
→ Internet Explorer 발표
- ✓ 1996.11 ECMA International ← JavaScript 표준 제안
- ✓ 1997.7 ECMAScript 1 language specification
- ✓ 1998 ECMAScript 2
- ✓ 1999 ECMAScript 3 → error handling, === 추가
- ✓ 2000 ECMAScript 4 → optional type annotation, class, ...
- ✓ 2000 Internet Explorer 시장 점유율 → 95%
- ✓ 2004 moz://a → Firefox 출시
ActionScript3 / Tamarin 엔진 표준안 제안
- ✓ 2004 Jesse James Garrett → AJAX(Asynchronous JavaScript and XML) 제안
IE, NetScape Navigator, Firefox 다양한 Web Browser 표준화 전쟁
jQuery, dojo, mootools APIs 제안
- ✓ 2008 Google Chrome 출시 → JIT(Just-In-Time Compilation)
- ✓ 2008.7 표준안 협의
- ✓ 2009 ECMAScript 5
- ✓ 2015 ECMAScript 6 → default parameter, class, arrow function, const, let
JavaScript → Mature, Settled down,
jQuery, dojo, mootools APIs 지원없이도 JavaScript 만으로도 가능
- ✓ 2016 ECMAScript 7
- ✓ 2017 ECMAScript 8
- ✓ 2018 ECMAScript 9
- ✓ 2019 ECMAScript 10

그림으로 보는 JavaScript의 역사



2015 ECMAScript 6

default parameter class
arrow function
const let



ECMAScript 5 in 2009
ECMAScript 6 in 2015
ECMAScript 7 in 2016
ECMAScript 8 in 2017
ECMAScript 9 in 2018
ECMAScript 10 in 2019

Javascript Engines

V8 Chrome	SpiderMonkey Firefox	JSCore Safari
Chakra MS Edge	Carakan Opera	Tamarin Adobe Flash

Javascript Engines

V8
Chrome

node

ELECTRON

BABEL

JavaScript transpiler

TS

SPA

Single Page Application

“Javascript”

React

ANGULAR

JS

“Javascript”

node
back-end

mobile

desktop

Javascript

JS

WebAPIs



☑ 왜 ES6 인가

ECMAScript는 Ecma international의 ECMA-262 기술 규격에 정의된 표준화된 Script Programming Language 이다. JavaScript를 표준화하기 위해서 만들어졌으며 꼭 JavaScript가 아니더라도 Adobe Flash를 사용하는 소프트웨어에서 사용하는 ActionScript, MS사의 IE(Internet Explorer)에 쓰이는 JScript 등 다른 구현체들 역시 포함하고 있다.

왜 ES6 에 대해 큰 관심을 가지는 것일까? ES6가 최신 버전이기 때문에 그럴까?

아니다. 현재 2019년 12월 기준으로 ES10까지 나왔다. 1997년 6월에 1판을 시작으로 불규칙적이게 개정되던 것이 2015년부터 매년 6월에 규칙적으로 개정되고 있다.

판	출판일	이름	이전 판과의 차이점
1	1997/6	ECMAScript 1	초판
2	1998/6	ECMAScript 2	ISO/IEC 16262 국제 표준과 완전히 동일한 규격을 적용하기 위한 변경.
3	1999/12	ECMAScript 3	강력한 정규 표현식, 향상된 문자열 처리, 새로운 제어문 , try/catch 예외 처리, 엄격한 오류 정의, 수치형 출력의 포매팅 등.
4	2000 버려짐	ECMAScript 4	4번째 판은 언어에 얽힌 정치적 차이로 인해 버려졌다. 이 판을 작업 가운데 일부는 5번째 판을 이루는 기본이 되고 다른 일부는 ECMAScript의 기본을 이루고 있다.
5	2009/12	ECMAScript 5	더 철저한 오류 검사를 제공하고 오류 경향이 있는 구조를 피하는 하부집합인 "strict mode"를 추가한다. 3번째 판의 규격에 있던 수많은 애매한 부분을 명확히 한다.
5.1	2011/6		ECMAScript 표준의 제 5.1판은 ISO/IEC 16262:2011 국제 표준 제3판과 함께 한다.
6	2015/6	ECMAScript 6 ES2015 (ES6)	6판에는 class와 module 같은 복잡한 응용 프로그램을 작성하기 위한 새로운 문법이 추가되었다. 하지만 이러한 문법의 의미는 5판의 strict mode와 같은 방법으로 정의된다. 이 판은 "ECMAScript Harmony" 혹은 "ES6 Harmony" 등으로 불리기도 한다.
7	2016/6	ES2016	제곱연산자 추가, Array.prototype.includes
8	2017/6	ES2017	함수 표현식의 인자에서 trailing commas 허용, Object values/entries method, async/await 등.
9	2018/6	ES2018	Promise.finally, Async iteration, object rest/spread property 등.
10	2019/6	ES2019	Object.fromEntries, flat, flatMap, Symbol.description, optional catch 등.

☑ JavaScript Engines

- ✓ V8 -> Chrome, MS Edge 채택
- ✓ SpiderMonkey -> Firefox
- ✓ JSCore -> Safari
- ✓ Chakra -> MS Edge, Opera, Adobe Flash

- ✓ V8 -> Nnodejs, ELECTRON 에서도 이용

- ✓ Babel -> JavaScript transcompiler

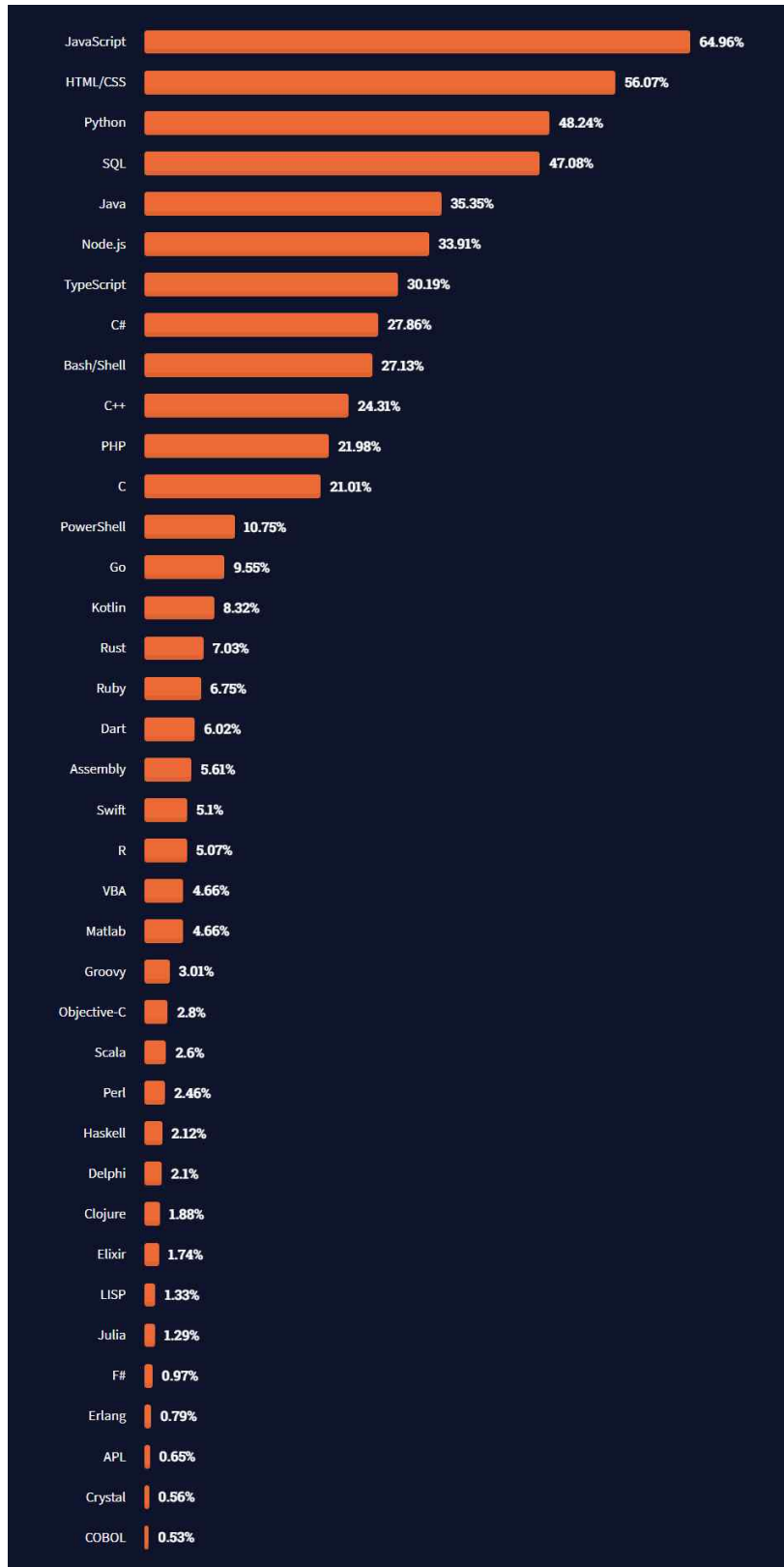
- ✓ SPA(Single Page Application) -> React, Vue, Angular

- ✓ JavaScript => Back-End -> Nodejs 등장
 - mobile -> React Naticce, Cordova
 - desktop app -> ELECTRON

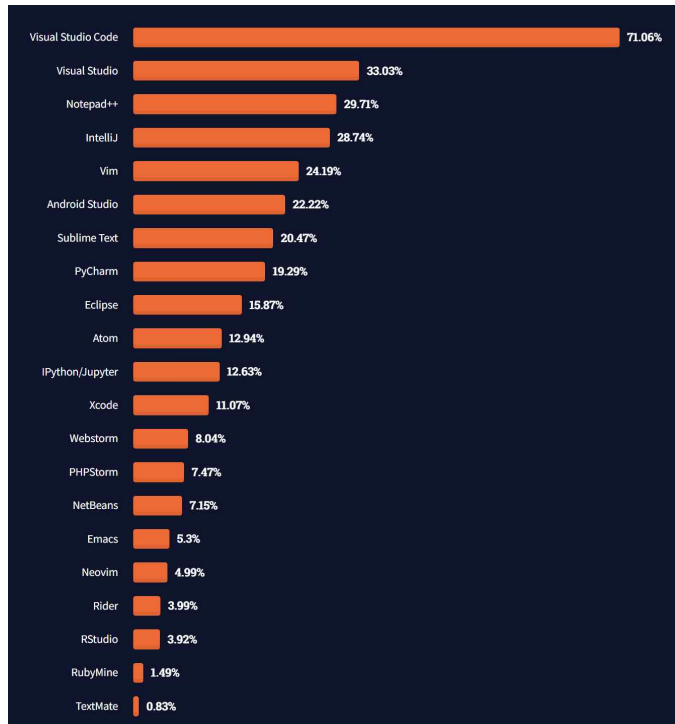
The 2021 Stack Overflow Developer Survey

<https://insights.stackoverflow.com/survey/2021>

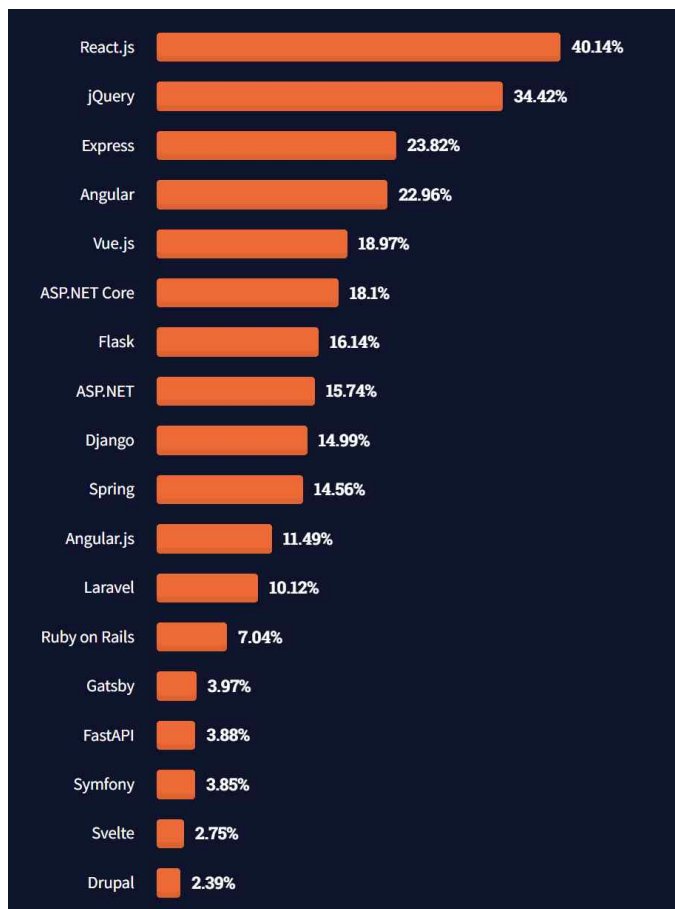
❖ the most commonly used programming language



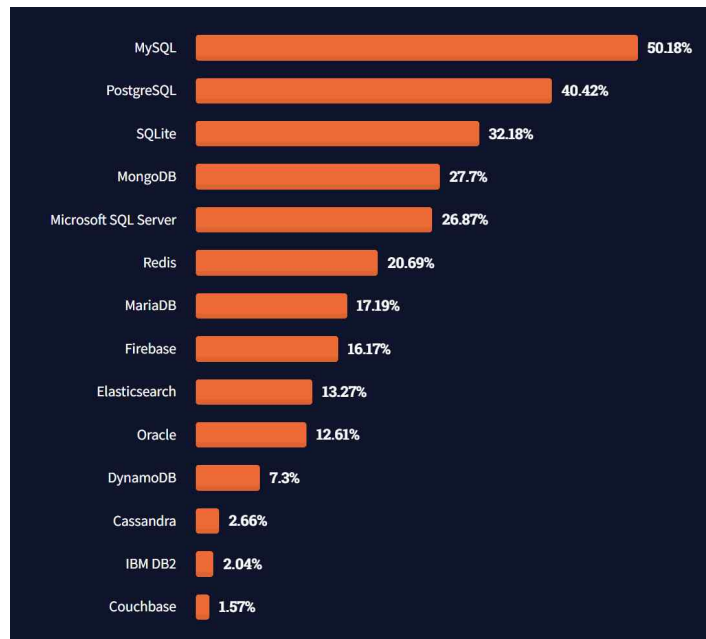
❖ IDE(Integrated Development Environment)



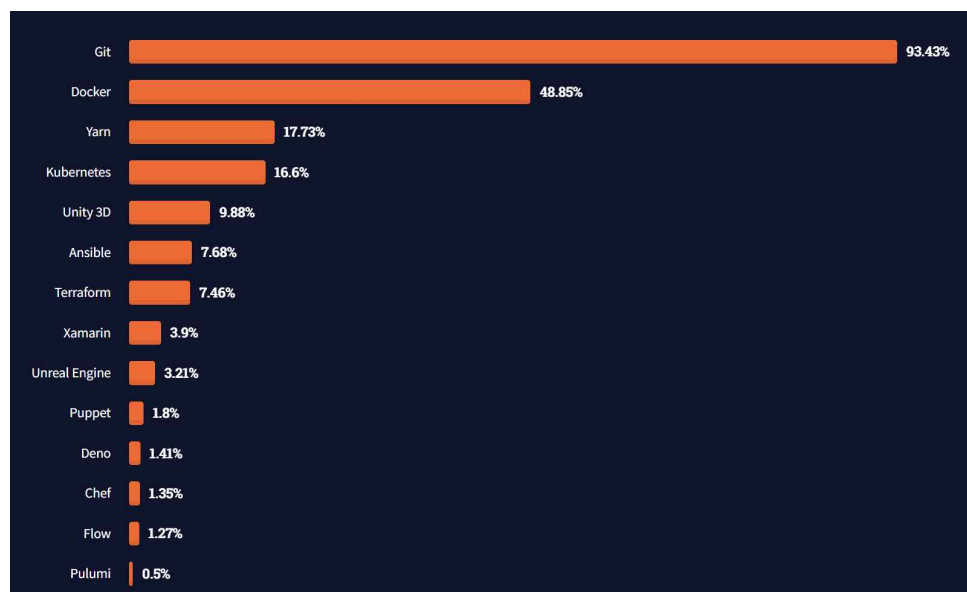
❖ Web Frameworks



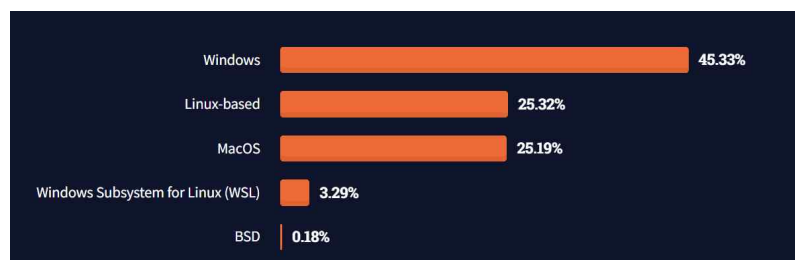
❖ DataBases



❖ Other tools



❖ Operating System



<https://2021.stateofjs.com/en-US/>