

# 25 Years of JavaScript

•••

Why History Matters and What We Can Learn From It

[laurasprauer.com/momentum-2021](https://laurasprauer.com/momentum-2021)



# Laura Sprauer

[laurasprauer.com](http://laurasprauer.com)

[linkedin.com/in/laurasprauer](https://linkedin.com/in/laurasprauer)

[ample.co](http://ample.co)

[laurasprauer.com/  
momentum-2021](http://laurasprauer.com/momentum-2021)

# 25 Years of JavaScript

The Web is For Everyone

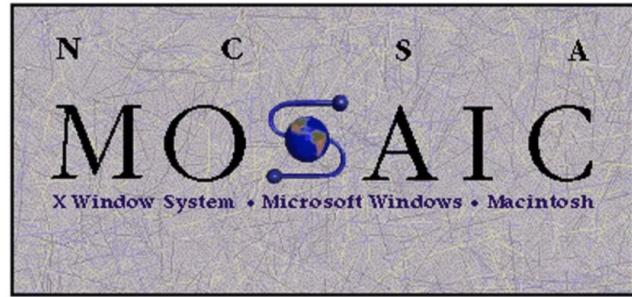
The Birth of JavaScript

Initial Response

Renaissance & Rebirth

ES6 & Modern Javascript

The Future & What We Can Learn From the Past

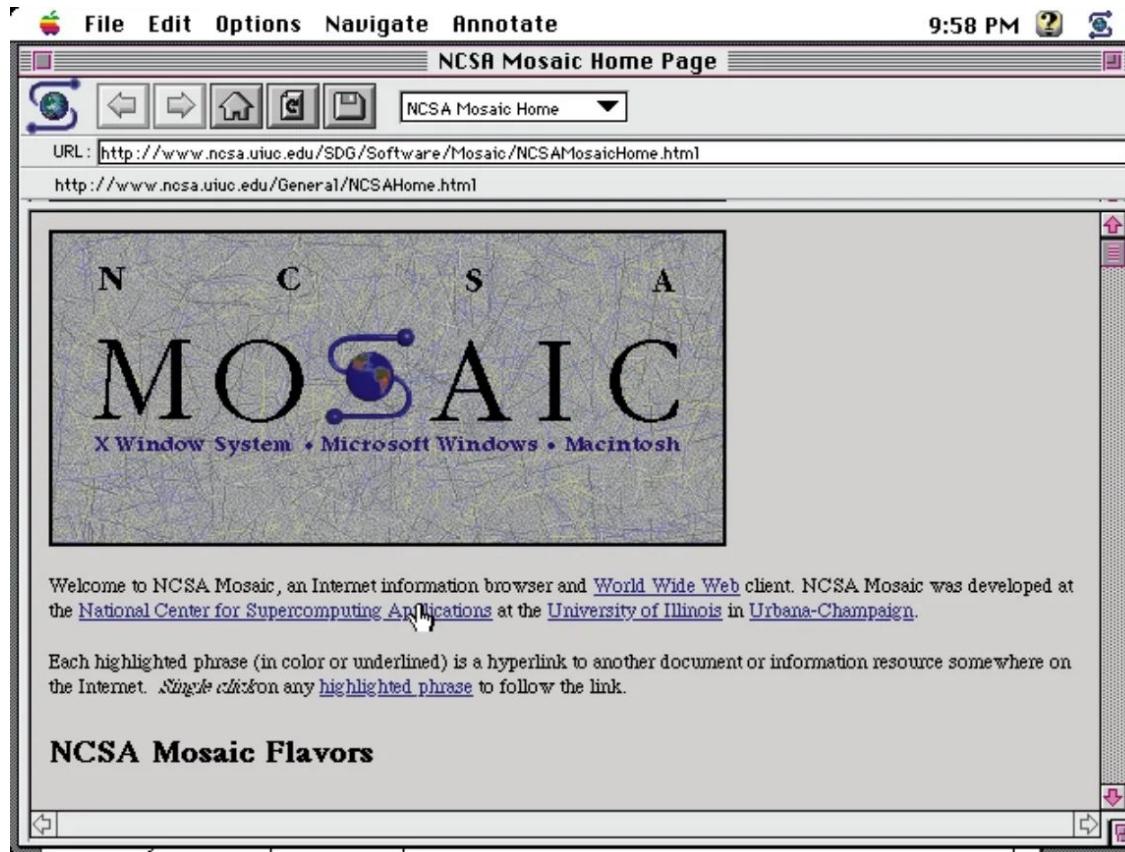


Netscape

# “The Web is for Everyone”

NCSA - MOSAIC (1993)

Mosaic Communications Corporation - Netscape (1994)



NCSA Mosaic for Macintosh (1993)



Mosaic Netscape 0.9 Beta for Windows (1994)



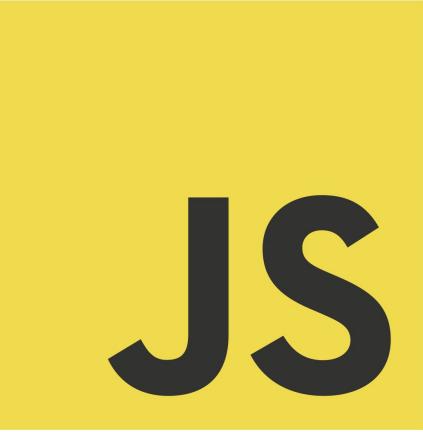
Source: [wikipedia.org/netscape\\_navigator](https://en.wikipedia.org/wiki/Netscape_Navigator), [wikipedia.org/netscape](https://en.wikipedia.org/wiki/Mosaic_(web_browser))



Netscape Navigator 1.2 Browser for Windows 3.1 (1995)



## Netscape Navigator (1994 - 1998)



**JS**

```
console.log("hello world");
```

# JavaScript is Born

Collaboration with Sun Microsystems (Java)

Brendan Eich, Father of JavaScript





# Brendan Eich

“ I'm to blame for JavaScript ... It was supposed to be a sidekick language to Sun's Java language, but times have changed... and Javascript had enough good in it that the sidekick became the superhero. ”



Object oriented language based  
on the concept of prototypes



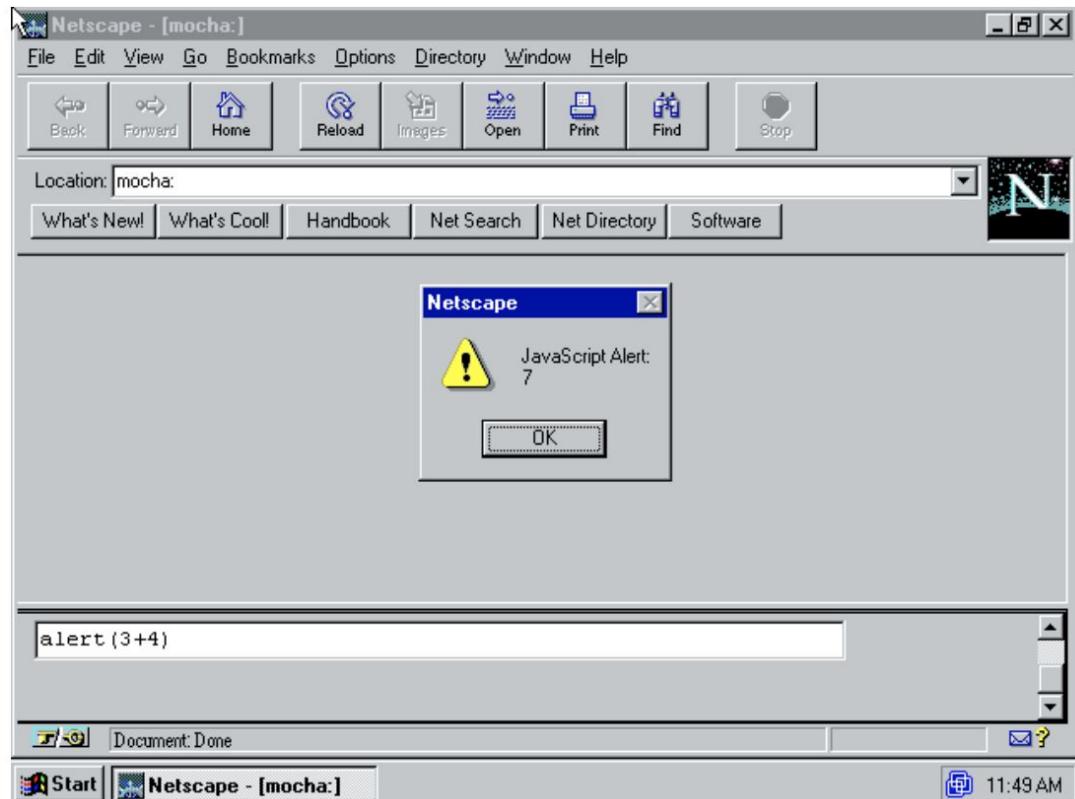
**Scheme**  
First Dialect of Lisp to use  
lexical scope and support  
functional programming

# 1995

**May**  
Eich invents Mocha

**September**  
Mocha renamed LiveScript

**December**  
LiveScript renamed JavaScript



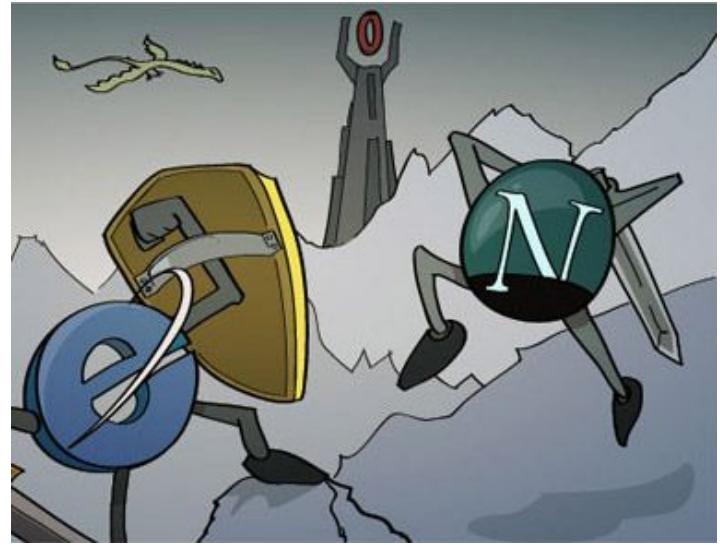
Netscape 2.02 on Windows 95

# JavaScript 1.0 Missing Features

A distinct Array object type  
Regular expressions  
A global binding for undefined  
typeof, void, delete operators  
do-while statement  
try-catch-finally statement  
Nested function declarations  
Function call and apply methods  
Prototype-based inheritance  
Cyclic garbage collection<sup>g</sup>

Array literals  
Object literals  
==== operator  
in, instanceof operators  
switch statement  
break/continue to label  
Function expressions  
prototype property of functions  
Access to built-in prototype objects  
HTML <script> tag src attribute

# ECMAScript



## Initial Response

Competition & Standardization

The First Browser War

Source: [Holy Browser Wars!](#)

# Competition & Standardization



ES1

ECMAScript 1  
(1997)

First edition

ES2

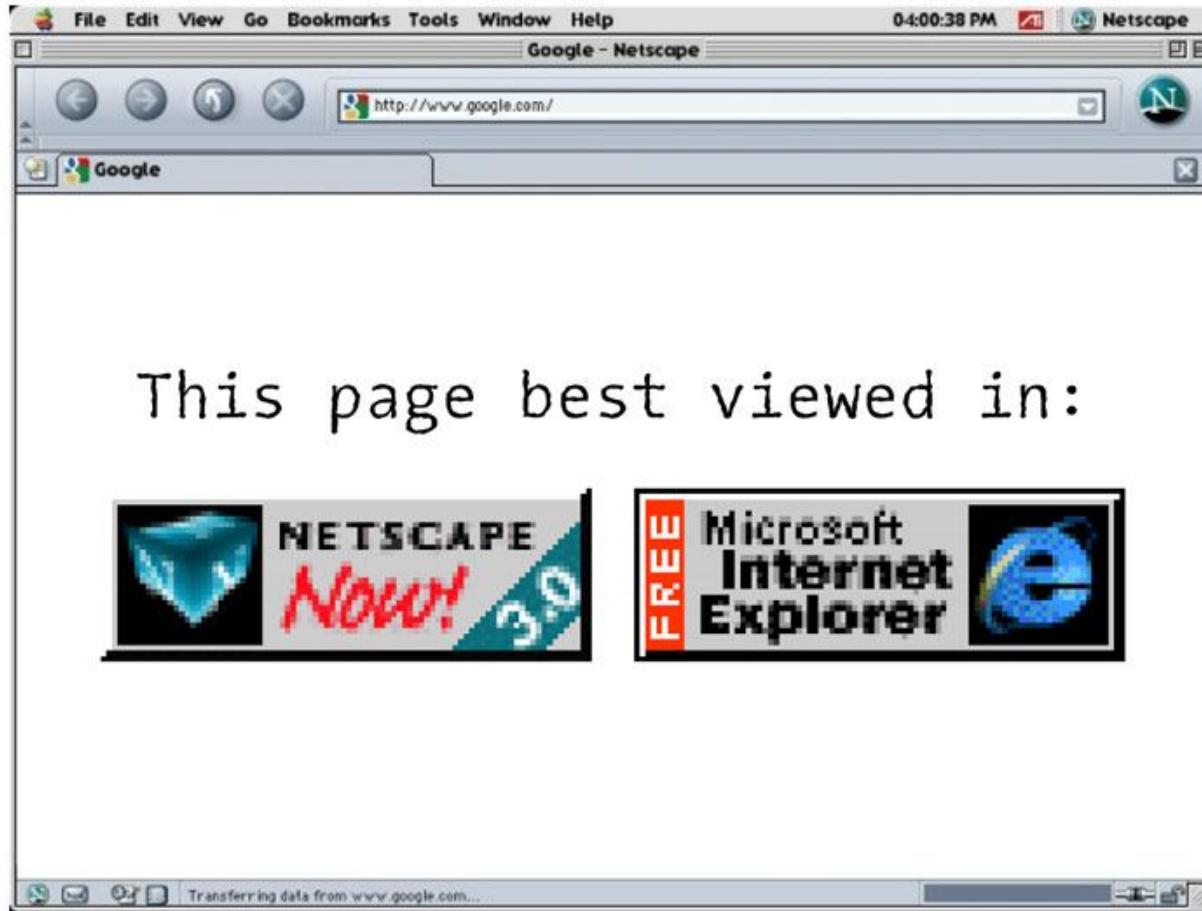
ECMAScript 2  
(1998)

Editorial changes

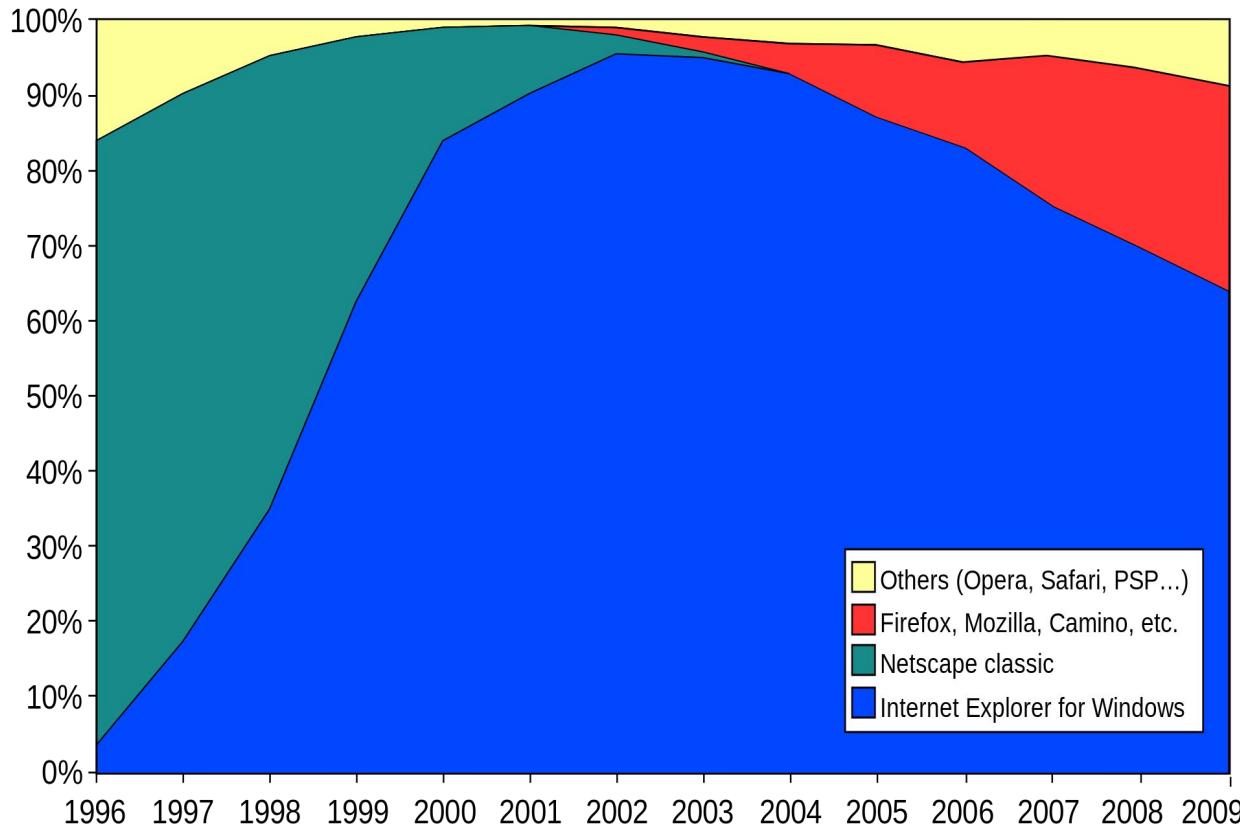
ES3

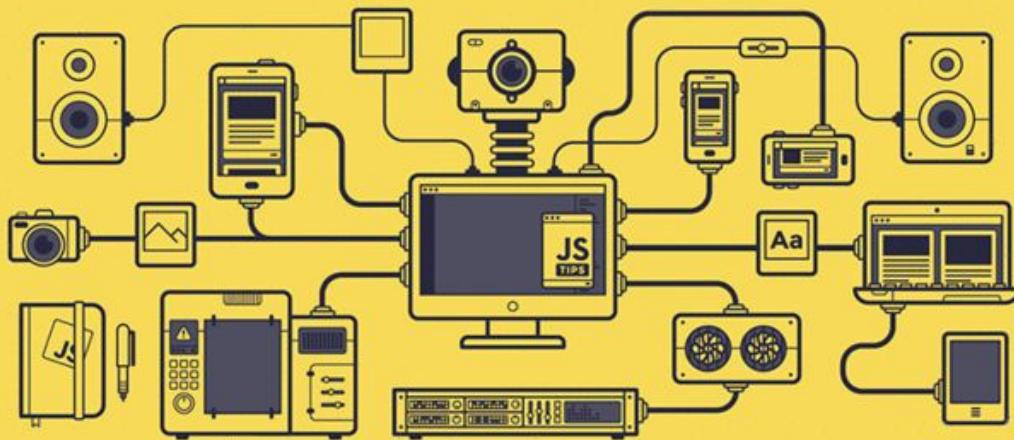
ECMAScript 3  
(1999)

Added regular ex-  
pressions &  
try/catch



# Browser Wars





# JavaScript Doesn't Need Java

Douglas Crockford (JSON, JSMIn, JSLint)

AJAX Programming Concept

Innovation & Conflict



# Douglas Crockford

“ I was the first person to recognize that JavaScript had good parts ... And when I announced my results they were met with wild skepticism. No way, there cannot possibly be any good parts in JavaScript. ”



```
<html><head><script>
    document.domain = 'fudco';
    parent.session.receive(
        { to: "session", do: "test",
          text: "Hello world" }
    )
</script></head></html>
```

# AJAX

An event occurred on a web page  
e.g. a button is clicked



Process the returned data using JavaScript  
and updates the page content

An HTTP request is sent to server  
using **XMLHttpRequest** object

Network

Process the HTTP request  
and creates a response



Send response back  
to the browser

# ES4?

1999



...

2004



Firefox

2005



Google maps

2006



2007



2008





# Renaissance & Rebirth

JavaScript Everywhere with CommonJS & Node.js

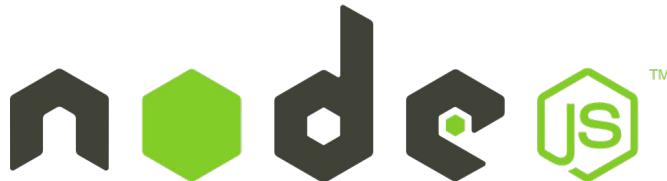
ES3.1 → ES5

Single Page Applications

# CommonJS

```
var $ = require('jquery')
```

```
module.exports = function add(a, b) {
  return a+b;
}
```



## 2009

- Node.js is born
- The first form of npm is created

## 2010

- Express is born
- Socket.io is born

## 2011

- npm hits version 1.0
- Larger companies start adopting Node.js: LinkedIn, Uber, etc.
- hapi is born

## 2012

- Adoption continues very rapidly

## 2013

- First big blogging platform using Node.js: Ghost
- Koa is born

# ES5

"use strict"	
String[number]	Array.indexOf()
Multiline strings	Array.lastIndexOf()
String.trim()	JSON.parse()
Array.isArray()	JSON.stringify()
Array.forEach()	Date.now()
Array.map()	Date.toISOString()
Array.filter()	Date.toJSON()
Array.reduce()	Property getters and setters
Array.reduceRight()	Reserved words as property names
Array.every()	Object methods
Array.some()	Object.defineProperty()
	Function.bind()
	Trailing commas

**Knockout.**

 BACKBONE.JS



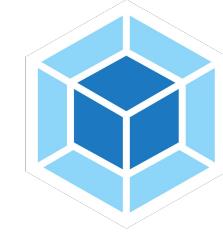
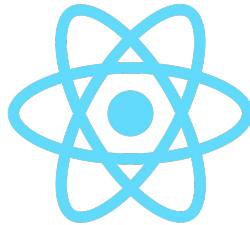
 **ANGULARJS**  
by Google

The logo for AngularJS consists of a large, stylized letter 'A' inside a red pentagon with a thin grey border. To the right of the icon, the word "ANGULARJS" is written in a large, bold, black sans-serif font. Below that, the words "by Google" are written in a smaller, regular black sans-serif font.



# Jeremy Ashkenas

“ The way you invent the future of JavaScript is to compile the JavaScript of the future into the JavaScript of today ”



# Harmony & Modern JavaScript

ES6 (2015) Harmony & Beyond

Transpilers - Babel / TypeScript

More New SPA Frameworks - React / Vue / Angular 2

Bundlers - Browserify / Webpack

# ES6

## Harmony

The <code>let</code> keyword	<code>String.startsWith()</code>
The <code>const</code> keyword	<code>String.endsWith()</code>
Arrow Functions	<code>Array.from()</code>
For/of	<code>Array.keys()</code>
Map Objects	<code>Array.find()</code>
Set Objects	<code>Array.findIndex()</code>
Classes	New Math Methods
Promises	New Number Properties
Symbol	New Number Methods
Default Parameters	New Global Methods
Function Rest Parameter	Iterables <code>Object.entries</code>
<code>String.includes()</code>	JavaScript Modules

2016

Exponentiation (\*\*), Exponentiation assignment (\*\*=)  
Array.prototype.includes

---

2017

String padding, Object.entries, Object.values, async functions, shared  
memory

---

2018

Asynchronous Iteration, Promise Finally, Object Rest Properties, New RegExp  
Features

---

2019

Array.prototype.flat, Array.prototype.flatMap, changes to  
Array.sort and Object.fromEntries

---

2020

Optional Chaining, BigInt, Nullish Coalescing, Promise.allSettled(),  
MatchAll()

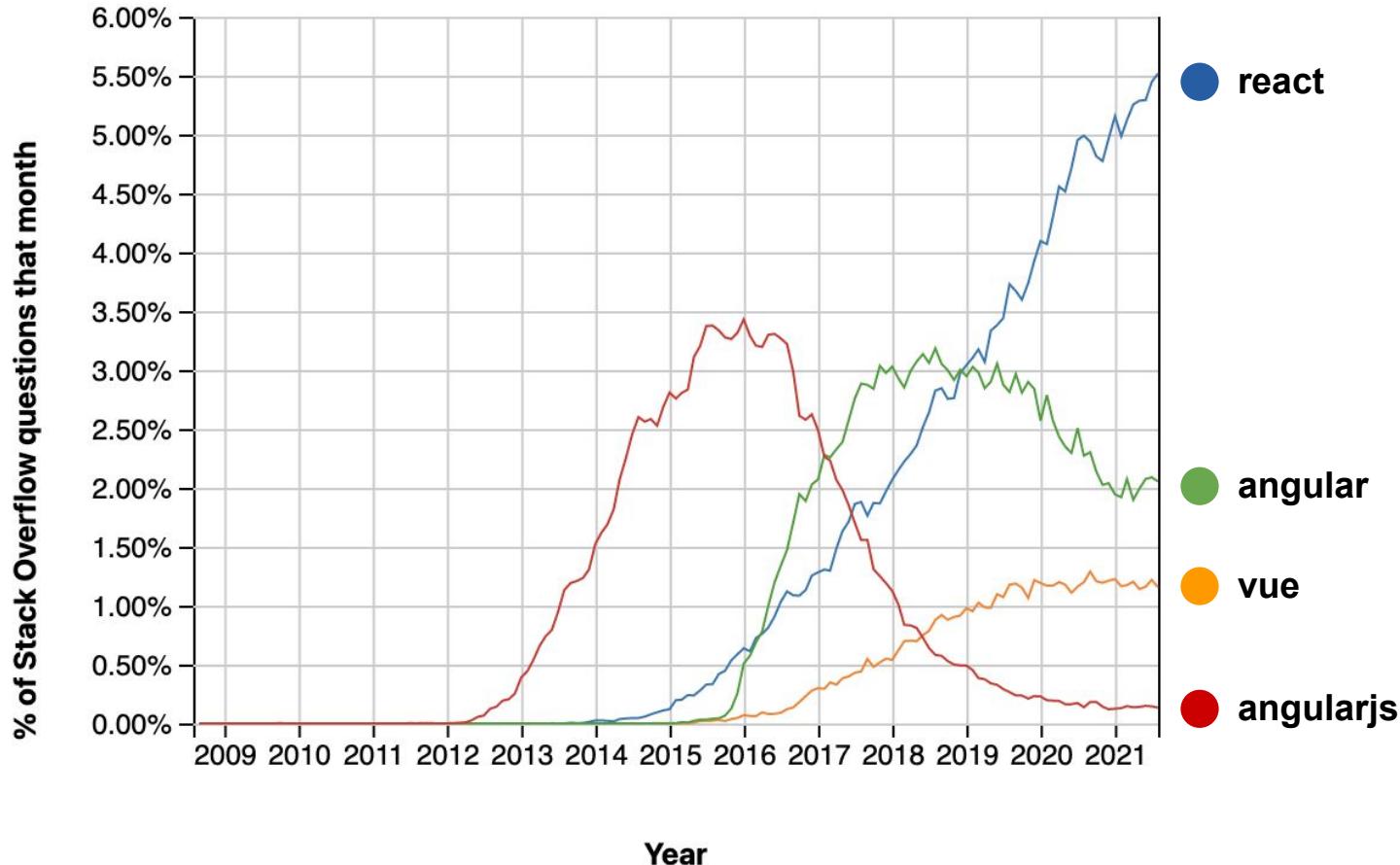
---

2021

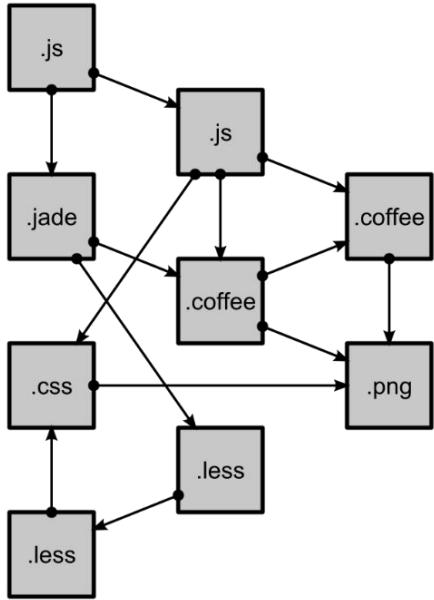
replaceAll(), Promise.any(), AggregateError, logical assignment  
operators (??=, &&=, ||=), WeakRef, FinalizationRegistry

# BABEL

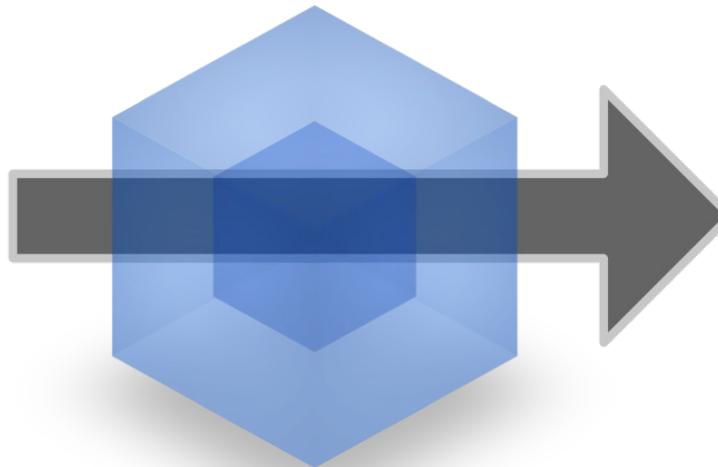




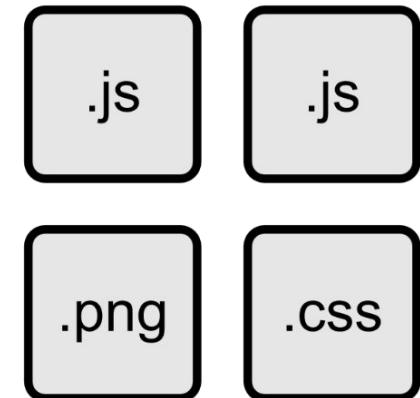
Source: [stackoverflow](#)



modules  
with dependencies



**webpack**  
MODULE BUNDLER



static  
assets



# The Future & What We Can Learn From The Past

ES.Next / New Tech

Why does this matter to me?

# ES.Next - The Process

## Stage 0 / Proposal

A new possible feature that is planned to be presented to TC39

## Stage 1 / Proposal

Still primitive, but TC39 is willing to spend time solving the proposals potential challenges

## Stage 2 / Draft

TC39 expects the proposal to be included in the standard, critical aspects and issues are resolved

## Stage 3 / Candidate

The proposal is meticulously reviewed, the final stage before being included in the standard

## Stage 4 / Finished

This addition is ready for inclusion in the next ECMA version

# ES.Next - Current Proposals

## Stage 0 / Proposal

`new` operator `::` for `this` binding, nested `import` declarations, object shorthand improvements

## Stage 1 / Proposal

`async do` expressions, array deduplication, array equality, new `slice` notation `[x:x]`

## Stage 2 / Draft

New `Array.prototype` methods that create a copy of the array, `throw` expressions

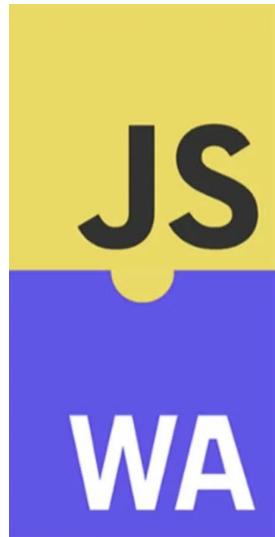
## Stage 3 / Candidate

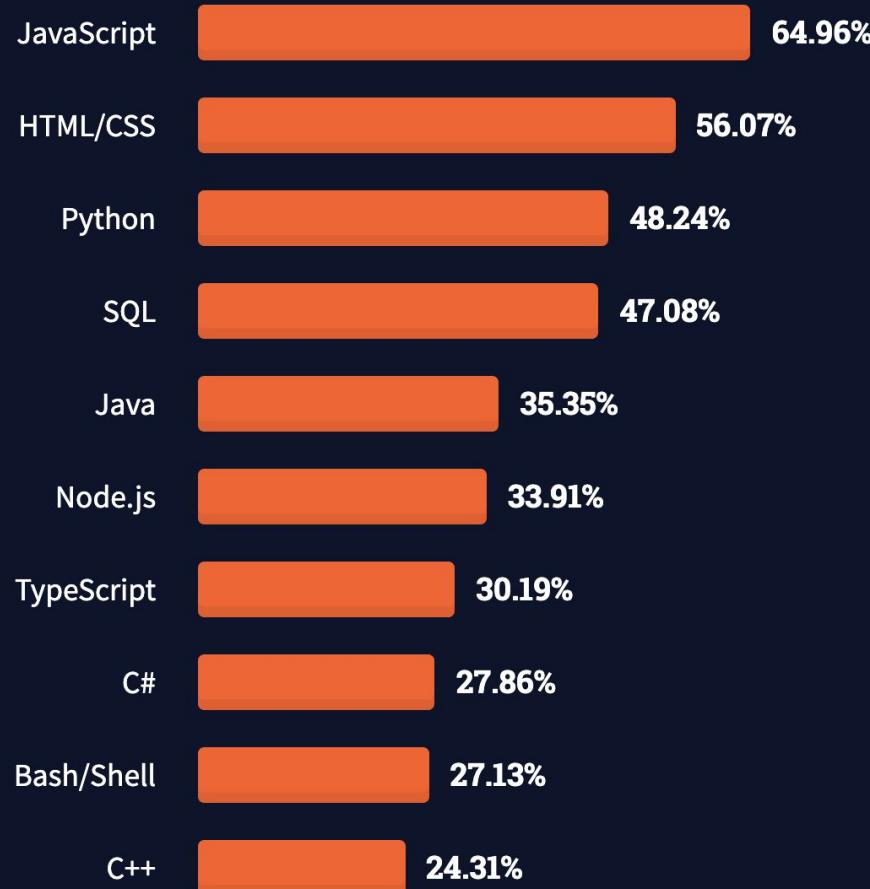
`.findLast()` and `.findLastIndex()`, common `JSON` imports

## Stage 4 / Finished

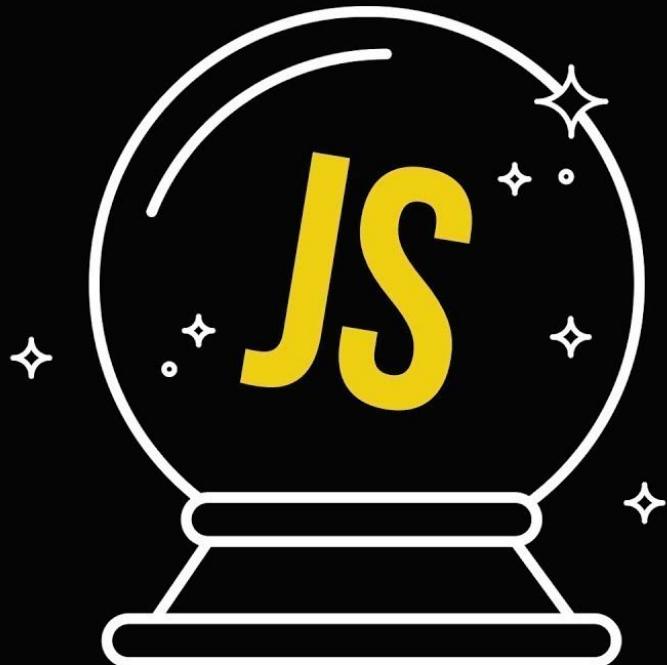
Top level `await`, `.at()` method, `Object.hasOwn()` method to make `Object.prototype.hasOwnProperty` more accessible

# The Future Is Now





# The Far Future



# Why Do I Care?

# Resources

[laurasprauer.com/momentum-2021](http://laurasprauer.com/momentum-2021)

Ample blog link

JavaScript: The First 20 Years by Allen Wirfs-Brock & Brendan Eich

TC39/proposals - github

# Questions?

...