



What the Heck-mascript?!


— Jean Lange, PGH TechFest 2021 —






**Jean Lange** 1 month ago



**Jean Lange** 1 month ago

JS, why are you like this?

**Jean Lange** 12:33 PM




Hi, friends! I've signed myself up to give a talk which I've loosely described as:


Title: Why is JS so ???! a brief history

Description: Why there are so many different ways to do things in JS, how to pick the framework/library/version to actually use.


I need help brainstorming a REAL title!

**Jean Lange** 1 month ago

JS: just don't crash the browser

**Zac Littleberry** 1 month ago

StrangeScript

**Remy Porter** 1 month ago

Tech Isn't a Meritocracy: the JavaScript Story

Dan Oswalt brainstormed the name of this talk on Code & Supply chat - thank you, Dan, and I strongly recommend this method of titling your talks.

Hi, I'm Jean!

Ruby

sticky
notes

board
games

git

jean@academypgh.com
github.com/jeanlange/slides



Credit to Miro for the sticky notes used in this presentation!

Look, here's the story

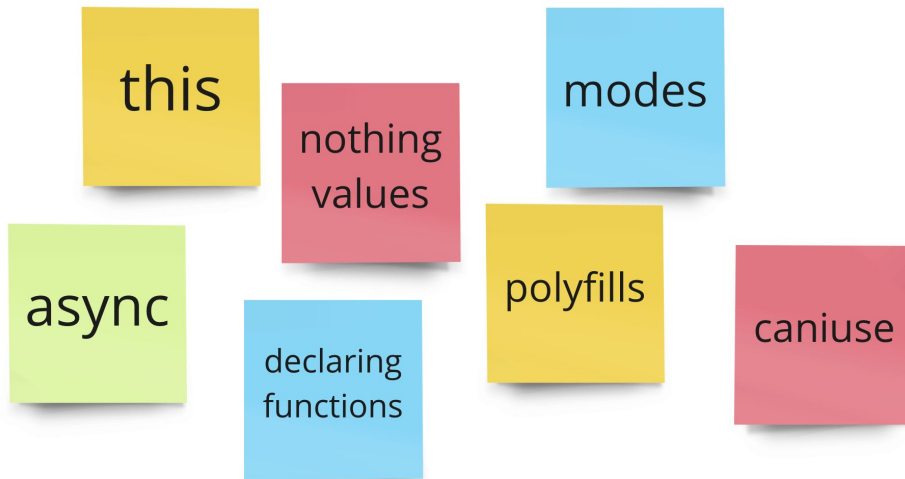
1. JavaScript is a screwy language and has a screwy ecosystem.
2. But it is used everywhere! By everyone!
3. Let's take a look at its history.
4. And talk about how we can make good choices about JS in this world we find ourselves in now.

JS is screwy



- When I'm teaching at Academy, when I get to JS!
- What is JS all about?
- Name
 - Java vs JS
 - ES vs JS. ES2015 = ES6.
- WAT - <https://www.destroyallsoftware.com/talks/wat> 1:20 - 2:42, especially
- Frameworks/package managers:
 - Let's play a game - how many frameworks can you think of for Ruby, C#, Python? What about JS?
 - What about package managers? Where do you go to get your libraries for Ruby? C#? Python? What about JS?

“some technical details”



Some things that confuse our students when they get to JS - things that are different from what they expect!

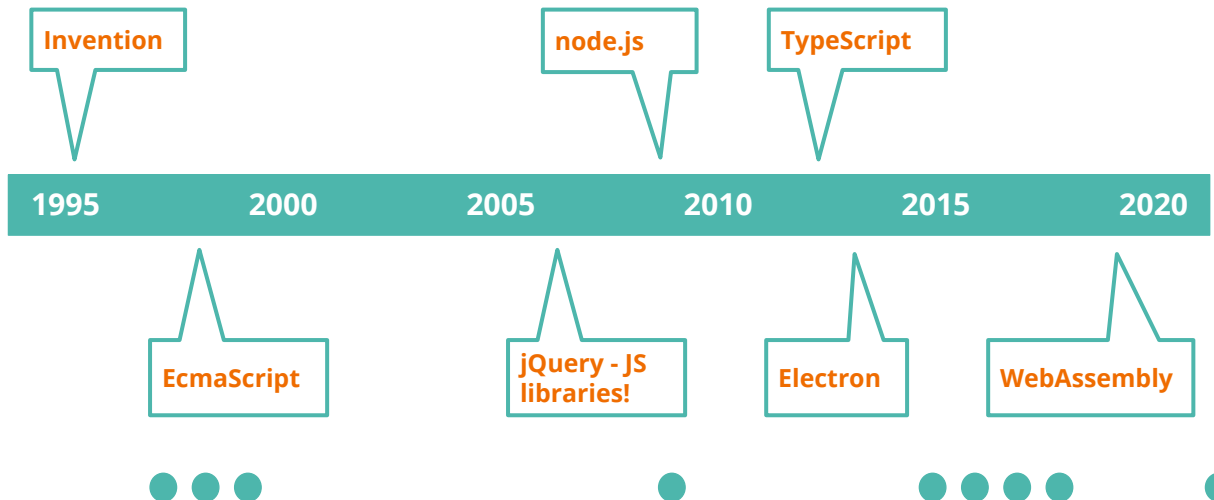
JS is everywhere

#1

97.5%

platforms

How did we get here?



- Web invented in 1989 - static HTML, URIs.
- Invention (1995)
 - Brandon Eich at Netscape. “Easier Java”. 10 days. Mocha -> LiveScript -> JavaScript (marketing team)
 - You know how the `<script>` tag has that ‘lang’ attribute? JS was intended to be one of many! But it took over.
- EcmaScript (1997)
 - used to be European Computer Manufacturer’s Association - standards org.
 - The ES standard is also recognized by ISO under a different number, but is the same specification. Interestingly, the standard is NOT controlled by the W3C because they didn’t want to get into whole languages.
<https://www.quora.com/Why-was-JavaScript-standardized-by-ECMA-and-not-W3C>
 - Ecma also responsible for (for example): language specs (C++, C#, Eiffel), file systems (FAT16/32, CD-ROM) and character sets.
 - Name of standard is because of copyright/trademark issues with the name JS.
 - JS has more than ES does - ES is a language specification, and JS adds DOM (Document Object Model - `document.getElementById`, eg) + BOM (Browser Object Model - location, history, back, eg)
 - There are other implementations of ES, like JScript (Microsoft) and ActionScript (Flash), but they are not nearly as popular.

- An illuminating SO question:
<https://stackoverflow.com/questions/912479/what-is-the-difference-between-javascript-and-ecmascript>
- Since its release in 1997, it's gone through periods of rapid and less rapid change! See https://www.w3schools.com/js/js_versions.asp. 1999 - 2009 for 3-5, then 2015 - 2018. ES2021 in June 2021
- jQuery (2006)
 - First big megacollection of JS functionalities, unify functions that work on all browsers.
 - still used by 78% of websites (August 2021):
https://w3techs.com/technologies/overview/javascript_library
 - This site also reports on other JS libraries, lots of familiar names in there! Includes:
 - graphical libraries like Bootstrap or Highcharts
 - Functionality libraries like lodash
 - Frameworks like React, Vue, and Angular
- Node.js (2009)
 - From OpenJs Foundation. Moves JavaScript to not-a-browser (using a browser engine, though!). JS runtime env, like ruby
 - Npm (ok, and yarn).
 - Express.js
 - Dev environment -> static, deployable assets.
- TypeScript (2012)
 - From Microsoft - another (early) implementation of ES2015 standard!
 - Adds optional typing. Strict superset of JS. Transpiles to JS.
 - Transcompilation
- Electron (2013)
 - Brought web tech (including JS) to desktop applications
- WebAssembly (2019)
 - W3C standard alongside HTML & CSS
 - New (assembly/binary) language for the web; can be written into from any language (in theory), major browsers support it with flags.
 - Focus on performance. Has some immaturities.

Summary trends to think about/notice:

- Proliferation of libraries
- SPAs (2012sish - squishy number)

Unless otherwise noted, info comes from Wikipedia pages.

Reframing questions about why JS is so ?!*&!

web

lure

These are my opinions!

- Web
 - Grew with it
 - DON'T BREAK THINGS
 - Falls into that singular tech bucket like HTML & CSS
- Lure
 - Attract non-devs
 - Gradually normal engineering practices surround it

How do I make choices?

shiny?

fit

ilities

- JS IS a reasonable - and pretty safe - choice.
- Same as any other software engineering choice!
- ilities
 - popularity/learnability/support/hireability/retainability/longevity - github stars
 - compatibility
- fit to your application - was it built to solve the kind of problems you're having?
 - react
 - angular - corporate style SPAs by google - like gmail
 - angular vs angular 2+ (copy the popularity of react)
 - vue?
 - svelte - goddammit, redux, but oooh the display of react. better datahandling react.
- New shiny?

Questions!

jean@academypgh.com
github.com/jeanlange/slides

