

CS 4530: Fundamentals of Software Engineering

Module 3, Lesson 5

JavaScript in the Browser

Rob Simmons

Khoury College of Computer Sciences

© 2025 Released under the [CC BY-SA](#) license

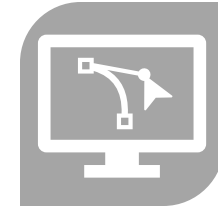
So, software engineering must encompass:



PEOPLE



PROCESSES



PROGRAMS

PLANNING



ORGANIZING



IMPLEMENTING



We're gonna be
stuck over here for
a bit.

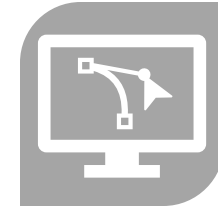
So, software engineering must encompass:



PEOPLE



PROCESSES



PROGRAMS

PLANNING



ORGANIZING



IMPLEMENTING



But also, there's
something to be said for
the firehose

We're gonna be
stuck over here for
a bit.

Learning Objectives for this Lesson

- By the end of this lesson, you should be able to:
 - Describe the relationship between JavaScript, TypeScript, HTML, and the Document Object Model (DOM)
 - Explain the historical and present value of “frontend tooling”


HTML: The Markup Language of the Web

- Language for describing structure of a document
- Denotes hierarchy of elements
- Browser turns this into a “live” version — the Document Object Model (DOM)
- (Inspector demo)



Frontend tooling

Approximate timeline:

- 2006: jQuery makes direct DOM manipulation fun
 - 2010: NodeJS
 - 2011: React (Facebook internal)
 - 2012: TypeScript
 - 2013: React&JSX (public)
- 
- NONE of these run
in the browser directly

People got very used to having to compile their hundreds of not-browser-JavaScript files to one Browser JavaScript to do *anything* (makes dev slow!)

Frontend tooling

Many build steps are technically unnecessary...
but it's been a long, long road

- 2015: ES Modules introduced (import)
- 2019: ES Modules in NodeJS, all major browsers support
- 2020: Vite explores using ES Modules to avoid regenerating the one-big-JavaScript-file for every minor change while developing code
- 2022: All supported versions of NodeJS support ES Modules
- 2022: (Still-controversial, possibly doomed) proposal for “types as comments” in the JavaScript standard
- January 7, 2025: NodeJS v23.6 allows direct running of TypeScript files with no extra configuration (just ignores the types)
- May 6, 2025: NodeJS 24.x is the first long-term supported version of NodeJS that allows direct running of TypeScript files

Frontend tooling

Frontend tooling is not going anywhere:

- It reduces the size of the JS that ever needs to be downloaded
- Frontend tooling enables powerful browser caching strategies

