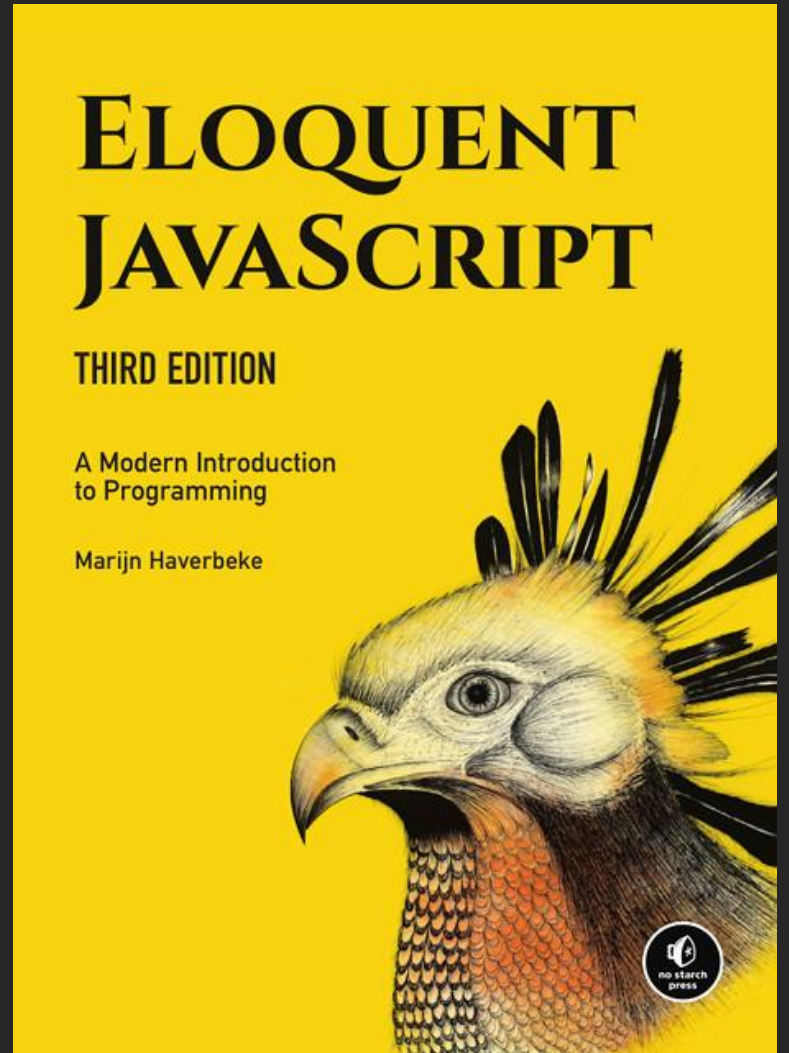


# Programming JS

Margit Tennosaar

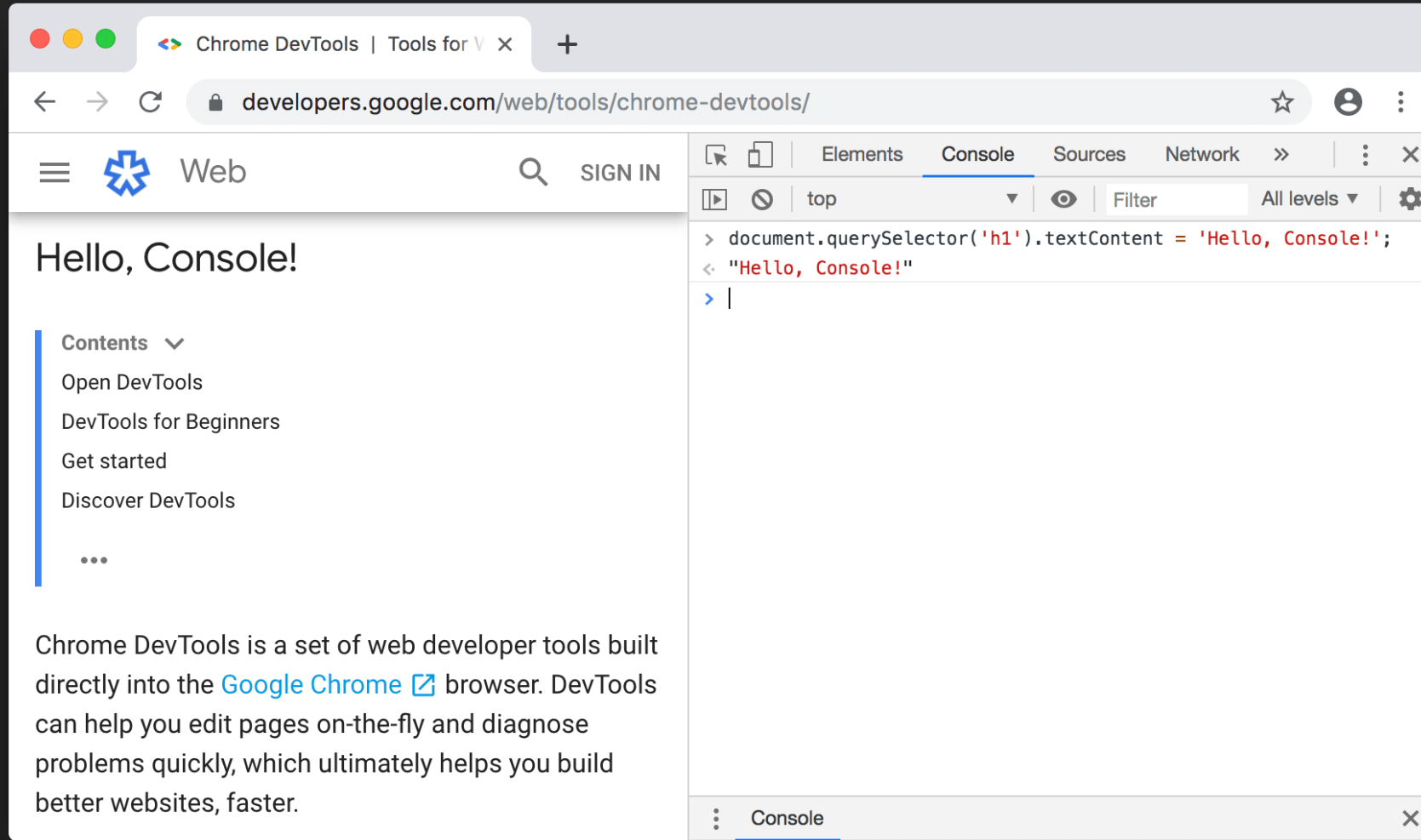
Week	Topic	Principles Needed to Know
1	What is JavaScript, History, Setting Up environment, CLI	Understanding the history of JavaScript, setting up the development environment, basic CLI commands, variables, data types, basic operators
2	Variables, Data Types, Operators, Basic Functions	Understanding variables, data types, operators, and basic functions
3	Conditionals, Loops, Scope Concepts	Mastering conditionals, loops, and understanding scope
4	DOM Basics, Selecting and Manipulating Elements, Functions	Basic DOM manipulation, element selection, and functions
5	Arrays, Arrow Functions	Manipulating arrays and using arrow functions
6	JSON, Objects	Understanding JSON and objects
7	Error Handling, JS Writing Rules	Handling errors, learning best practices in writing JavaScript
8	Asynchronous JS, LocalStorage, Fetch and APIs	Understanding asynchronous JavaScript, promises, async/await, working with LocalStorage, and APIs
9	Capstone Project and Presentation	Integrating concepts into a final project, best practices, review, and capstone project presentation

## Course cookbook



<https://eloquentjavascript.net/>

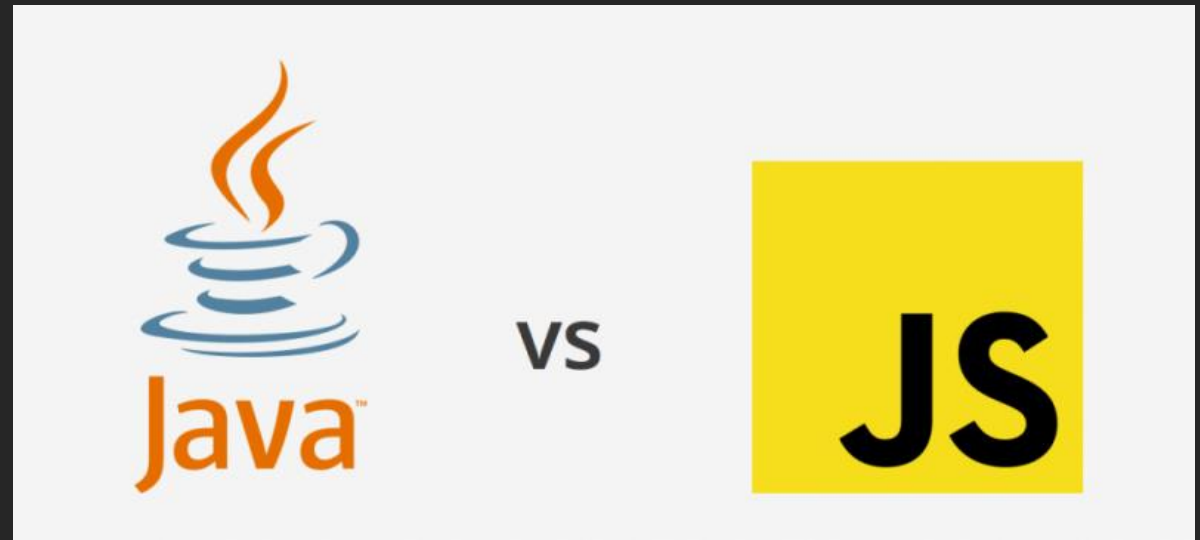
# Developer console



## What is JavaScript?

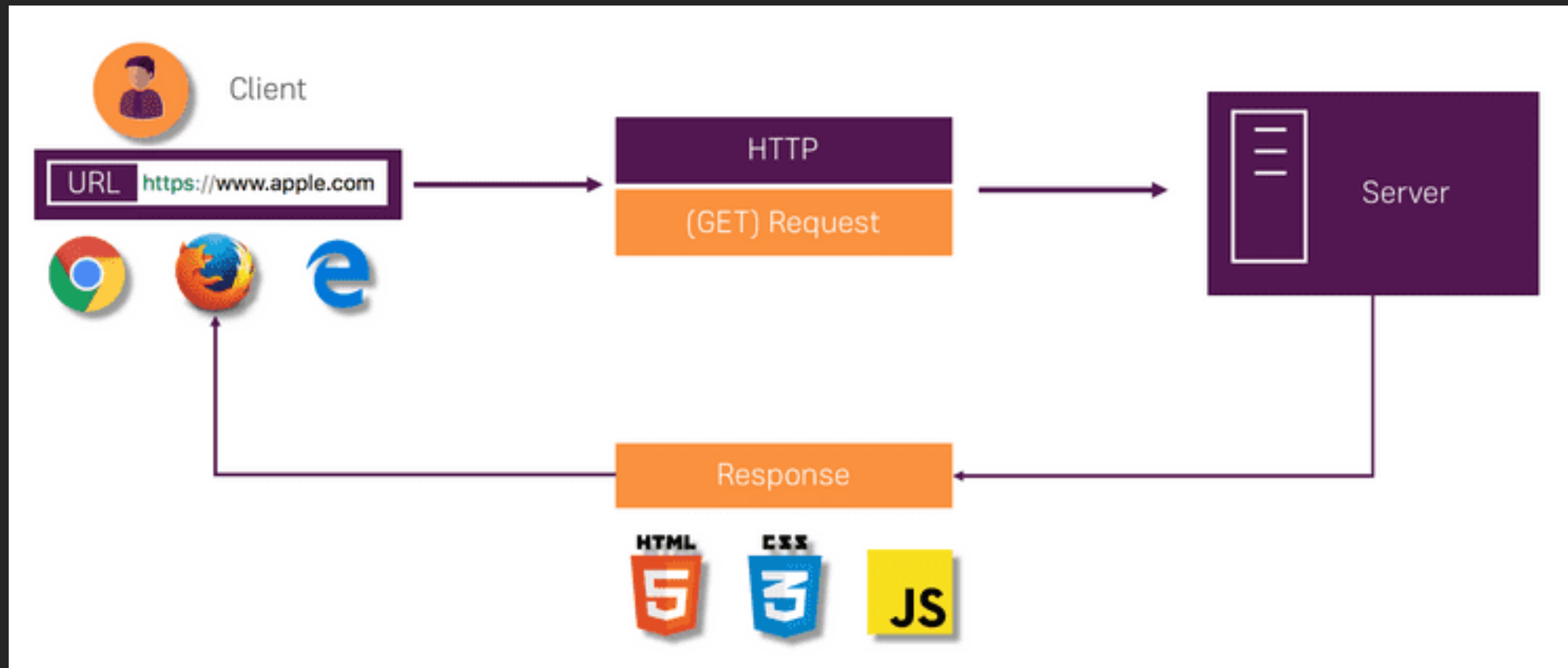
JavaScript is **a scripting language, one** of the three core languages used to develop websites. Whereas HTML and CSS give a website structure and style, JavaScript lets you add functionality and behaviours to your website, allowing visitors to interact with content in imaginative ways.

What is the difference  
between Java and  
JavaScript?



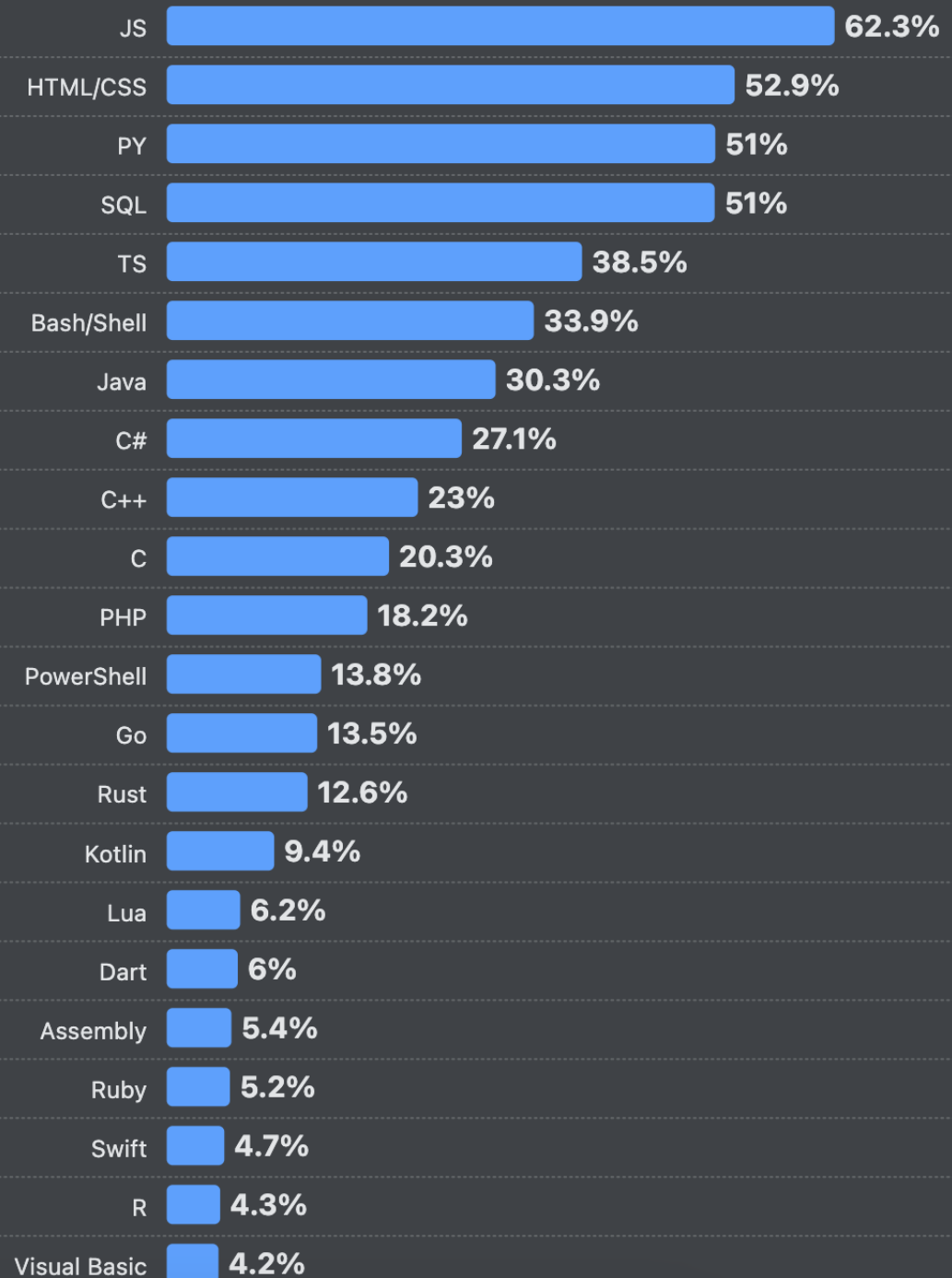
Same as between a car and  
a carpet...

Feature	JavaScript	Java	C/C++	Python	PHP
Paradigm	Multi-paradigm: Event-driven, functional	Object-oriented, class-based	Procedural, object- oriented (C++)	Multi-paradigm: Object-oriented, procedural	Server-side scripting
Typing	Dynamic	Static	Static	Dynamic	Dynamic
Execution Environment	Browser, server (Node.js)	Java Virtual Machine (JVM)	System-level, standalone applications	Server-side, standalone applications	Server-side (Web development)
Syntax	Prototype-based, less formal	Strict, verbose	Low-level, verbose	Easy to read, less formal	Embedded in HTML
Primary Use	Web development, real-time applications	Enterprise applications, Android apps	System/Software development, applications	Data science, web development, automation	Web development





# Most popular programming, scripting, and markup languages



<https://survey.stackoverflow.co/2024/technology#most-popular-technologies-language>

# ECMAScript

JavaScript was invented by Brendan Eich in 1995. It was developed for Netscape 2. After Netscape handed JavaScript over to ECMA, the Mozilla foundation continued to develop JavaScript for the Firefox browser.

JavaScript standard is commonly used for client-side scripting.

**1995:** Created by Brendan Eich at Netscape; initially called LiveScript, renamed to JavaScript.

**1997:** JavaScript is standardized as ECMAScript to ensure consistent behavior across browsers.

**2000s:** Adoption of AJAX enables dynamic web applications; jQuery simplifies cross-browser scripting.

**2009:** Introduction of Node.js allows JavaScript to run server-side, expanding its capabilities beyond the browser.

**2010s:** The rise of front-end frameworks (Angular, React, Vue.js) revolutionizes web application development.

**2015-onwards:** ECMAScript updates (ES6 and beyond) introduce new features like arrow functions, promises, classes, and modules, enhancing the language's capabilities.

**Today:** JavaScript is a versatile, widely-used language for both client-side and server-side applications, with a rich ecosystem of libraries and tools.

Year	Version Name	Description
1997	ECMAScript 1	The first edition, establishing the core features of JavaScript.
1998	ECMAScript 2	Minor revisions and bug fixes to align with the ISO/IEC 16262 standard.
1999	ECMAScript 3	Introduced regular expressions, better string handling, and try-catch for error handling.
2009	ES5	Added strict mode, JSON support, and improved object property manipulation methods.
2015	ES6/ES2015	Introduced let/const, arrow functions, classes, modules, promises, and template literals.
2016	ES7/ES2016	Added exponentiation operator (**) and Array.prototype.includes().
2017	ES8/ES2017	Introduced async/await, Object.entries(), Object.values(), and string padding.
2018	ES9/ES2018	Added rest/spread properties, asynchronous iteration, and Promise.finally().
2019	ES10/ES2019	Introduced Array.flat(), Array.flatMap(), Object.fromEntries(), and optional catch binding.
2020	ES11/ES2020	Added BigInt, dynamic import(), nullish coalescing operator (??), and optional chaining (?.).
2021	ES12/ES2021	Introduced logical assignment operators, replaceAll(), and Promise.any().
2022	ES13/ES2022	Added top-level await, new array methods, and enhanced private fields in classes.
2023	ES14/ES2023	Introduced Array.prototype.toSorted(), Array.prototype.with(), and Symbol.prototype.description.

<https://developer.mozilla.org/en-US/docs/Web/JavaScript>