



PDF 2 — How JavaScript Works

Browsers, JavaScript Engines, and Code Execution Explained

Learn what happens behind the scenes when JavaScript runs in a browser. Understand browsers, JavaScript engines, parsing, compilation, execution, memory, and the execution environment before writing your first program.

Table of Contents

01	Introduction
02	Why Beginners Should Learn How JavaScript Works
03	What Happens When You Open a Website?
04	What Is a Browser?
05	The Browser Rendering Process
06	Where JavaScript Fits In
07	What Is a JavaScript Engine?
08	How JavaScript Code Is Executed
09	Parsing and Compilation
10	Execution Context (Beginner Introduction)
11	Memory and Variables
12	Browser APIs
13	JavaScript Is Single-Threaded
14	Why Understanding Execution Matters
15	Real-World Examples
16	Common Beginner Mistakes
17	Practical Action Plan
18	Mini Project
19	Key Takeaways
20	Summary Page
21	JavaScript Execution Cheat Sheet
22	Related Resources
23	Recommended Next Learning Path

SECTION 01

Introduction

When you write your first JavaScript program, it can feel like magic.

You type code, refresh the page, and something happens.

But what actually happens between writing code and seeing the result?

Understanding this process helps you debug problems, write better code, and develop confidence as a programmer.

This PDF explains the journey of JavaScript from your source code to a running program.

SECTION 02

Why Beginners Should Learn How JavaScript Works

Many courses begin with variables and syntax.

While syntax is important, it answers how to write code, not how code runs.

Knowing how JavaScript executes will help you:

- understand errors more easily
- debug with confidence
- build a stronger mental model
- avoid memorizing code without understanding it

Think of it as learning how a car engine works before trying to repair one.

SECTION 03

What Happens When You Open a Website?

When you visit a webpage, your browser performs several steps:

- 1 Requests the webpage from a server.
- 2 Downloads the HTML document.
- 3 Downloads linked CSS files.
- 4 Downloads JavaScript files.
- 5 Builds the page structure.
- 6 Applies styles.
- 7 Executes JavaScript.
- 8 Displays the finished page.

These steps happen very quickly, often in less than a second.

Understanding this sequence explains why JavaScript can change webpage content after it has loaded.

SECTION 04

What Is a Browser?

A browser is software that retrieves, interprets, and displays web content.

Its responsibilities include:

- downloading resources
- rendering HTML and CSS
- executing JavaScript
- handling user interactions
- communicating with web servers

Without a browser, HTML, CSS, and JavaScript would simply be text files.

SECTION 05

The Browser Rendering Process

The browser transforms your code into a visible webpage.

The simplified process is:

- Read HTML.
- Build the document structure.
- Apply CSS styling.
- Calculate layout.
- Paint the page.
- Execute JavaScript.
- Update the page if JavaScript changes anything.

This process repeats whenever JavaScript modifies the document.

SECTION 06

Where JavaScript Fits In

JavaScript acts as the logic layer of a webpage.

It can:

- change text
- update styles
- add new elements
- remove existing elements
- respond to user interactions
- fetch data from servers

Rather than replacing HTML or CSS, JavaScript works alongside them.

SECTION 07

What Is a JavaScript Engine?

A JavaScript engine is the component inside a browser that understands and executes JavaScript code.

Its responsibilities include:

- reading source code
- checking syntax
- converting code into instructions
- executing those instructions
- managing memory
- optimizing performance

Different browsers use different engines, but they all serve the same purpose.

SECTION 08

How JavaScript Code Is Executed

When JavaScript runs, the engine follows a series of steps:

- 1 Read the source code.
- 2 Check for syntax errors.
- 3 Parse the code.
- 4 Prepare memory.
- 5 Execute statements in order.
- 6 Produce results.

Most beginner programs execute from top to bottom unless control structures change the flow.

SECTION 09

Parsing and Compilation

Before execution, the engine analyzes your code.

This stage identifies:

- syntax errors
- keywords
- variable declarations
- function definitions

Modern JavaScript engines also optimize code before executing it, allowing programs to run much faster than early JavaScript implementations.

SECTION 10

Execution Context (Beginner Introduction)

JavaScript needs an environment in which to execute code.

This environment is called an execution context.

It keeps track of:

- variables
- functions
- the current line of execution

For beginners, it's enough to understand that JavaScript always knows:

- what code is currently running
- what information is available
- where execution should continue next

You'll explore execution contexts in more depth later in the curriculum.

SECTION 11

Memory and Variables

Whenever your program stores information, the JavaScript engine places it in memory.

Examples include:

- numbers
- text
- arrays
- objects

Variables act as labels that allow your program to retrieve this stored information when needed.

Understanding memory now will make variables much easier to learn in the next PDF.

SECTION 12

Browser APIs

The browser provides additional capabilities through Browser APIs.

These allow JavaScript to:

- interact with the webpage
- respond to clicks
- use timers
- access storage
- communicate over the network

Although JavaScript controls these features, the browser itself provides them.

SECTION 13

JavaScript Is Single-Threaded

JavaScript executes one task at a time.

This means it processes instructions sequentially.

While browsers can perform other tasks simultaneously, JavaScript itself follows a single main execution thread.

Later in the curriculum, you'll learn how asynchronous programming allows JavaScript to work efficiently despite this limitation.

SECTION 14

Why Understanding Execution Matters

Knowing how JavaScript executes helps you:

- debug errors
- predict program behavior
- write cleaner code
- understand performance
- learn advanced concepts more easily

This knowledge becomes increasingly valuable as your applications grow.

Learn More

What Is JavaScript? A Complete Beginner's Guide to the Language That Powers the Web
Read this resource to understand JavaScript's role in modern web development before diving deeper into execution.

SECTION 15

Real-World Examples

Form Validation

When a user submits a form, JavaScript executes validation logic before the browser sends the data to the server.

Image Slider

JavaScript updates which image is displayed based on user interaction.

Weather Dashboard

JavaScript requests new weather information, receives the response, and updates the interface without reloading the page.

haas.dev

Interactive quizzes, search filters, and downloadable resources all rely on JavaScript executing efficiently within the browser.

SECTION 16

Common Beginner Mistakes

- Assuming JavaScript runs before HTML is available.
- Confusing the browser with the JavaScript engine.
- Believing JavaScript performs every browser function itself.
- Ignoring execution order when debugging.
- Focusing only on syntax rather than understanding the runtime.

SECTION 17

Practical Action Plan

Open your browser's Developer Tools.

Visit several websites and observe:

- downloaded resources
- HTML
- CSS
- JavaScript files

Notice how multiple resources work together to create a single webpage.

Understanding the environment is the first step toward understanding programming.

SECTION 18

Mini Project

Create a simple webpage containing:

- a heading
- a paragraph
- a button

Add a JavaScript file (without writing logic yet).

Use Developer Tools to confirm that the browser downloads and executes the script successfully.

The goal is to observe the execution process rather than build functionality.

SECTION 19

Key Takeaways

- Browsers download and render webpages.
- JavaScript engines execute JavaScript code.
- Code is parsed before execution.
- Variables and functions exist within an execution context.
- Browser APIs extend JavaScript's capabilities.
- Understanding execution builds a strong programming foundation.

Summary Page

JAVASCRIPT EXECUTION CHEAT SHEET

- ☒ Browser downloads webpage resources.
- ☒ HTML creates structure.
- ☒ CSS applies styling.
- ☒ JavaScript engine executes code.
- ☒ Parsing checks syntax before execution.
- ☒ Variables are stored in memory.
- ☒ Browser APIs provide additional functionality.
- ☒ JavaScript executes one task at a time.

SECTION 22

Related Resources

What Is JavaScript? A Complete Beginner's Guide to the Language That Powers the Web

Why read it: Understand JavaScript's purpose before learning how it executes.

Build Your First Complete Responsive Website

Why read it: See how JavaScript fits into a complete frontend project.

Accessibility in Frontend Development

Why read it: Ensure future JavaScript interactions remain accessible.

Recommended Next Learning Path

STEP 1 **What Is JavaScript?**

STEP 2 **How JavaScript Works (Current PDF)**

STEP 3 **Writing Your First JavaScript Program**

STEP 4 **Variables & Data Types**

STEP 5 **Operators & Expressions**

STEP 6 **Input, Output & User Interaction**