A Brief Introduction to JavaScript

Introduction to JavaScript

You have already written a line or two of JavaScript, but what exactly is JavaScript and what can we do with it? In this lecture, we will set the stage for the rest of the course. It is really important that you follow this one until the end.

We can define JavaScript in many different ways. My definition is that it is a high-level, object-oriented, multi-paradigm programming language. But what does any of that actually mean? Let's deconstruct this a little bit to make some sense out of it.

Programming Language

A programming language is basically a tool that allows us to write code that instructs a computer to do something. This is our main goal of using JavaScript.

High-Level Language

JavaScript is a high-level language, which means we do not have to think about complex details such as managing the computer's memory while it runs a program. JavaScript provides many abstractions over these small details, making the language easier to write and learn.

Object-Oriented

JavaScript is object-oriented, meaning the language is mostly based on the concept of objects for storing most kinds of data. We will learn all about object-oriented programming throughout this course.

Multi-Paradigm

JavaScript is also a multi-paradigm language, which means it is flexible and versatile. We can use different programming styles such as imperative and declarative programming. These styles are just different ways of structuring our code. You will learn all about this throughout the course.

Role of JavaScript in Web Development

Now that we know what JavaScript is, at least kind of, what role does it play in web development? In other words, what do we actually use it for? To answer this, let's look at the three core technologies of the web: HTML, CSS, and JavaScript.

These three technologies work together to create beautiful, interactive, and dynamic websites or web applications. HTML is responsible for the content of the page — the text, images, buttons, and all the contents you see on the webpage are written in HTML.

CSS is responsible for the presentation of that content, basically for styling and laying out the elements on a webpage. Finally, JavaScript is the real programming language of the internet. It allows developers to add dynamic and interactive effects to any webpage.

JavaScript is also used to manipulate content or CSS, load data from remote servers, and build entire applications in the browser, which we call web applications.

Analogy: Nouns, Adjectives, and Verbs

To make this separation of roles easier to understand, we can use the analogy of nouns, adjectives, and verbs. HTML represents the nouns, for example, the P element is a paragraph, which is the noun here. CSS is the adjective because it describes the noun, like saying the paragraph text is red. JavaScript is the verb, like saying hide the paragraph, where we are doing something.

Real Example: Twitter Web Application

Let's look at a real example to make it clearer what JavaScript is capable of. On the Twitter web application, when opening a profile page, you see two rotating spinners, indicating JavaScript is loading data from the Twitter service in the background.

As the data arrives, JavaScript hides these spinners and shows the loaded content. This is a cool example of manipulating styles and content. There are also simpler dynamic effects, like showing the tweet box when clicking the tweet button and hiding it when clicking outside the box. JavaScript also displays user information as you hover over any user, probably loading this data on the fly.

All these dynamic and interactive features are powered by JavaScript. By the end of this course, you will know how to use all these techniques.

Recap and Further Possibilities

JavaScript allows us to add dynamic effects to pages and build entire web applications in the browser that feel like native apps on computers and phones. JavaScript made modern web development and the modern web itself possible.

If you are familiar with web development, you may have heard of modern JavaScript libraries and frameworks such as React, Angular, or Vue. These tools make writing modern, large-scale web applications easier and faster. However, all these frameworks and libraries are based 100% on JavaScript.

It is important to become really good at JavaScript before learning and using any of these frameworks, rather than jumping into a framework after writing just a few lines of JavaScript. Your library of choice might not always be as popular as it is today, so mastering JavaScript prepares you to learn whatever is popular next.

Learning JavaScript properly is the single best investment you can make in your web development career right now.

JavaScript Beyond the Browser

JavaScript and the web browser are two separate things. JavaScript can also run outside of web browsers. For example, it can be used on a web server using Node.js, which does not require a browser. This allows us to create backend applications that run on a web server and interact with databases.

When used in the browser, JavaScript creates front-end applications. In this course, we will learn the ins and outs of the JavaScript language itself and how to use it in the browser to create amazing effects. We will be building front-end web applications. In the future, a section about Node.js might be added to show how JavaScript can be used outside the browser environment.

JavaScript for Mobile and Desktop Applications

JavaScript can also be used to build native mobile applications and native desktop applications for any phone or computer operating system. Modern tools like React Native, the Ionic framework, and Electron have changed the development industry, making things possible that were never possible before.

There is almost nothing you cannot build after becoming really good at JavaScript, which is truly mind-blowing. The possibilities are endless.

JavaScript Releases and Versions

Finally, let's briefly talk about JavaScript releases or versions. There was a huge update to the language in 2015, officially called ES2015 but commonly known as ES6 because it followed ES5. ES stands for ECMAScript.

Since ES2015, there has been a yearly new release with new JavaScript features. All these releases starting from ES2015 are considered modern JavaScript. In this course, you will learn modern JavaScript from the beginning. However, it is also important to understand what came before ES2015, so we will cover important aspects of ES5 as well.

We will return to this topic in a dedicated video about JavaScript releases at the end of this section. Now, let's finally get started with the course.

Key Takeaways

- JavaScript is a high-level, object-oriented, multi-paradigm programming language.
- It plays a crucial role in web development alongside HTML and CSS, enabling dynamic and interactive web applications.
- JavaScript runs both in browsers for front-end development and on servers using Node.js for back-end development.
- Mastering JavaScript fundamentals is essential before learning frameworks like React, Angular, or Vue.