An Introduction

▼ What is JavaScript?

JavaScript was initially created to "make web pages alive".

The programs in this language are called *scripts*. They can be written right in a web page's HTML and run automatically as the page loads.

Scripts are provided and executed as plain text. They don't need special preparation or compilation to run.

In this aspect, JavaScript is very different from another language called <u>Java</u>.

▼ What can in-browser JavaScript do?

Modern JavaScript is a "safe" programming language. It does not provide low-level access to memory or the CPU, because it was initially created for browsers which do not require it.

JavaScript's capabilities greatly depend on the environment it's running in. For instance, <u>Node.js</u> supports functions that allow JavaScript to read/write arbitrary files, perform network requests, etc.

In-browser JavaScript can do everything related to webpage manipulation, interaction with the user, and the webserver.

For instance, in-browser JavaScript is able to:

- Add new HTML to the page, change the existing content, modify styles.

- React to user actions, run on mouse clicks, pointer movements, key presses.

- Send requests over the network to remote servers, download and upload files (so-called <u>AJAX</u> and <u>COMET</u> technologies).

- Get and set cookies, ask questions to the visitor, show messages.

- Remember the data on the client-side ("local storage").

▼ Developer Console

Code is prone to errors. You will quite likely make errors… Oh, what am I talking about? You are absolutely going to make errors, at least if you're a human, not a robot.

But in the browser, users don't see errors by default. So, if something goes wrong in the script, we won't see what's broken and can't fix it.

To see errors and get a lot of other useful information about scripts, "developer tools" have been embedded in browsers.

Most developers lean towards Chrome or Firefox for development because those browsers have the best developer tools. Other browsers also provide developer tools, sometimes with special features, but are usually playing "catch-up" to Chrome or Firefox. So most developers have a "favorite" browser and switch to others if a problem is browser-specific.

Developer tools are potent; they have many features. To start, we'll learn how to open them, look at errors, and run JavaScript commands.

▼ Summary

JavaScript was initially created as a browser-only language, but it is now used in many other environments as well.
Today, JavaScript has a unique position as the most widely-adopted browser language, fully integrated with HTML/CSS.
There are many languages that get "transpiled" to JavaScript and provide certain features. It is recommended to take a look at them, at least briefly, after mastering JavaScript.