JavaScript

JavaScript (**JS**) is a lightweight interpreted (or just-in-time compiled) programming language with first-class functions. While it is most well-known as the scripting language for Web pages, many non-browser environments also use it, such as Node.js, Apache CouchDB and Adobe Acrobat. JavaScript is a prototype-based, garbage-collected, dynamic language, supporting multiple paradigms such as imperative, functional, and object-oriented.

JavaScript's dynamic capabilities include runtime object construction, variable parameter lists, function variables, dynamic script creation (via `eval`), object introspection (via `for...in` and `Object` utilities), and source-code recovery (JavaScript functions store their source text and can be retrieved through `toString()`).

This section is dedicated to the JavaScript language itself, and not the parts that are specific to Web pages or other host environments. For information about APIs that are specific to Web pages, please see Web APIs and DOM.

The standards for JavaScript are the ECMAScript Language Specification (ECMA-262) and the ECMAScript Internationalization API specification (ECMA-402). As soon as one browser implements a feature, we try to document it. This means that cases where some proposals for new ECMAScript features have already been implemented in browsers, documentation and examples in MDN articles may use some of those new features. Most of the time, this happens between the stages 3 and 4, and is usually before the spec is officially published.

Do not confuse JavaScript with the Java programming language — **JavaScript is *not* "Interpreted Java"**. Both "Java" and "JavaScript" are trademarks or registered trademarks of Oracle in the U.S. and other countries. However, the two programming languages have very different syntax, semantics, and use.

JavaScript documentation of core language features (pure ECMAScript, for the most part) includes the following:

- The JavaScript guide
- The JavaScript reference

For more information about JavaScript specifications and related technologies, see JavaScript technologies overview.

Beginner's tutorials

Learn how to program in JavaScript from the ground up with our beginner's tutorials.

Your first website: Adding interactivity

This article provides a brief tour of what JavaScript is and how to use it, aimed at people who are completely new to web development.

Dynamic scripting with JavaScript

Our Learn web development section's JavaScript module teaches all the JavaScript fundamentals from the ground up.

JavaScript frameworks and libraries

JavaScript frameworks are an essential part of modern front-end web development, providing developers with tried and tested tools for building scalable, interactive web applications. Many modern companies use frameworks as a standard part of their tooling, so many front-end development jobs now require framework experience. In this set of articles, we aim to give you a comfortable starting point to help you begin learning frameworks.

JavaScript guides

Fundamental language guides

JavaScript Guide

A much more detailed guide to the JavaScript language, aimed at those with previous programming experience either in JavaScript or another language.

Intermediate

Advanced JavaScript objects

The object-oriented nature of JavaScript is important to understand if you want to go further with your knowledge of the language and write more efficient code, therefore we've provided this module to help you.

Asynchronous JavaScript

In this module, we take a look at asynchronous JavaScript, why it is important, and how it can be used to effectively handle potential blocking operations, such as fetching resources from a server.

Client-side web APIs

Explores what APIs are, and how to use some of the most common APIs you'll come across often in your development work.

JavaScript language overview

An overview of the basic syntax and semantics of JavaScript for those coming from other programming languages to get up to speed.

JavaScript data structures

Overview of available data structures in JavaScript.

Equality comparisons and sameness

JavaScript provides three different value comparison operations: strict equality using `===`, loose equality using `==`, and the `Object.is()` method.

Enumerability and ownership of properties

How different methods that visit a group of object properties one-by-one handle the enumerability and ownership of properties.

Closures

mdn web docs

Advanced

Inheritance and the prototype chain

Explanation of the widely misunderstood and underestimated prototype-based inheritance.

Memory Management

Memory life cycle and garbage collection in JavaScript.

Reference

Browse the complete JavaScript reference documentation.

Standard objects

Get to know standard built-in objects: `Array`, `Boolean`, `Error`, `Function`, `JSON`, `Math`, `Number`, `Object`, `RegExp`, `String`, `Map`, `Set`, `WeakMap`, `WeakSet`, and others.

Expressions and operators

Learn more about the behavior of JavaScript's operators `instanceof`, `typeof`, `new`, `this`, the operator precedence, and more.

Statements and declarations

Learn how `do-while`, `for-in`, `for-of`, `try-catch`, `let`, `var`, `const`, `if-else`, `switch`, and more JavaScript statements and keywords work.

Functions

Learn how to work with JavaScript's functions to develop your applications.

Classes

JavaScript classes are the most appropriate way to do object-oriented programming.

Help improve MDN

Was this page helpful to you?

Yes No

Learn how to contribute.

This page was last modified on Jul 8, 2025 by MDN contributors.