

[Technologies](#) ▼[References & Guides](#) ▼[Feedback](#) ▼[Sign in](#) 

JavaScript

JavaScript (JS) is a lightweight interpreted or JIT-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, multi-paradigm, dynamic language, supporting object-oriented, imperative, and declarative (e.g. functional programming) styles. Read more [about JavaScript](#).

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 specific to Web pages, please see [Web APIs](#) and [DOM](#).

The standard for JavaScript is [ECMAScript](#). As of 2012, all [modern browsers](#) fully support ECMAScript 5.1. Older browsers support at least ECMAScript 3. On June 17, 2015, [ECMA International](#) published the sixth major version of ECMAScript, which is officially called ECMAScript 2015, and was initially referred to as ECMAScript 6 or ES6. Since then, ECMAScript standards are on yearly release cycles. This documentation

refers to the latest draft version, which is currently [ECMAScript 2019](#).

Do not confuse JavaScript with the [Java programming language](#). 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 uses.

Tutorials

Learn how to program in JavaScript with guides and tutorials.

For complete beginners



Head over to our [Learning Area JavaScript topic](#) if you want to learn JavaScript but have no previous experience of JavaScript or programming. The complete modules available there are as follows:

JavaScript first steps

Answers some fundamental questions such as "what is JavaScript?", "what does it look like?", and "what can it do?", along with discussing key JavaScript features such as variables, strings, numbers, and arrays.

JavaScript building blocks

Continues our coverage of JavaScript's key fundamental

Reference

Browse the complete [JavaScript reference](#) documentation.

Standard objects

Get to know standard built-in objects [Array](#), [Boolean](#), [Date](#), [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

features, turning our attention to commonly-encountered types of code block such as conditional statements, loops, functions, and events.

Introducing 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.

JavaScript guide

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

Client-side web APIs

When writing client-side JavaScript for web sites or applications, you won't go very far before you start to use APIs — interfaces for manipulating different aspects of the browser and operating system the site is running on, or even data from other web sites or services. In this module we will explore what APIs

JavaScript's functions to develop your applications.

Tools & resources

Helpful tools for writing and debugging your **JavaScript** code.

Firefox Developer Tools

[Scratchpad](#), [Web Console](#), [JavaScript Profiler](#), [Debugger](#), and more.

JavaScript Shells

A JavaScript shell allows you to quickly test snippets of JavaScript code.

TogetherJS

Collaboration made easy. By adding TogetherJS to your site, your users can help each other out on a website in real time!

Stack Overflow

Stack Overflow questions tagged with "JavaScript".

JavaScript versions and release notes

Browse JavaScript's feature history and implementation status.

JSFiddle

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

Edit JavaScript, CSS, HTML and get live results. Use external resources and collaborate with your team online.

A re-introduction to JavaScript

An overview for those who *think* they know about JavaScript.

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.

Closures

A closure is the combination of a function and the lexical environment within which that function was declared.

Advanced

Inheritance and the prototype chain

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

Strict mode

Plunker

Plunker is an online community for creating, collaborating on and sharing your web development ideas. Edit your JavaScript, CSS, HTML files and get live results and file structure.

JSBin

JS Bin is an open source collaborative web development debugging tool.

Strict mode defines that you can not use any variable before initializing it. It is a restricted variant of ECMAScript 5, for faster performance and easier debugging.

JavaScript typed arrays

JavaScript typed arrays provide a mechanism for accessing raw binary data.

Memory Management

Memory life cycle and garbage collection in JavaScript.

Concurrency model and Event Loop

JavaScript has a concurrency model based on an "event loop".

Join the JavaScript community

Choose your preferred method for joining the discussion:

 [Mailing list](#)  [Newsgroup](#)

 [RSS feed](#)

IRC: [#js](#) ( [learn more](#))

ES discuss:  [esdiscuss.org](#)

SpiderMonkey:  [Project page](#)

Twitter:  [@SpiderMonkeyJS](#)
