WebAssembly

WebAssembly is a type of code that can be run in modern web browsers — it is a low-level assembly-like language with a compact binary format that runs with near-native performance and provides languages such as C/C++, C# and Rust with a compilation target so that they can run on the web. It is also designed to run alongside JavaScript, allowing both to work together.

In a Nutshell

WebAssembly has huge implications for the web platform — it provides a way to run code written in multiple languages on the web at near-native speed, with client apps running on the web that previously couldn't have done so.

WebAssembly is designed to complement and run alongside JavaScript — using the WebAssembly JavaScript APIs, you can load WebAssembly modules into a JavaScript app and share functionality between the two. This allows you to take advantage of WebAssembly's performance and power and JavaScript's expressiveness and flexibility in the same app, even if you don't know how to write WebAssembly code.

And what's even better is that it is being developed as a web standard via the W3C WebAssembly Working Group and Community Group with active participation from all major browser vendors.

Guides

WebAssembly concepts

Get started by reading the high-level concepts behind WebAssembly — what it is, why it is so useful, how it fits into the web platform (and beyond), and how to use it.

Compiling a New C/C++ Module to WebAssembly

When you've written code in C/C++, you can then compile it into Wasm using a tool like Emscripten. Let's look at how it works.

Compiling an Existing C Module to WebAssembly

A core use-case for WebAssembly is to take the existing ecosystem of C libraries and allow developers to use them on the web.

Compiling from Rust to WebAssembly

If you've written some Rust code, you can compile it into WebAssembly! This tutorial takes you through all you need to know to compile a Rust project to Wasm and use it in an existing web app.

Loading and running WebAssembly code

After you have a Wasm module, this article covers how to fetch, compile and instantiate it, combining the WebAssembly JavaScript API with the Fetch or XHR APIs.

Using the WebAssembly JavaScript API

Once you've loaded a Wasm module, you'll want to use it. In this article, we show you how to use WebAssembly via the WebAssembly JavaScript API.

Exported WebAssembly functions

Exported WebAssembly functions are the JavaScript reflections of WebAssembly functions, which allow calling WebAssembly code from JavaScript. This article describes what they are.

Understanding WebAssembly text format

This article explains the Wasm text format. This is the low-level textual representation of a Wasm module shown in browser developer tools when debugging.

Converting WebAssembly text format to Wasm

This article provides a guide on how to convert a WebAssembly module written in text format into a Wasm binary.

API reference

WebAssembly instruction reference

Reference documentation with interactive samples for the set of WebAssembly operators.

WebAssembly JavaScript interface

This object acts as the namespace for all WebAssembly-related functionality.

`WebAssembly.Global()`

A `WebAssembly.Global` object represents a global variable instance, accessible from both JavaScript and importable/exportable across one or more `WebAssembly.Module` instances. This allows dynamic linking of multiple modules.

`WebAssembly.Module()`

A `WebAssembly.Module` object contains stateless WebAssembly code that has already been compiled by the browser and can be efficiently shared with Workers, and instantiated multiple times.

`WebAssembly.Instance()`

A `WebAssembly.Instance` object is a stateful, executable instance of a `Module`. `Instance` objects contain all the Exported WebAssembly functions that allow calling into WebAssembly code from JavaScript.

`WebAssembly.compile()`

The `WebAssembly.compile()` function compiles WebAssembly binary code into a `WebAssembly.Module` object.

`WebAssembly.compileStreaming()`

The `WebAssembly.compileStreaming()` function compiles a `WebAssembly.Module` directly from a streamed underlying source.

`WebAssembly.instantiate()`

The `WebAssembly.instantiate()` function allows you to compile and instantiate WebAssembly code.

`WebAssembly.instantiateStreaming()`

The `WebAssembly.instantiateStreaming()` function is the primary API for compiling and instantiating WebAssembly code, returning both a `Module` and its first `Instance`.

`WebAssembly.validate()`

The `WebAssembly.validate()` function validates a given typed array of WebAssembly binary code.

`WebAssembly.Memory()`

A `WebAssembly.Memory` object is a resizable `ArrayBuffer` that holds the raw bytes of memory accessed by an `Instance`.

`WebAssembly.Table()`

A `WebAssembly.Table` object is a resizable typed array of opaque values, like function references, that are accessed by an `Instance`.

`WebAssembly.Tag()`

The `WebAssembly.Tag` object defines a type of WebAssembly exception that can be thrown to/from WebAssembly code.

`WebAssembly.Exception()`

The `WebAssembly.Exception` object represents a runtime exception thrown from WebAssembly to JavaScript, or thrown from JavaScript to a WebAssembly exception handler.

`WebAssembly.CompileError()`

Creates a new WebAssembly `CompileError` object.

`WebAssembly.LinkError()`

Creates a new WebAssembly `LinkError` object.

`WebAssembly.RuntimeError()`

Creates a new WebAssembly `RuntimeError` object.

Examples

- WASMSobel
- See our webassembly-examples repo for a number of other examples.

Specifications

Specification
WebAssembly JavaScript Interface # webassembly-namespace
WebAssembly JavaScript Interface # ref-for-syntax-numtype①⓪
Unknown specification
Unknown specification
Unknown specification
Unknown specification
WebAssembly Core: Garbage Collection # garbage-collection①
Unknown specification
Unknown specification
WebAssembly JavaScript Interface # dom-globaldescriptor-mutable
Unknown specification
Unknown specification
Unknown specification
Unknown specification
Unknown specification

Browser compatibility

webassembly.api

Report problems with this compatibility data on GitHub

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android
api	✓ 57	✓ 16	✓ 52	✓ 44	✓ 11	✓ 57	✓ 52	✓ 43	✓ 11	✓ 7.0	✓ 57
CompileError	✓ 57	✓ 16	✓ 52	✓ 44	✓ 11	✓ 57	✓ 52	✓ 43	✓ 11	✓ 7.0	✓ 57
CompileError() constructor	✓ 57	✓ 16	✓ 52	✓ 44	✓ 11	✓ 57	✓ 52	✓ 43	✓ 11	✓ 7.0	✓ 57
Exception	✓ 95	✓ 95	✓ 100	✓ 81	✓ 15.2	✓ 95	✓ 100	✓ 67	✓ 15.2	✓ 17.0	✓ 95
Exception() constructor	✓ 95	✓ 95	✓ 100	✓ 81	✓ 15.2	✓ 95	✓ 100	✓ 67	✓ 15.2	✓ 17.0	✓ 95
options.traceStack parameter	✓ 95	✓ 95	✓ 100	✓ 81	✓ 15.2	✓ 95	✓ 100	✓ 67	✓ 15.2	✓ 17.0	✓ 95
Exception.getArg	✓ 95	✓ 95	✓ 100	✓ 81	✓ 15.2	✓ 95	✓ 100	✓ 67	✓ 15.2	✓ 17.0	✓ 95
Exception.is	✓ 95	✓ 95	✓ 100	✓ 81	✓ 15.2	✓ 95	✓ 100	✓ 67	✓ 15.2	✓ 17.0	✓ 95
Stack trace	✓ 95	✓ 95	✓ 100	✓ 81	✗ No	✓ 95	✓ 100	✓ 67	✗ No	✓ 17.0	✓ 95
Global	✓ 69	✓ 79	✓ 62	✓ 56	✓ 13.1	✓ 69	✓ 62	✓ 48	✓ 13.4	✓ 10.0	✓ 69
Global() constructor	✓ 69 *	✓ 79 *	✓ 62	✓ 56 *	✓ 13.1	✓ 69 *	✓ 62	✓ 48 *	✓ 13.4	✓ 10.0 *	✓ 69
Global.value	✓ 69	✓ 79	✓ 62	✓ 56	✓ 13.1	✓ 69	✓ 62	✓ 48	✓ 13.4	✓ 10.0	✓ 69
Global.valueOf	✓ 69	✓ 79	✓ 62	✓ 56	✓ 13.1	✓ 69	✓ 62	✓ 48	✓ 13.4	✓ 10.0	✓ 69
Instance	✓ 57	✓ 16	✓ 52	✓ 44	✓ 11	✓ 57	✓ 52	✓ 43	✓ 11	✓ 7.0	✓ 57
Instance() constructor	✓ 57	✓ 16	✓ 52	✓ 44	✓ 11	✓ 57	✓ 52	✓ 43	✓ 11	✓ 7.0	✓ 57
Instance.exports	✓ 57	✓ 16	✓ 52	✓ 44	✓ 11	✓ 57	✓ 52	✓ 43	✓ 11	✓ 7.0	✓ 57
LinkError	✓ 57	✓ 16	✓ 52	✓ 44	✓ 11	✓ 57	✓ 52	✓ 43	✓ 11	✓ 7.0	✓ 57
LinkError() constructor	✓ 57	✓ 16	✓ 52	✓ 44	✓ 11	✓ 57	✓ 52	✓ 43	✓ 11	✓ 7.0	✓ 57
Memory	✓ 57	✓ 16	✓ 52	✓ 44	✓ 11	✓ 57	✓ 52	✓ 43	✓ 11	✓ 7.0	✓ 57
Memory() constructor	✓ 57	✓ 16	✓ 52	✓ 44	✓ 11	✓ 57	✓ 52	✓ 43	✓ 11	✓ 7.0	✓ 57
shared flag	✓ 74	✓ 79	✓ 78	✓ 62	✓ 15.2 *	✓ 88	✓ 79	✓ 63	✓ 15.2 *	✓ 15.0	✗ No
Memory.buffer	✓ 57	✓ 16	✓ 52	✓ 44	✓ 11	✓ 57	✓ 52	✓ 43	✓ 11	✓ 7.0	✓ 57
Memory.grow	✓ 57	✓ 16	✓ 52	✓ 44	✓ 11	✓ 57	✓ 52	✓ 43	✓ 11	✓ 7.0	✓ 57
Module	✓ 57	✓ 16	✓ 52	✓ 44	✓ 11	✓ 57	✓ 52	✓ 43	✓ 11	✓ 7.0	✓ 57

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android
`Module()` constructor	✓ 57	✓ 16	✓ 52	✓ 44	✓ 11	✓ 57	✓ 52	✓ 43	✓ 11	✓ 7.0	✓ 57
`customSections()` static method	✓ 57	✓ 16	✓ 52	✓ 44	✓ 11	✓ 57	✓ 52	✓ 43	✓ 11	✓ 7.0	✓ 57
`exports()` static method	✓ 57	✓ 16	✓ 52	✓ 44	✓ 11	✓ 57	✓ 52	✓ 43	✓ 11	✓ 7.0	✓ 57
`imports()` static method	✓ 57	✓ 16	✓ 52	✓ 44	✓ 11	✓ 57	✓ 52	✓ 43	✓ 11	✓ 7.0	✓ 57
`RuntimeError`	✓ 57	✓ 16	✓ 52	✓ 44	✓ 11	✓ 57	✓ 52	✓ 43	✓ 11	✓ 7.0	✓ 57
`RuntimeError()` constructor	✓ 57	✓ 16	✓ 52	✓ 44	✓ 11	✓ 57	✓ 52	✓ 43	✓ 11	✓ 7.0	✓ 57
`Table`	✓ 57	✓ 16	✓ 52	✓ 44	✓ 11	✓ 57	✓ 52	✓ 43	✓ 11	✓ 7.0	✓ 57
`Table()` constructor	✓ 57	✓ 16	✓ 52	✓ 44	✓ 11	✓ 57	✓ 52	✓ 43	✓ 11	✓ 7.0	✓ 57
`Table.get`	✓ 57	✓ 16	✓ 52	✓ 44	✓ 11	✓ 57	✓ 52	✓ 43	✓ 11	✓ 7.0	✓ 57
`Table.grow`	✓ 57	✓ 16	✓ 52	✓ 44	✓ 11	✓ 57	✓ 52	✓ 43	✓ 11	✓ 7.0	✓ 57
`Table.length`	✓ 57	✓ 16	✓ 52	✓ 44	✓ 11	✓ 57	✓ 52	✓ 43	✓ 11	✓ 7.0	✓ 57
`Table.set`	✓ 57	✓ 16	✓ 52	✓ 44	✓ 11	✓ 57	✓ 52	✓ 43	✓ 11	✓ 7.0	✓ 57
`Tag`	✓ 95	✓ 95	✓ 100	✓ 81	✓ 15.2	✓ 95	✓ 100	✓ 67	✓ 15.2	✓ 17.0	✓ 95
`Tag()` constructor	✓ 95	✓ 95	✓ 100	✓ 81	✓ 15.2	✓ 95	✓ 100	✓ 67	✓ 15.2	✓ 17.0	✓ 95
`Tag.type`	✗ No	✗ No	✗ No	✗ No	✓ 15.2	✗ No	✗ No	✗ No	✓ 15.2	✗ No	✗ No
`compileStreaming()` static method	✓ 61	✓ 16	✓ 58	✓ 47	✓ 15	✓ 61	✓ 58	✓ 45	✓ 15	✓ 8.0	✓ 61
`compile()` static method	✓ 57	✓ 16	✓ 52	✓ 44	✓ 11	✓ 57	✓ 52	✓ 43	✓ 11	✓ 7.0	✓ 57
`instantiateStreaming()` static method	✓ 61	✓ 16	✓ 58	✓ 47	✓ 15	✓ 61	✓ 58	✓ 45	✓ 15	✓ 8.0	✓ 61
`instantiate()` static method	✓ 57	✓ 16	✓ 52	✓ 44	✓ 11	✓ 57	✓ 52	✓ 43	✓ 11	✓ 7.0	✓ 57
`validate()` static method	✓ 57	✓ 16	✓ 52	✓ 44	✓ 11	✓ 57	✓ 52	✓ 43	✓ 11	✓ 7.0	✓ 57

Tip: you can click/tap on a cell for more information.

✓ Full support ✗ No support ⚠ Non-standard. Check cross-browser support before using. * See implementation notes.

🏳 User must explicitly enable this feature.

webassembly.BigInt-to-i64-integration

Report problems with this compatibility data on GitHub

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android	WebView on iOS
`BigInt-to-i64-integration`	✓ 85	✓ 85	✓ 78	✓ 71	✓ 14.1	✓ 85	✓ 79	✓ 60	✓ 14.5	✓ 14.0	✓ 85	✓ 14.5

Tip: you can click/tap on a cell for more information.

✓ Full support

webassembly.bulk-memory-operations

Report problems with this compatibility data on GitHub

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android	WebView on iOS
`bulk-memory-operations`	✓ 75	✓ 80	✓ 78	✓ 62	✓ 15.1	✓ 75	✓ 79	✓ 54	✓ 15.1	✓ 11.0	✓ 75	✓ 15.1

Tip: you can click/tap on a cell for more information.

✓ Full support

webassembly.exception-handling

Report problems with this compatibility data on GitHub

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android	WebView on iOS
`exception-handling`	✓ 95	✓ 95	✓ 100	✓ 81	✓ 15.2	✓ 95	✓ 100	✓ 67	✓ 15.2	✓ 17.0	✓ 95	✓ 15.2

Tip: you can click/tap on a cell for more information.

✓ Full support

webassembly.extended-constant-expressions

Report problems with this compatibility data on GitHub

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android	WebView on iOS
`extended-constant-expressions`	✓ 114	✓ 114	✓ 112	✓ 100	✓ 17.4	✓ 114	✓ 112	✓ 76	✓ 17.4	✓ 23.0	✓ 114	✓ 17.4

Tip: you can click/tap on a cell for more information.

✓ Full support ⊗ No support

webassembly.fixed-width-SIMD

Report problems with this compatibility data on GitHub

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android	WebView on iOS
`fixed-width-SIMD`	✓ 91	✓ 91	✓ 89	✓ 77	✓ 16.4	✓ 91	✓ 89	✓ 64	✓ 16.4	✓ 16.0	✓ 91	✓ 16.4

Tip: you can click/tap on a cell for more information.

✓ Full support

webassembly.garbage-collection

Report problems with this compatibility data on GitHub

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android	WebView on iOS
`garbage-collection`	✓ 119	✓ 119	✓ 120	✓ 105	18.2	✓ 119	✓ 120	✓ 79	18.2	✓ 25.0	✓ 119	18.2

Tip: you can click/tap on a cell for more information.

✓ Full support ⚗ In development. Supported in a pre-release version. ⊗ No support

webassembly.multiMemory

Report problems with this compatibility data on GitHub

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android	WebView on iOS
`multiMemory`	✓ 120	✓ 120	✓ 125	✓ 106	✗ No	✓ 120	✓ 125	✓ 80	✗ No	✓ 25.0	✓ 120	✗ No

Tip: you can click/tap on a cell for more information.

✓ Full support ✗ No support

webassembly.multi-value

Report problems with this compatibility data on GitHub

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android	WebView on iOS
`multi-value`	✓ 86	✓ 86	✓ 78	✓ 72	✓ 13.1	✓ 86	✓ 79	✓ 61	✓ 13.4	✓ 14.0	✓ 86	✓ 13.4

Tip: you can click/tap on a cell for more information.

✓ Full support

webassembly.mutable-globals

Report problems with this compatibility data on GitHub

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android	WebView on iOS
`mutable-globals`	✓ 69	✓ 80	✓ 62	✓ 56	✓ 13.1	✓ 69	✓ 62	✓ 48	✓ 13.4	✓ 10.0	✓ 69	✓ 13.4

Tip: you can click/tap on a cell for more information.

✓ Full support

webassembly.non-trapping-float-to-int-conversions

Report problems with this compatibility data on GitHub

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android	WebView on iOS
`non-trapping-float-to-int-conversions`	✓ 75	✓ 18	✓ 64	✓ 62	✓ 15.1	✓ 75	✓ 64	✓ 54	✓ 15.1	✓ 11.0	✓ 75	✓ 15.1

Tip: you can click/tap on a cell for more information.

✓ Full support

webassembly.reference-types

Report problems with this compatibility data on GitHub

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android	WebView on iOS
`reference-types`	✓ 96	✓ 96	✓ 79	✓ 82	✓ 15.1	✓ 96	✓ 79	✓ 67	✓ 15.1	✓ 17.0	✓ 96	✓ 15.1

Tip: you can click/tap on a cell for more information.

✓ Full support

webassembly.sign-extension-operations

Report problems with this compatibility data on GitHub

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android	WebView on iOS
`sign-extension-operations`	✓ 69	✓ 18	✓ 62	✓ 56	✓ 14.1	✓ 69	✓ 62	✓ 48	✓ 14.5	✓ 10.0	✓ 69	✓ 14.5

Tip: you can click/tap on a cell for more information.

✓ Full support

webassembly.tail-calls

Report problems with this compatibility data on GitHub

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android	WebView on iOS
`tail-calls`	✓ 112	✓ 112	✓ 121	✓ 98	18.2	✓ 112	✓ 121	✓ 75	18.2	✓ 23.0	✓ 112	18.2

Tip: you can click/tap on a cell for more information.

✓ Full support In development. Supported in a pre-release version. No support

webassembly.threads-and-atomics

Report problems with this compatibility data on GitHub

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android	WebView on iOS
`threads-and-atomics`	✓ 74	✓ 79	✓ 79	✓ 62	✓ 15.2	✓ 88	✓ 79	✓ 63	✓ 15.2	✓ 15.0	No	✓ 15.2

Tip: you can click/tap on a cell for more information.

✓ Full support No support

See also

- WebAssembly on Mozilla Research
- webassembly.org
- WebAssembly articles on Mozilla Hacks blog
- W3C WebAssembly Community Group
- Emscripting a C Library to Wasm