

このページはコミュニティの尽力で英語から翻訳されました。MDN Web Docs コミュニティについてもっと知り、仲間になるには [こちらから](#)。

# WebAssembly

WebAssembly は現代のウェブブラウザで実行できる新しい種類のコードです。ネイティブに近いパフォーマンスで動作する、コンパクトなバイナリー形式の低レベルなアセンブリ風言語です。さらに、C/C++、C# や Rust などの言語のコンパイル先となり、それらの言語をウェブ上で実行することができます。WebAssembly は JavaScript と並行して動作するように設計されているため、両方を連携させることができます。

## 概要

WebAssembly はウェブプラットフォームに大きな影響を与えます。以前ではできなかったようなウェブ上で動作するクライアントアプリケーションのために、複数の言語で記述されたコードをウェブ上でネイティブに近い速度で実行する方法を提供します。

WebAssembly は JavaScript を補完、並行して動作するように設計されています。WebAssembly JavaScript API を使用して、WebAssembly モジュールを JavaScript アプリケーションに読み込み、2 つの間で機能を共有できます。これにより、WebAssembly コードの記述方法を知らなくても、WebAssembly のパフォーマンスとパワー、JavaScript の表現力と柔軟性を同じアプリケーションで活用できます。

さらに、[W3C WebAssembly Working Group](#) と [コミュニティグループ](#) を介して、ウェブ標準として開発されており、主要なブラウザベンダーも積極的に参加しています。

## ガイド

### [WebAssembly の概要](#)

まずは WebAssembly の大まかな概念 — WebAssembly とはなにか、有用性、ウェブプラットフォーム (またはそれ以上) にどのように適合するか、どのように使用するか — の理解から始めてください。

### [C/C++ から WebAssembly へのコンパイル](#)

C/C++ で書いたコードを [Emscripten](#) のようなツールを使って .wasm にコンパイルできます。どのように動作するか確認してみましょう。

### [既存の C モジュールから WebAssembly へのコンパイル](#)

WebAssembly のコアユースケースは、既存の C ライブラリーのエコシステムを利用し、開発者がウェブ上でそれらを使用できるようにすることです。

### [Rust から WebAssembly へのコンパイル](#)

Rust コードを書いているだけで、WebAssembly にコンパイルすることができます。このチュートリアルでは、Rust プロジェクトをコンパイルして既存のウェブアプリケーションで使用するために知っておく必要があるすべてのことを説明します。

### [WebAssembly コードの読み込みと実行](#)

.wasm を手に入れたら、この記事ではそれを読み取り、コンパイルし、インスタンス化する方法を説明します。 [WebAssembly JavaScript API](#) と [Fetch](#) または [XHR](#) API とを組み合わせます。

### [WebAssembly JavaScript API の使用](#)

.wasm モジュールを読み込んだら、それを使いたくなるでしょう。この記事では、WebAssembly JavaScript API を用いて WebAssembly を使用する方法を説明します。

## [エクスポートされた WebAssembly 関数](#)

エクスポートされた WebAssembly 関数は、WebAssembly 関数の JavaScript リフレクションであり、JavaScript から WebAssembly コードを呼び出すことができます。この記事では、それらが何なのか説明します。

## [WebAssembly テキスト形式を理解する](#)

この記事では wasm テキスト形式について説明します。これは .wasm モジュールの低レベルなテキスト表現で、デバッグ時にブラウザーの開発者ツールに表示されます。

## [WebAssembly テキスト形式から wasm に変換する](#)

この記事では、テキスト形式で書かれた WebAssembly モジュールを .wasm バイナリーに変換する方法について説明します。

# API リファレンス

## [WebAssembly の制御フロー \(en-US\)](#)

WebAssembly の制御フロー操作の設定に関するリファレンスドキュメントです。

## [WebAssembly](#)

このオブジェクトは、WebAssembly に関連する全ての機能の名前空間として振る舞います。

## [WebAssembly.Global\(\)](#)

WebAssembly.Global オブジェクトは JavaScript と 1 つ以上の [WebAssembly.Module](#) インスタンス（のインポート/エクスポート可能な値）を横断してアクセスできるグローバル変数のインスタンスを表現します。これによって、複数モジュールでの動的リンクを実現できます。

## [WebAssembly.Module\(\)](#)

WebAssembly.Module オブジェクトにはブラウザーでコンパイルされたステートレスな WebAssembly コードが含まれており、効率的に[ワーカースレッドで共有](#)したり、複数回インスタンス化したりすることができます。

## [WebAssembly.Instance\(\)](#)

WebAssembly.Instance オブジェクトはステートフルで、実行可能な Module のインスタンスです。Instance オブジェクトには JavaScript から WebAssembly コードを呼び出すことを許可された[エクスポートされた WebAssembly 関数](#)が含まれます。

## [WebAssembly.instantiateStreaming\(\)](#)

WebAssembly.instantiate() 関数は WebAssembly コードをコンパイル、インスタンス化するための主要な API で、Module と、その最初の Instance を返します。

## [WebAssembly.Memory\(\)](#)

WebAssembly.Memory オブジェクトはリサイズ可能な [ArrayBuffer](#) で、Instance からアクセスされる生のバイト列を保持します。

## [WebAssembly.Table\(\)](#)

WebAssembly.Table オブジェクトは Instance からアクセスされる関数参照などの不透明値のリサイズ可能な型付き配列です。

## [WebAssembly.CompileError\(\)](#)

WebAssembly CompileError オブジェクトを生成します。

## [WebAssembly.LinkError\(\)](#)

WebAssembly LinkError オブジェクトを生成します。

## [WebAssembly.RuntimeError\(\)](#)

WebAssembly RuntimeError オブジェクトを生成します。

# 例

- [WASMSobel](#)
- 他の多くの例については [webassembly-examples](#) リポジトリを参照してください。

# 仕様書

Specification
<a href="#">WebAssembly JavaScript Interface</a> <a href="#"># webassembly-namespace</a>

# ブラウザーの互換性

[Report problems with this compatibility data on GitHub](#)

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android
<a href="#">WebAssembly</a>	Chrome 57	Edge 16	Firefox 52	Opera 44	Safari 11	Chrome 57 Android
<a href="#">CompileError</a>	Chrome 57	Edge 16	Firefox 52	Opera 44	Safari 11	Chrome 57 Android
<a href="#">CompileError().constructor</a>	Chrome 57	Edge 16	Firefox 52	Opera 44	Safari 11	Chrome 57 Android
<a href="#">Exception</a>	Chrome 95	Edge 95	Firefox 100	Opera 81	Safari 15.2	Chrome 95 Android
<a href="#">Exception().constructor</a>	Chrome 95	Edge 95	Firefox 100	Opera 81	Safari 15.2	Chrome 95 Android
<a href="#">options.traceStack parameter</a>	Chrome 95	Edge 95	Firefox 100	Opera 81	Safari 15.2	Chrome 95 Android
<a href="#">Exception.getArg</a>	Chrome 95	Edge 95	Firefox 100	Opera 81	Safari 15.2	Chrome 95 Android
<a href="#">Exception.is</a>	Chrome 95	Edge 95	Firefox 100	Opera 81	Safari 15.2	Chrome 95 Android
<a href="#">Stack trace</a>	Chrome 95	Edge 95	Firefox 100	Opera 81	Safari 15.2	Chrome 95 Android
<a href="#">Global</a>	Chrome 69	Edge 79	Firefox 62	Opera 56	Safari 13.1	Chrome 69 Android

<a href="#">Global().constructor</a>	Chrome 69	Edge 79	Firefox 62	Opera 56	Safari 13.1	Chrome 69 Android	
<a href="#">Global.value</a>	Chrome 69	Edge 79	Firefox 62	Opera 56	Safari 13.1	Chrome 69 Android	
<a href="#">Global.valueOf</a>	Chrome 69	Edge 79	Firefox 62	Opera 56	Safari 13.1	Chrome 69 Android	
<a href="#">Instance</a>	Chrome 57	Edge 16	Firefox 52	Opera 44	Safari 11	Chrome 57 Android	
<a href="#">Instance().constructor</a>	Chrome 57	Edge 16	Firefox 52	Opera 44	Safari 11	Chrome 57 Android	
<a href="#">Instance.exports</a>	Chrome 57	Edge 16	Firefox 52	Opera 44	Safari 11	Chrome 57 Android	
<a href="#">LinkError</a>	Chrome 57	Edge 16	Firefox 52	Opera 44	Safari 11	Chrome 57 Android	
<a href="#">LinkError().constructor</a>	Chrome 57	Edge 16	Firefox 52	Opera 44	Safari 11	Chrome 57 Android	
<a href="#">Memory</a>	Chrome 57	Edge 16	Firefox 52	Opera 44	Safari 11	Chrome 57 Android	
<a href="#">Memory().constructor</a>	Chrome 57	Edge 16	Firefox 52	Opera 44	Safari 11	Chrome 57 Android	
shared flag	Chrome 74	Edge 79	Firefox 78	Opera 62	Safari 15.2	Chrome 88 Android	
<a href="#">Memory.buffer</a>	Chrome 57	Edge 16	Firefox 52	Opera 44	Safari 11	Chrome 57 Android	
<a href="#">Memory.grow</a>	Chrome 57	Edge 16	Firefox 52	Opera 44	Safari 11	Chrome 57 Android	
<a href="#">Module</a>	Chrome 57	Edge 16	Firefox 52	Opera 44	Safari 11	Chrome 57 Android	
<a href="#">Module().constructor</a>	Chrome 57	Edge 16	Firefox 52	Opera 44	Safari 11	Chrome 57 Android	
<a href="#">Module.customSections</a>	Chrome 57	Edge 16	Firefox 52	Opera 44	Safari 11	Chrome 57 Android	
<a href="#">Module.exports</a>	Chrome 57	Edge 16	Firefox 52	Opera 44	Safari 11	Chrome 57 Android	

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	
<a href="#">Module.imports</a>	Chrome 57	Edge 16	Firefox 52	Opera 44	Safari 11	Chrome 57 Android	
<a href="#">RuntimeError</a>	Chrome 57	Edge 16	Firefox 52	Opera 44	Safari 11	Chrome 57 Android	
<a href="#">RuntimeError(). constructor</a>	Chrome 57	Edge 16	Firefox 52	Opera 44	Safari 11	Chrome 57 Android	
<a href="#">Table</a>	Chrome 57	Edge 16	Firefox 52	Opera 44	Safari 11	Chrome 57 Android	
<a href="#">Table(). _constructor</a>	Chrome 57	Edge 16	Firefox 52	Opera 44	Safari 11	Chrome 57 Android	
<a href="#">Table.get</a>	Chrome 57	Edge 16	Firefox 52	Opera 44	Safari 11	Chrome 57 Android	
<a href="#">Table.grow</a>	Chrome 57	Edge 16	Firefox 52	Opera 44	Safari 11	Chrome 57 Android	
<a href="#">Table.length</a>	Chrome 57	Edge 16	Firefox 52	Opera 44	Safari 11	Chrome 57 Android	
<a href="#">Table.set</a>	Chrome 57	Edge 16	Firefox 52	Opera 44	Safari 11	Chrome 57 Android	
<a href="#">Tag</a>	Chrome 95	Edge 95	Firefox 100	Opera 81	Safari 15.2	Chrome 95 Android	
<a href="#">Tag(). _constructor</a>	Chrome 95	Edge 95	Firefox 100	Opera 81	Safari 15.2	Chrome 95 Android	
<a href="#">Tag.type</a>	Chrome No	Edge No	Firefox No	Opera No	Safari 15.2	Chrome No Android	
<a href="#">compile</a>	Chrome 57	Edge 16	Firefox 52	Opera 44	Safari 11	Chrome 57 Android	
<a href="#">compileStreaming</a>	Chrome 61	Edge 16	Firefox 58	Opera 47	Safari 15	Chrome 61 Android	
<a href="#">instantiate</a>	Chrome 57	Edge 16	Firefox 52	Opera 44	Safari 11	Chrome 57 Android	

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android
<a href="#">instantiateStreaming</a>	Chrome 61	Edge 16	Firefox 58	Opera 47	Safari 15	Chrome 61 Android
<a href="#">validate</a>	Chrome 57	Edge 16	Firefox 52	Opera 44	Safari 11	Chrome 57 Android

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.

関連情報

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

This page was last modified on 2022年9月28日 by [MDN contributors](#).