WebAssembly PHP

□□

- Networking support varies between platforms

WordPress Playground build the PHP interpreter to WebAssembly using Emscripten and a dedicated pipeline. Building C programs to WebAssembly Building PHP to WebAssembly is very similar to building vanilla PHP. The wasm build required adjusting a function signature here, forcing a config variable there, and applying a few small patches, but there's relatively few adjustments involved. Building PHP to WebAssembly However, vanilla PHP builds aren't very useful in the browser. As a server software, PHP doesn't have a JavaScript API to pass the request body, upload files, or populate the `php://stdin` stream. WordPress Playground had to build one from scratch. The WebAssembly binary comes with a dedicated PHP API module written in C and a JavaScript PHP class that exposes methods like writeFile() or run(). Because every PHP version is just a static .wasm file, the PHP version switcher is actually pretty boring. It simply tells the browser to download, for example, `php_7_3.wasm` instead of, say, `php_8_2.wasm`.
Building different versions of PHP to WebAssembly

Networking support varies between platforms

When it comes to networking, WebAssembly programs are limited to calling JavaScript APIs. It is a safety feature, but also presents a challenge. How do you support low-level, synchronous networking code used by PHP with the high-level asynchronous APIs available in JavaScript? In Node.js, the answer involves a WebSocket to TCP socket proxy, Asyncify, and patching deep PHP internals like php_select. It's complex, but there's a reward. The Node.js-targeted PHP build can request web APIs, install composer packages, and even connect to a MySQL server. In the browser, networking is supported in two ways:

- A fast transport for `wp_safe_remote_get` to translate them into `fetch()` calls.
- A slower transport for all other network calls that parses the TLS transmission initiated by PHP and translates it to a `fetch()` call.

□□□ #2
□ □□□ □□□ 2026-08-29 15:36:35 UTC
□ □□□ □□□ 2026-08-29 15:36:35 UTC