

Deno



Basics

What is Deno?

Deno is a simple, modern and secure runtime for JavaScript and TypeScript that uses V8 and is built in Rust.

- Secure by default. No file, network, or environment access, unless explicitly enabled.
- Supports TypeScript out of the box.
- Ships only a single executable file.
- Has built-in utilities like a dependency inspector and a code formatter.
- Has a set of reviewed (audited) standard modules that are guaranteed to work with Deno: deno.land/std

Is Deno a Node.js killer?

Nope XD

REPL

read-eval-print loop:

```
> deno
```

*does not support TypeScript

Running scripts from the web

Deno allows you to run scripts directly from an url:

```
> deno run https://deno.land/std/examples/welcome.ts
```

Installing Deno scripts

Deno allows you to run scripts directly from an url:

```
> deno install --allow-net --allow-read  
    https://deno.land/std@0.74.0/http/file_server.ts
```

Features

- TS or JS
- Secure by Default
- De-centralized packages
- STD library
- Modern JS
- ES Modules
- Top Level Await
- Built in Testing
- Browser compatible API
- WASM!

Server in node vs Server in Deno

Demo Timeeee!

WebAssembly

Deno supports WebAssembly out-of-the-box and can execute WebAssembly binaries.

```
const wasmCode = new Uint8Array([
  0, 97, 115, 109, 1, 0, 0, 0, 1, 133, 128, 128, 128, 0, 1, 96, 0, 1, 127,
  3, 130, 128, 128, 128, 0, 1, 0, 4, 132, 128, 128, 128, 0, 1, 112, 0, 0,
  5, 131, 128, 128, 128, 0, 1, 0, 1, 6, 129, 128, 128, 128, 0, 0, 7, 145,
  128, 128, 128, 0, 2, 6, 109, 101, 109, 111, 114, 121, 2, 0, 4, 109, 97,
  105, 110, 0, 0, 10, 138, 128, 128, 128, 0, 1, 132, 128, 128, 128, 0, 0,
  65, 42, 11
]);
const wasmModule = new WebAssembly.Module(wasmCode);
const wasmInstance = new WebAssembly.Instance(wasmModule);
const main = wasmInstance.exports.main as CallableFunction
console.log(wasmInstance.exports.main().toString());
```

*Right now it is the best way to use native modules in Deno.

Q&A



Thank you!