



Deno: the JavaScript Runtime for the Serverless Era

Ryan Dahl
2022-05-25, GOTO Conf

The web has become the medium of human information.

The web will still be here 5 years from now. If not 10 or 20.

JavaScript is **inherently tied** to the web.

Therefore,

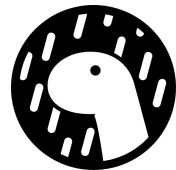
JavaScript will still be important many years from now.



Because of this,
JavaScript has become the default scripting language.

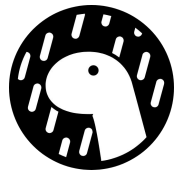
If not the default language...

Yet so much of the tooling and libraries around JavaScript remains hopelessly sophomoric.



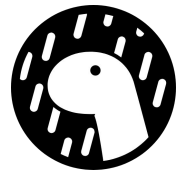
My Dream Stack

- Reduces boilerplate - ideally very small apps can be defined in a single file
- Uses JavaScript, the universal scripting language
- Async I/O and optimal HTTP server performance
- Built-in, uniform dev tools - code formatter, linter, doc generation, etc.



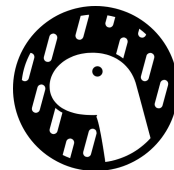
My dream stack extends to the cloud

- Serverless - managed, pay per request, free for low-traffic/demo apps
- Edge - code runs geographically near users. Excellent latency everywhere.
- Cold starts must be fast.
- No config. Reduce boilerplate!



Deno is JavaScript for professional software engineers.

- Simple
- Secure
- Optimal



Simple

Built-in JavaScript type annotations (TypeScript).

Common tooling built-in (code formatter, linter, LSP, testing)

Standard Library

Third party modules via HTTPS or NPM.

deno compile packages it all up into an executable.

Browser knowledge transfers easily, because Deno adheres to browser standard APIs.

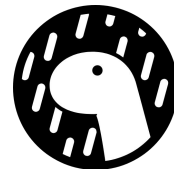
```
import express from "npm:express";

const app = express();

app.get("/", function (req, res) {
  res.send("Hello World");
});

app.listen(3000);

console.log("listening http://localhost:3000/");
```



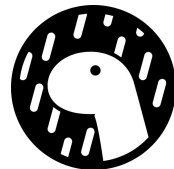
Secure

By default Deno is a secure sandbox (no access to local file system nor to network)

Unlike in Node.js, you do not need to allow remote code execution to install a module.

Written in Rust instead of C++. This avoids whole classes of memory safety bugs.

```
deno
~# deno run https://deno.land/std/http/file_server.ts
⚠ Deno requests read access to <CWD>.
  Requested by `Deno.cwd()` API
  Run again with --allow-read to bypass this prompt.
  Allow? [y/n] (y = yes, allow; n = no, deny) > 
```



Optimal

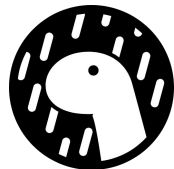
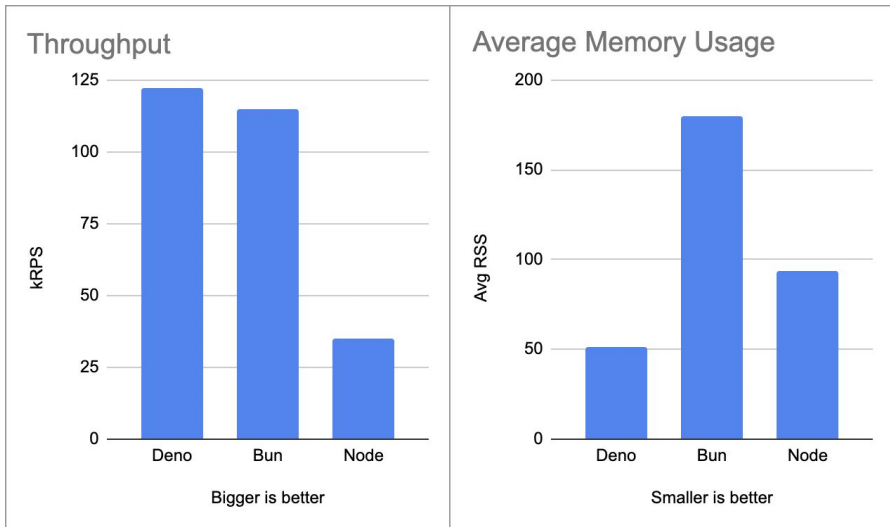
We strive to make Deno optimal in every benchmark.

It helps that Deno is built on the shoulders of giants: **V8** and **Rust**.

We're working on a comprehensive benchmark suite. Watch the Deno blog in the coming weeks!

A small preview: hello-world web server performance in throughput and memory usage.

Linux x64, wrk2 1024 connections,
deno-1.26.0 using Deno.serve, bun-0.1.13,
node-18.7.0



Web standard APIs

- globalThis
- webAssembly
- crypto.subtle
- Worker
- TransformStream
- EventTarget
- AbortController
- location
- FormData
- Request and Response
- window
- window.close()
- localStorage
- navigator.gpu



WinterCG

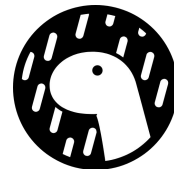
Web-interoperable Runtimes Community Group

[Work](#)[FAQ](#)[GitHub](#)[Charter](#)

This community group aims to provide a space for JavaScript runtimes to collaborate on API interoperability. We focus on documenting and improving interoperability of web platform APIs across runtimes (especially non-browser ones). This is done through discussions among runtimes, proposals in specification venues (WHATWG, W3C) for new web APIs and for changes to current web APIs, and documentation of existing runtime behaviours. [Learn more.](#)



The work of the WinterCG is supported by:



Deno Deploy: instant *globally*

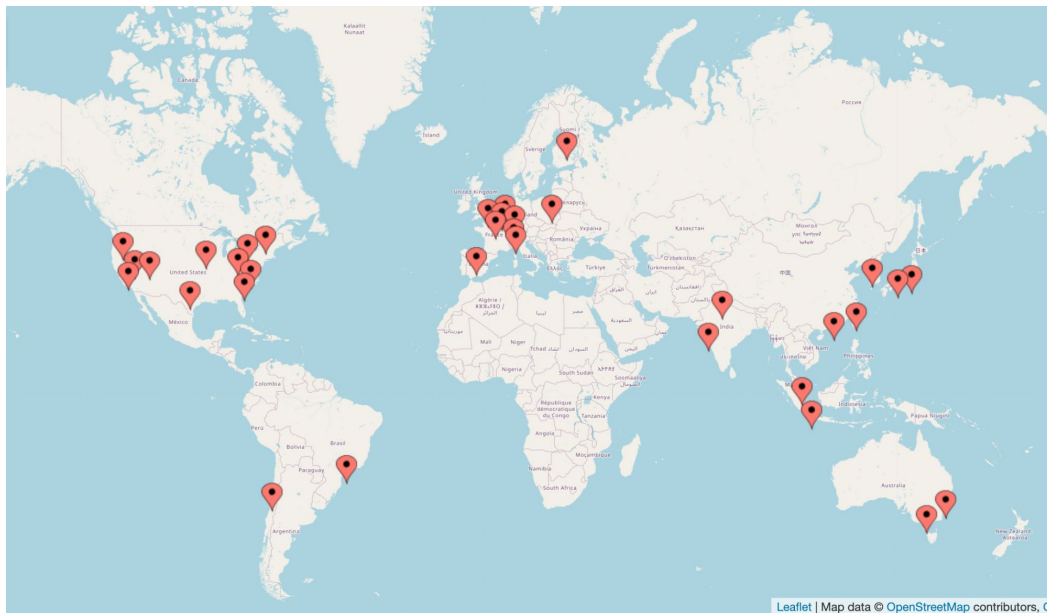
Our serverless edge platform

- 34 data centers
- GitHub integration
- Instant deploys

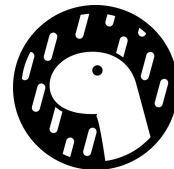
Within 100ms of majority of internet users (with more regions on the way)

<https://deno.com/deploy>

Deno Deploy powers Netlify Edge Functions

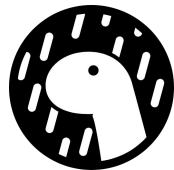


Leaflet | Map data © OpenStreetMap contributors,



The Post-Unix Future

- Serverless at Edge JavaScript runtimes like **Cloudflare Workers** and **Deno Deploy** are very cheap and very fast.
- Performance and simplicity is achieved by using V8 Isolates for multi-tenancy rather than Linux VMs
- **Prediction:** to maximize the utility of serverless, web frameworks will soon be built on post-Unix primitives rather than Node.js primitives.



Putting it all together: **Fresh**

A next-generation web framework, **built to be fast.**

<https://fresh.deno.dev/>

- Render everything JIT, on the server
- No JS shipped to client by default
- Island based client interactivity
- Automatic inline CSS with Tailwind plugin



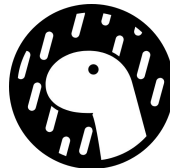
No build step.



**No configuration
necessary.**



TypeScript support out of
the box.



Demo



Bootstrap a new project:

```
deno run -A -r https://fresh.deno.dev  
my-project
```

<https://fresh.deno.dev/>

Enter the created directory and run:

```
deno task start
```

Go to <http://localhost:8000> to view your project.



We're hiring!

<https://deno.com/jobs>

Questions?

ry@deno.com

