



Deno 2

Ryan Dahl

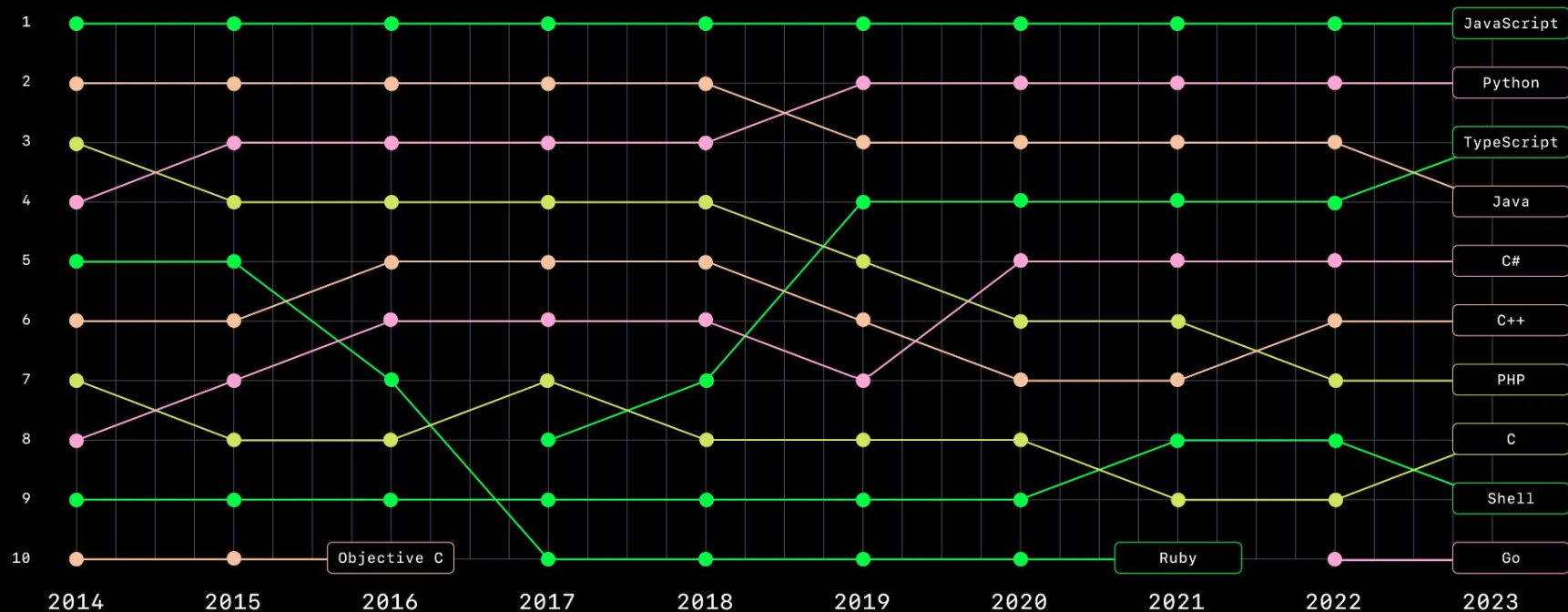
2024-10-22

GOTO Chicago

Deno is an open source
runtime for JavaScript and
TypeScript

Deno 2.0 was just released
two weeks ago!

Top 10 programming languages on GitHub



Deno

- ✓ Native TypeScript support
- ✓ Built on web standards
- ✓ All-in-one batteries included toolchain
- ✓ Secure by default
- ✓ Backwards compatible with Node.js



ls demo

We're going to re-implement the unix `ls` command that lists out all the files in a directory, in TypeScript.

No need for a directory filled with `package.json` and `node_modules`.

Just open a file and start coding.

```
import { walk } from "jsr:@std/fs";
import { relative } from "jsr:@std/path";

export async function tree(dir: string): string[] {
  const out = [];
  for await (const entry of walk(dir)) {
    const x = relative(dir, entry.path);
    out.push(x);
  }
  return out;
}

if (import.meta.main) {
  const x = await tree(Deno.args[0]);
  console.log("%c" + x.join("\n"), "color: yellow");
}
```

```
node file.js
```

```
deno file.js
```

```
node file.js  
deno file.js
```

```
tsc  
deno check
```

```
node file.js  
deno file.js
```

```
tsc  
deno check
```

```
npm install  
deno install
```

node file.js
deno file.js

tsc
deno check

npm install
deno install

node -e
deno eval

npm run
deno run

eslint
deno lint

prettier
deno fmt

package.json
deno.json

ts-node file.ts
deno file.ts

typedoc
deno doc

jest / ava / mocha
deno test

nodemon
deno run --watch

nexe / pkg
deno compile

npm explain
deno info

nvm / n / fnm
deno upgrade

tsserver
deno lsp

nyc / c8 / istanbul
deno coverage

benchmarks
deno bench

npm add
deno add

npm init
deno init

nexe / pkg
deno compile

Deno 2

But you could do all of that with Deno 1 already. Let's talk about what's new in Deno 2.

Deno 2 is a revamp in multiple areas

- Ecosystem compatibility,
- Package management, and
- Enterprise features

Ecosystem compatibility

Most NPM packages are supported now. (Even things complex modules like gRPC or those using N-API)

Deno understands package.json and node_modules

[Next demo]

Ecosystem compatibility

Deno programs can import NPM packages without requiring a package.json or node_modules folder.

Secure by default execution is maintained

[return to [ls](#) example, use chalk]

Ecosystem compatibility

- Run existing Node.js applications written in ESM with minimal changes in Deno.
- Deno understands `package.json` and `node_modules` out of the box.
- If you're not using `package.json` or `node_modules`, Deno allows you to use the full breadth of npm libraries in your deno code security with no configuration

Package management

deno install, deno add, deno remove

[deno install with create-next-app]

Package Management: JSR

Native TypeScript support

ECMAScript modules only

Designed to work alongside npm.

Explicitly built for all runtimes - Node / Deno / Bun
/ Cloudflare Workers / Browsers

Provides great DX for publishers and consumers



HTTP imports to NPM and JSR

```
import { assertEquals } from "https://deno.land/std@0.224.0/assert/mod.ts";  
  
assertEquals(1, 2);
```



```
import { assertEquals } from "jsr:@std/assert@1";  
  
assertEquals(1, 2);
```

Standard
library is
stabilized at
1.0 versions

jsr.io/@std

jsr.io/@std

Search for packages (35K)

Browse packages | Docs

@std

Packages Members Settings

- @std/archive**
Build and extract archives like .tar 94
- @std/assert**
Common assertion functions, especially useful for testing 94
- @std/async**
Utilities for asynchronous operations, like delays, debouncing, or pooling 94
- @std/bytes**
Utilities to manipulate Uint8Arrays that are not built-in to JavaScript 100
- @std/cli**
Tools for creating interactive command line tools 88
- @std/collections**
Pure functions for common tasks related to collection types like arrays and objects 100
- @std/console**
DEPRECATED: Use @std/cli/unicode-width instead 94
- @std/crypto**
Extensions to the Web Crypto API 94
- @std/csv**
Reading and writing of comma-separated values (CSV) files 100
- @std/data-structures**
Common data structures like red-black trees and binary heaps 94

Is demo

- use walk from std library
- publish to JSR

Enterprise Features

We've been talking to a lot of users at larger companies using Deno and the feedback we have received points to three major missing pieces:

- Private registries
- Workspaces and Monorepos
- Long Term Support

Workspaces and Monorepos

Deno has the ability to manage multiple packages in a monorepo

NPM workspaces are also supported and you're allowed to mix and match between deno.json packages and package.json packages

 deno.json

 deno.lock

 packages/

 add/

 package.json

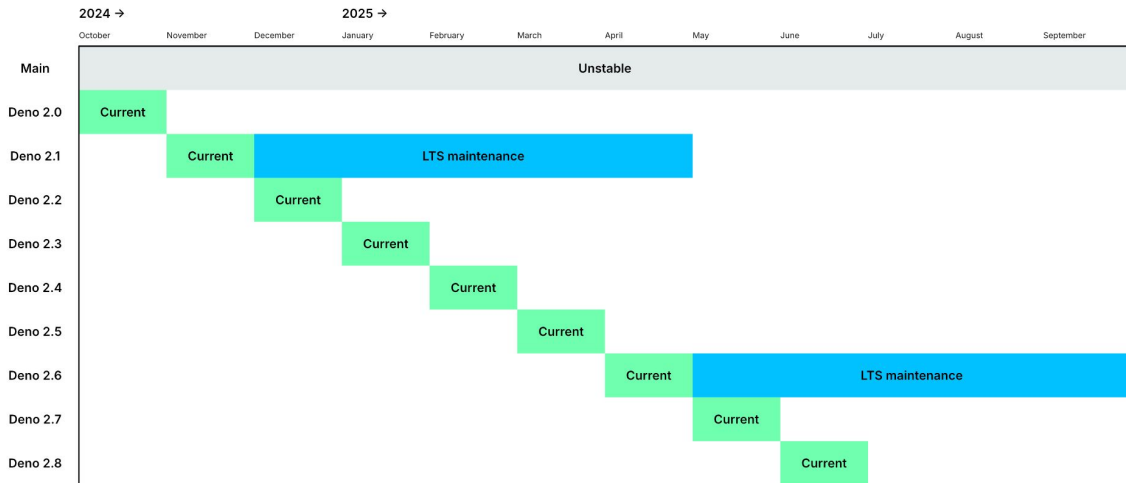
 index.js

 subtract/

 deno.json

 mod.ts

Long Term Support



- First LTS release in Deno 2.1
- Freely available, MIT license
- If you need it, we have enterprise level support: deno.com/enterprise

Much more!

WebGPU

Documentation generation: `deno doc --html`

Fast FFI

Fresh

Deno Deploy

Jupyter Notebooks

`[jupyter-notebook` demo if time]



deno.com/2