Dabbling with Deno

Tales from a web developer
playing with a new toy

Phil Hawksworth
he/him

Deno

Phil Hawksworth

he/him

Head of Developer Relations, Deno

Twitter

LinkedIn

GitHub

BlueSky

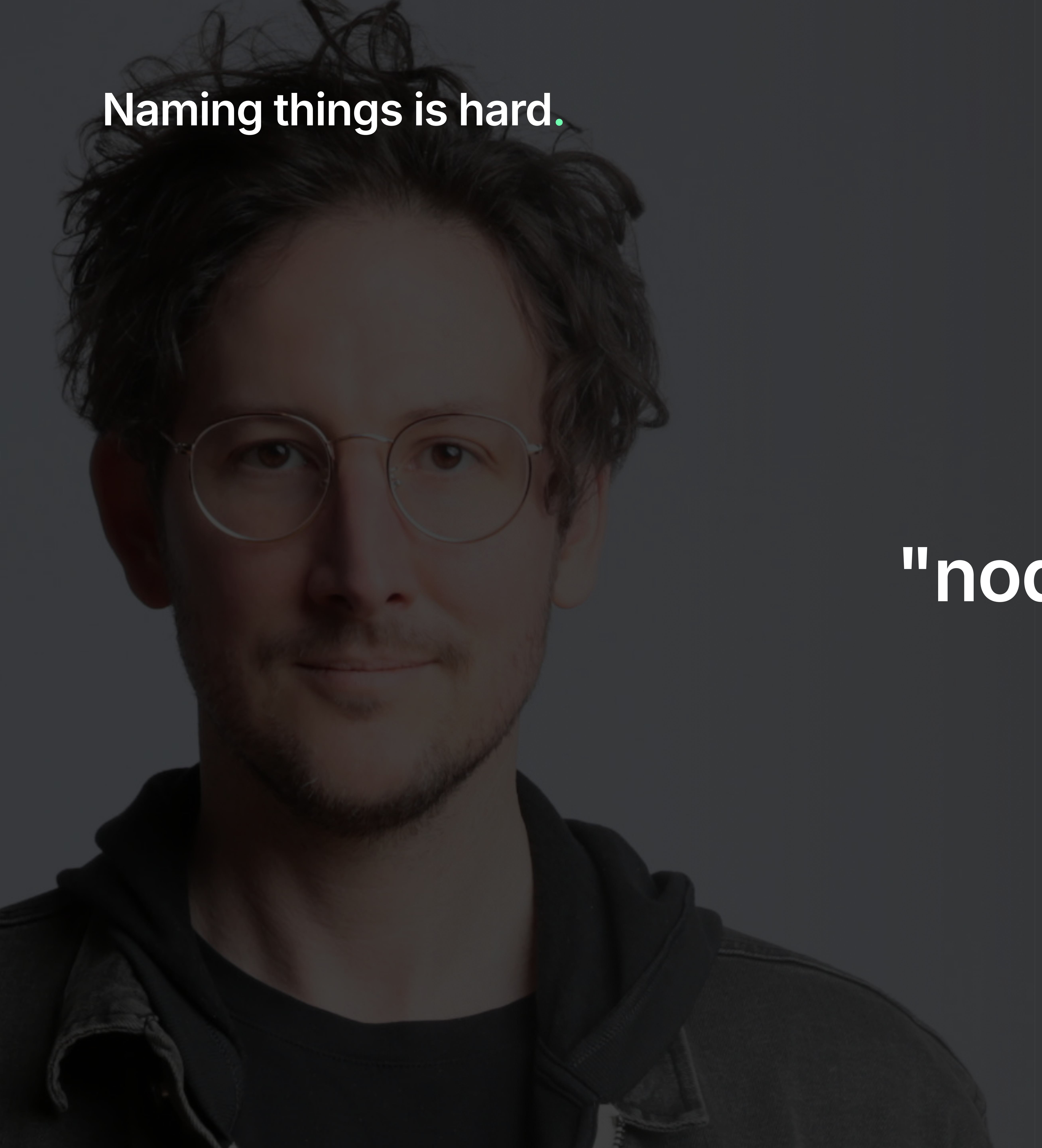
Web
hawksworx.com

@PhilHawksworth.dev

he/him

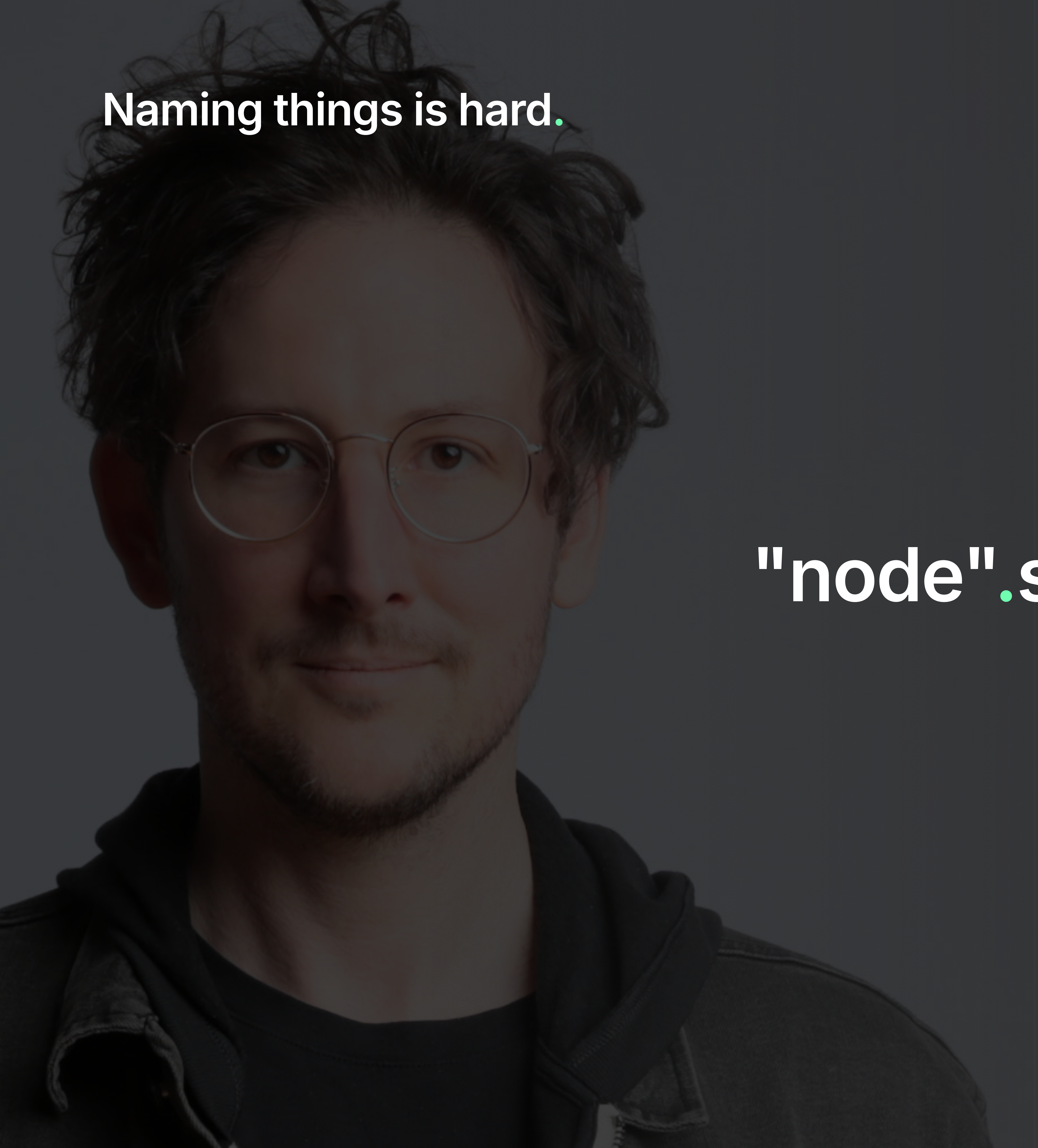
Head of Developer Relations, Deno

Naming things is hard.



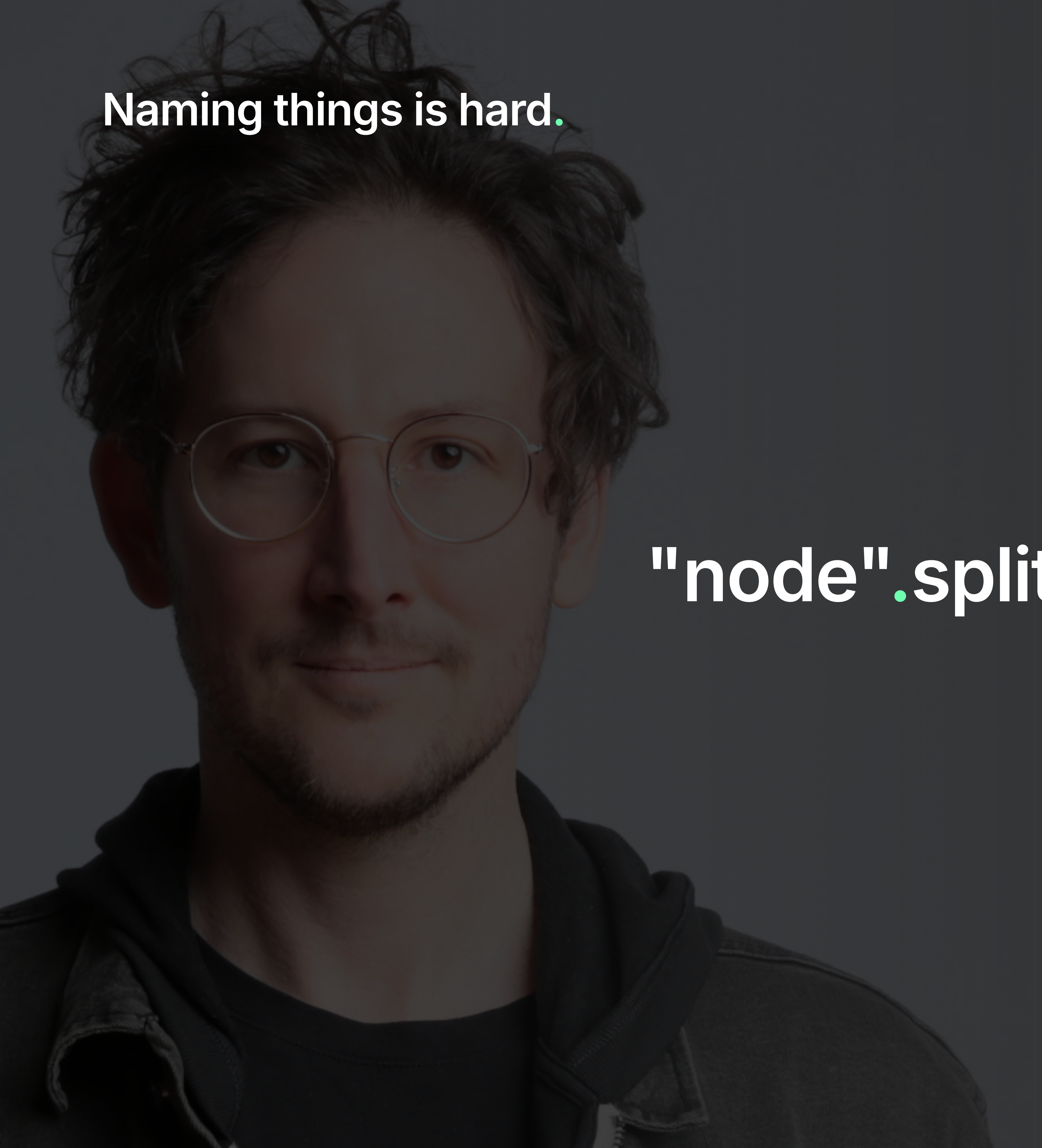
"node"

Naming things is hard.



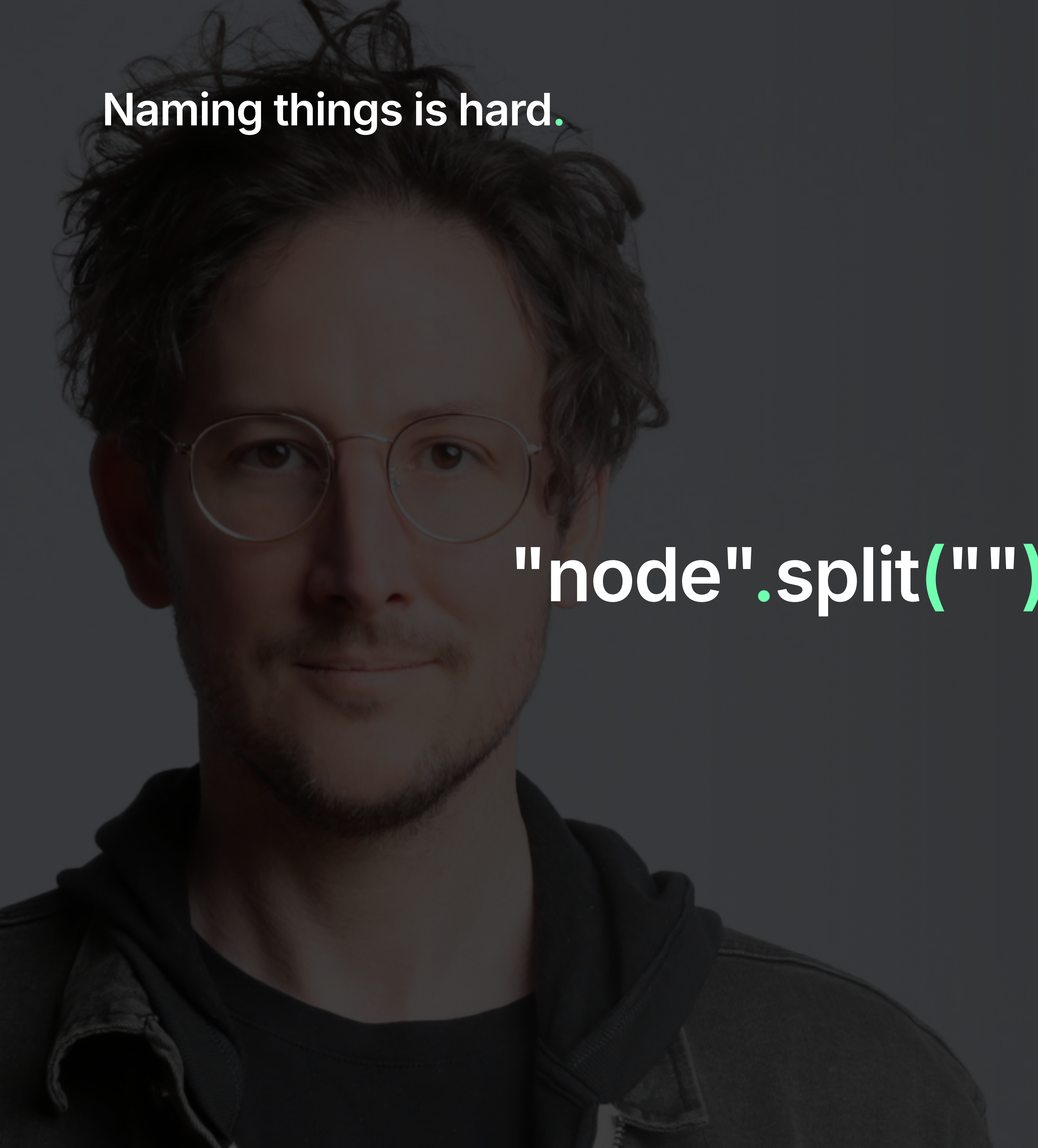
```
"node".split("")
```

Naming things is hard.



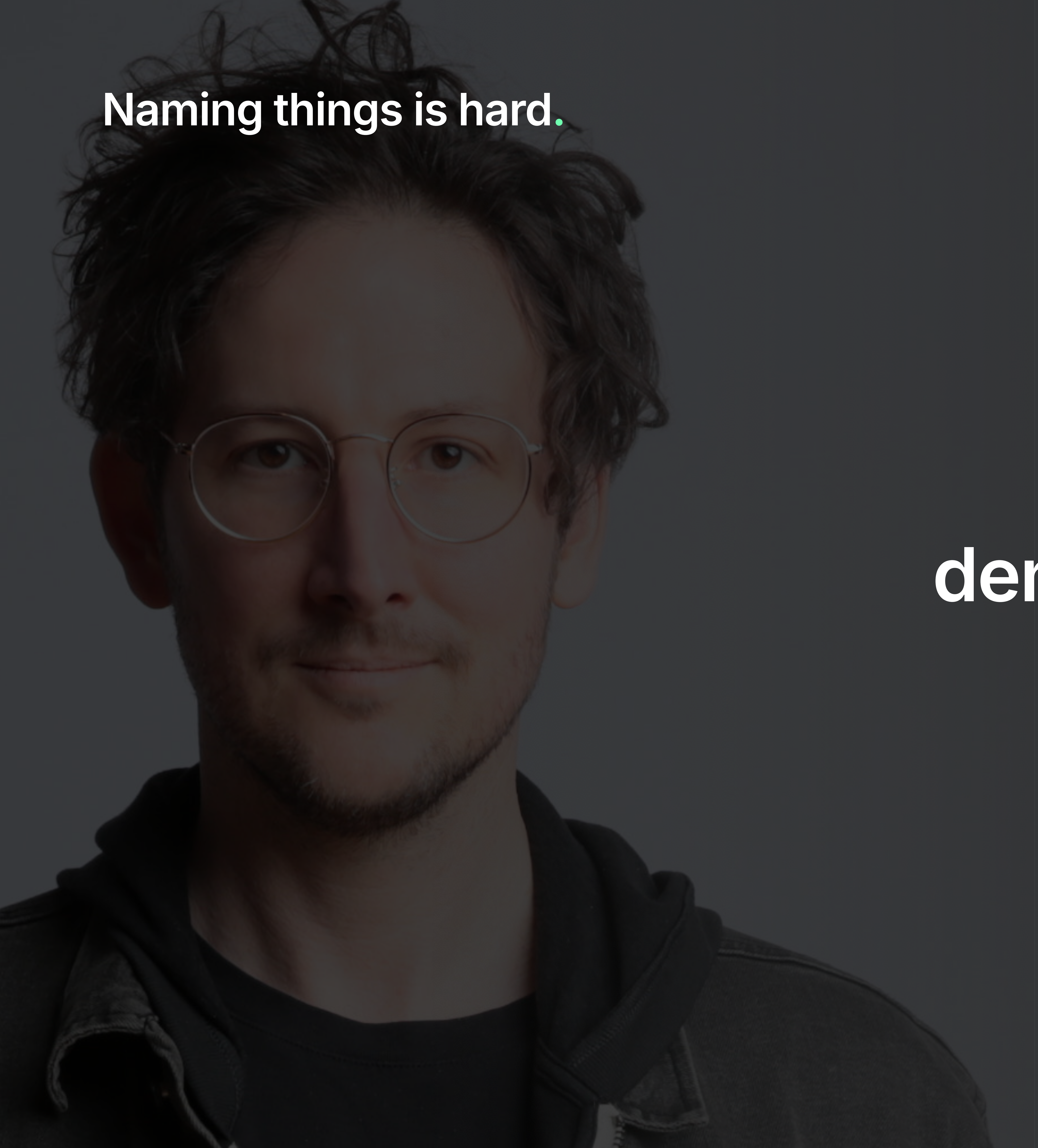
```
"node".split("").sort()
```

Naming things is hard.



```
"node".split("").sort().join("")
```

Naming things is hard.



deno

Naming things is hard.

ahkorswwx

Some technologies come and go

Some technologies stick around

Some technologies become ubiquitous

Tools and trends.

JavaScript™

Frontend / Backend / Edges

node

JavaScript™

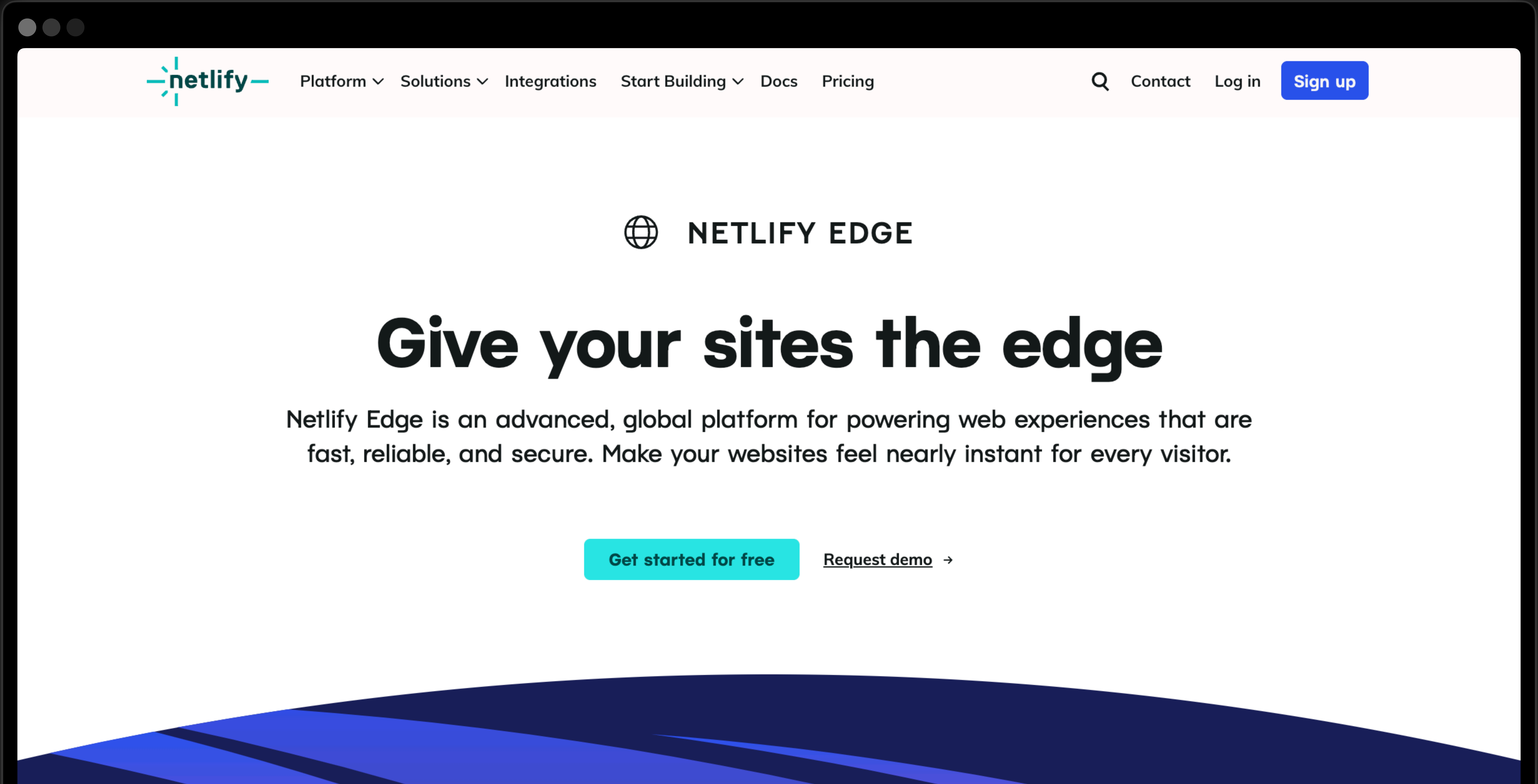
TypeScript

The pull of new interesting tools

I first tried Deno by accident

Deno first contact.

netlify.com



NETLIFY EDGE

Give your sites the edge

Netlify Edge is an advanced, global platform for powering web experiences that are fast, reliable, and secure. Make your websites feel nearly instant for every visitor.

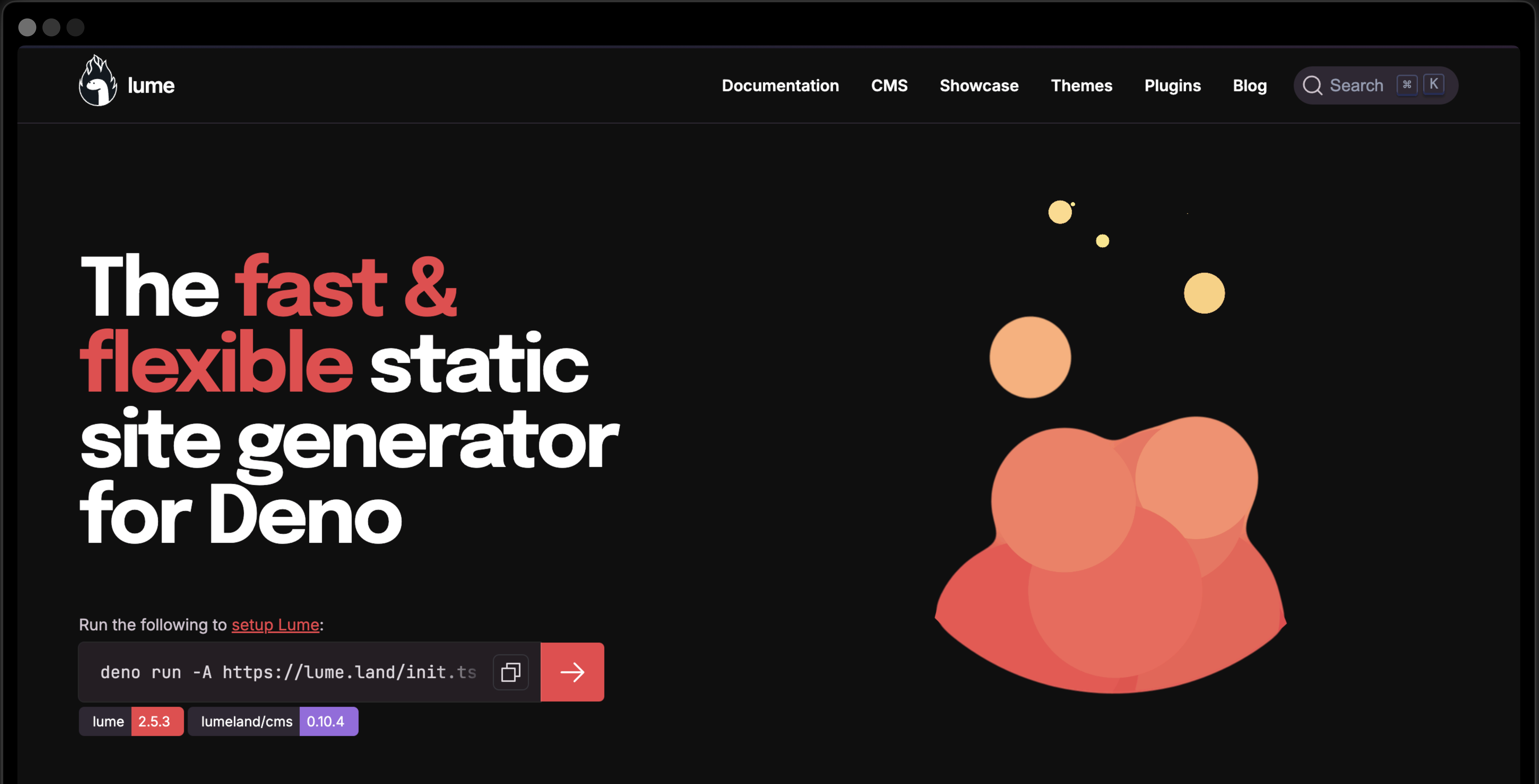
Deno first contact.

```
export default async () ⇒ {

  return new Response("Hello, World!", {
    headers: {
      "content-type": “text/html"
    }
  });

};
```

A gateway drug took me deeper

Deno first contact.

lume.land



The fast & flexible static site generator for Deno

Run the following to setup Lume:

```
deno run -A https://lume.land/init.ts
```

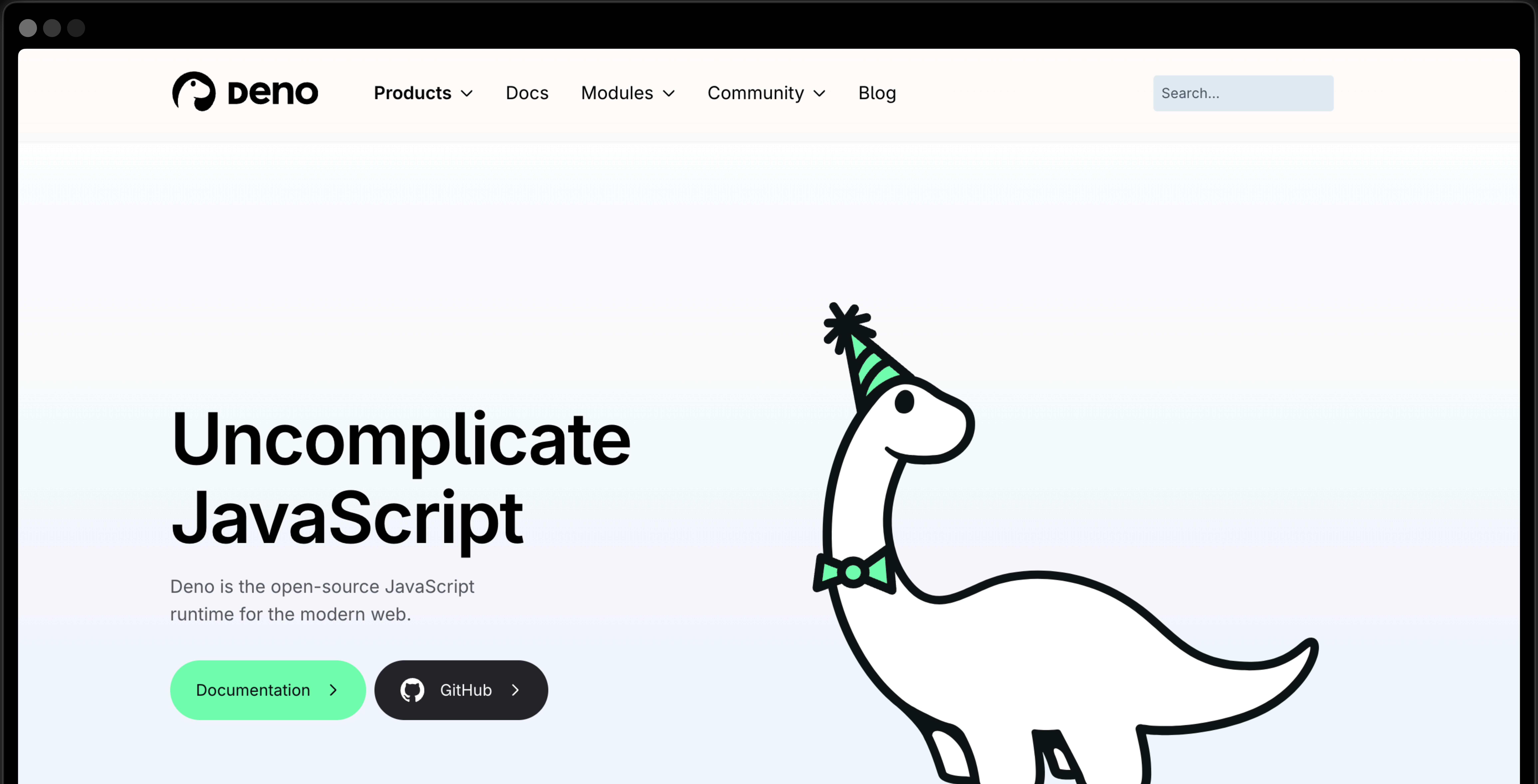
lume 2.5.3 lumeland/cms 0.10.4

Run the following to setup Lume:

```
deno run -A https://lume.land/init.ts
```

Deno first contact.

deno.com

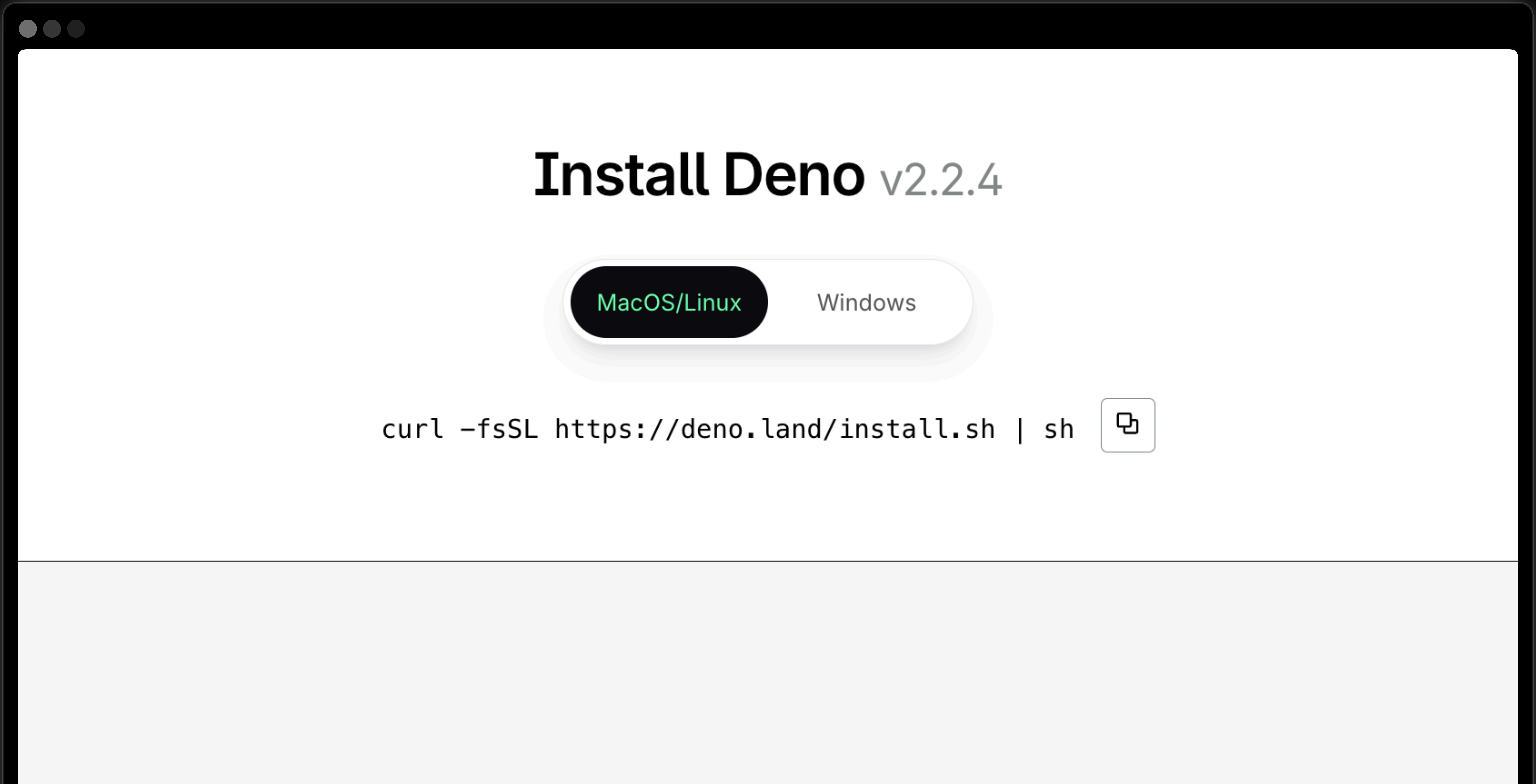


Uncomplicate JavaScript

Deno is the open-source JavaScript runtime for the modern web.

Deno first contact.

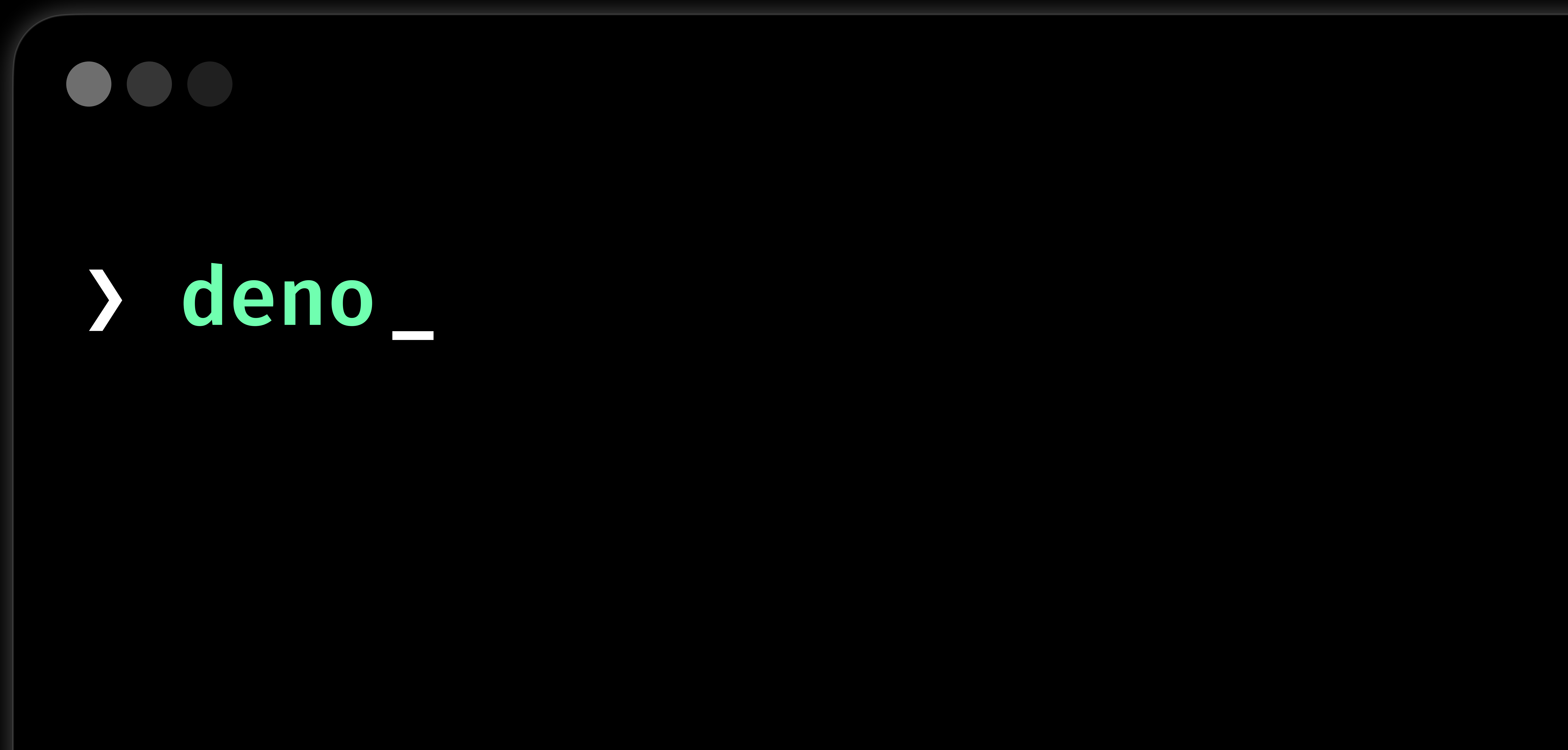
deno.com



Install Deno v2.2.4

```
curl -fsSL https://deno.land/install.sh | sh
```

Deno first contact.



```
❯ deno_
```

Um..what is Deno?

What is Deno?

1. Hosting platform
2. JavaScript runtime
3. TypeScript toolchain
4. Package manager
5. Task runner
6. Testing tool
7. Linter, formatter
8. Benchmarking tool
9. Compiler

What is Deno?

1. Hosting platform
2. JavaScript runtime
3. TypeScript toolchain
4. Package manager
5. Task runner
6. Testing tool
7. Linter, formatter
8. Benchmarking tool
9. Compiler

What got me intrigued?

What got me intrigued?

An API design based on web standards

What is Deno?

It's web standards.

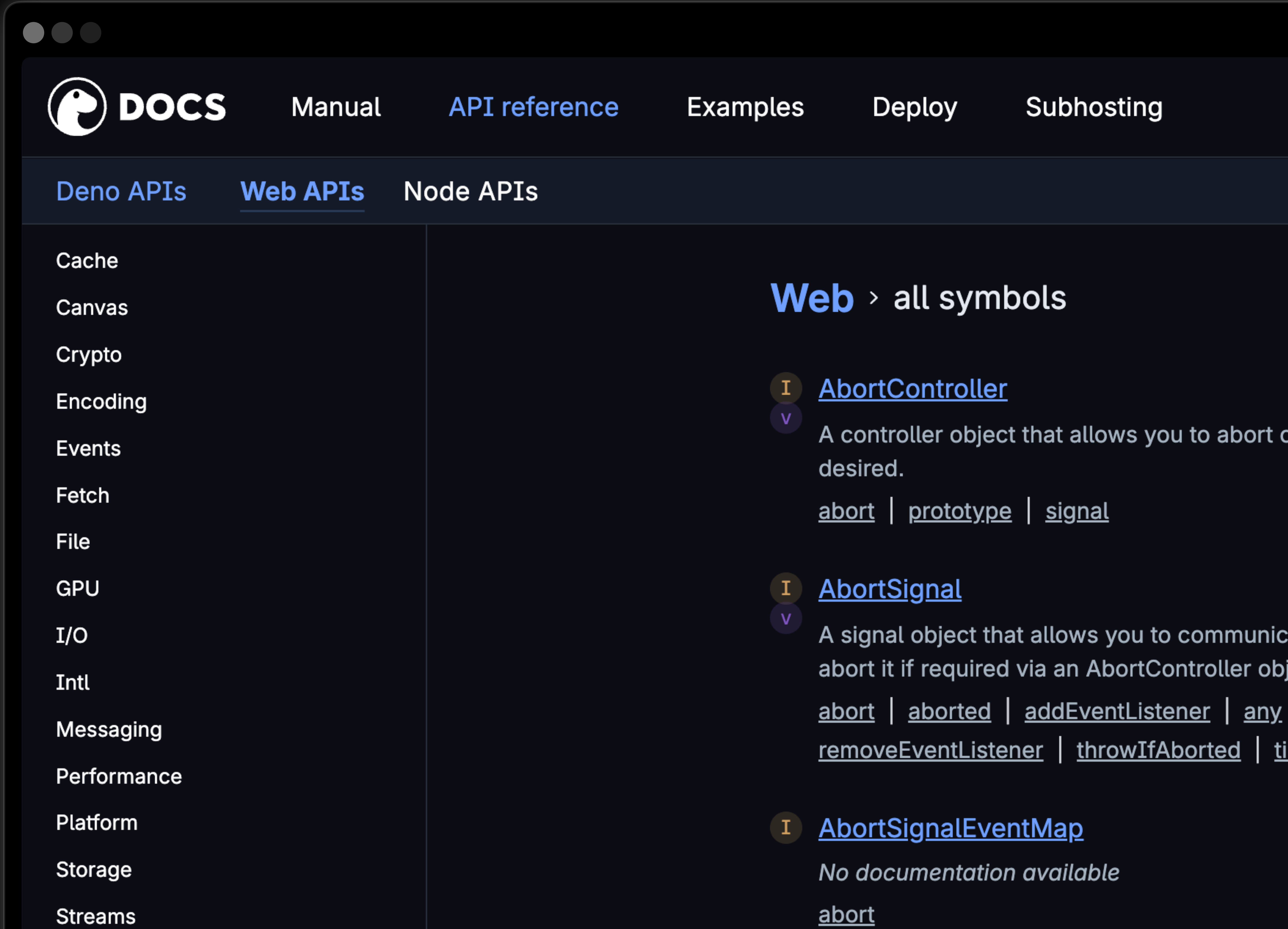
I know this.

What is Deno?

docs.deno.com

Familiar APIs

Heavily based on web standards
there is less re-learning required



Web › all symbols

I V AbortController

A controller object that allows you to abort o
desired.

abort | prototype | signal

I V AbortSignal

A signal object that allows you to communic
abort it if required via an AbortController obj

abort | aborted | addEventListener | any
removeEventListener | throwIfAborted | ti

I AbortSignalEventMap

No documentation available

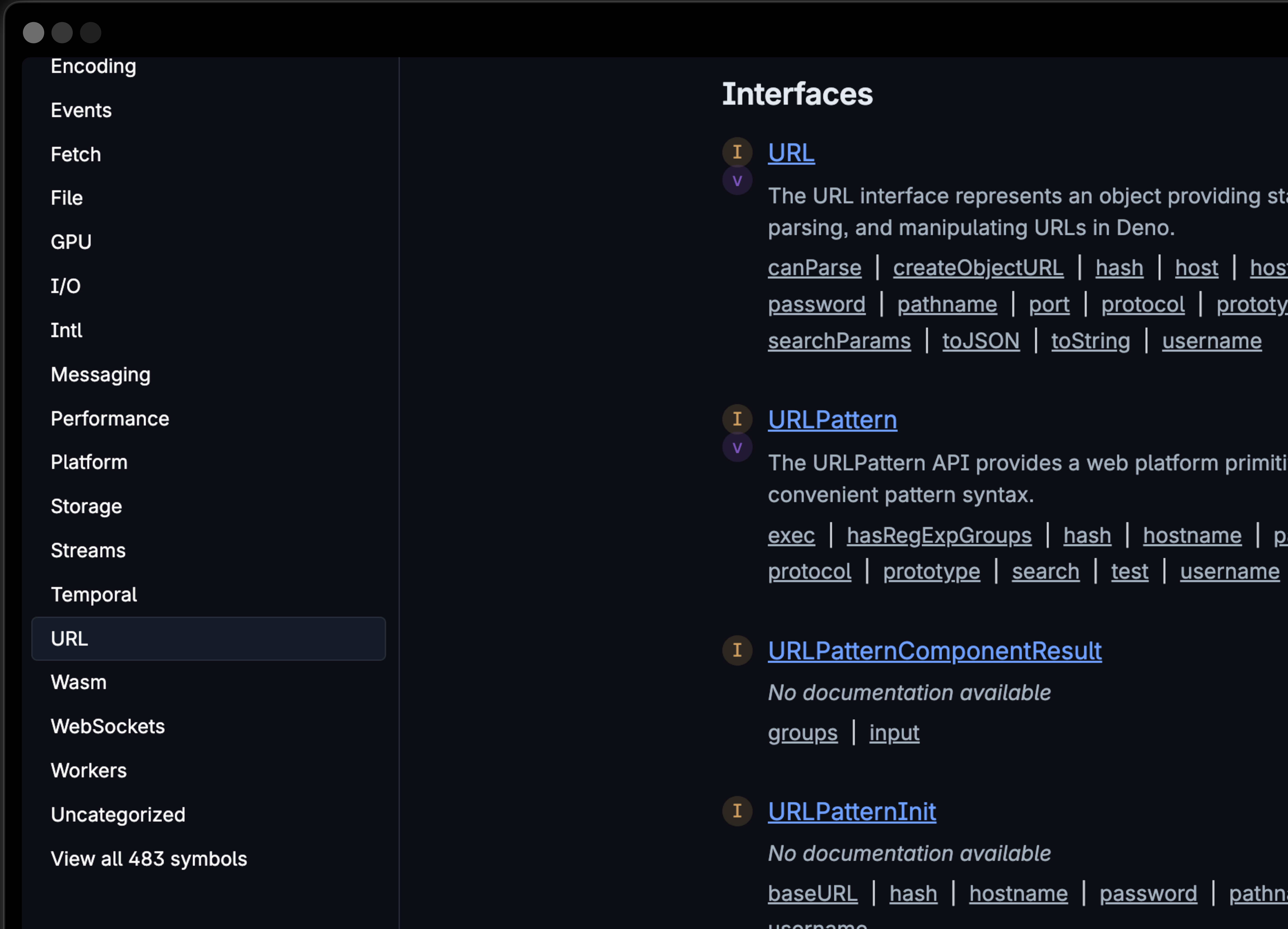
abort

What is Deno?

docs.deno.com

Familiar APIs

Heavily based on web standards there is less re-learning required



Encoding
Events
Fetch
File
GPU
I/O
Intl
Messaging
Performance
Platform
Storage
Streams
Temporal
URL
Wasm
WebSockets
Workers
Uncategorized
View all 483 symbols

Interfaces

I V URL

The URL interface represents an object providing st
parsing, and manipulating URLs in Deno.

canParse | createObjectURL | hash | host | hos
password | pathname | port | protocol | prototy
searchParams | toJSON | toString | username

I V URLPattern

The URLPattern API provides a web platform primiti
convenient pattern syntax.

exec | hasRegExpGroups | hash | hostname | p
protocol | prototype | search | test | username

I URLPatternComponentResult

No documentation available

groups | input

I URLPatternInit

No documentation available

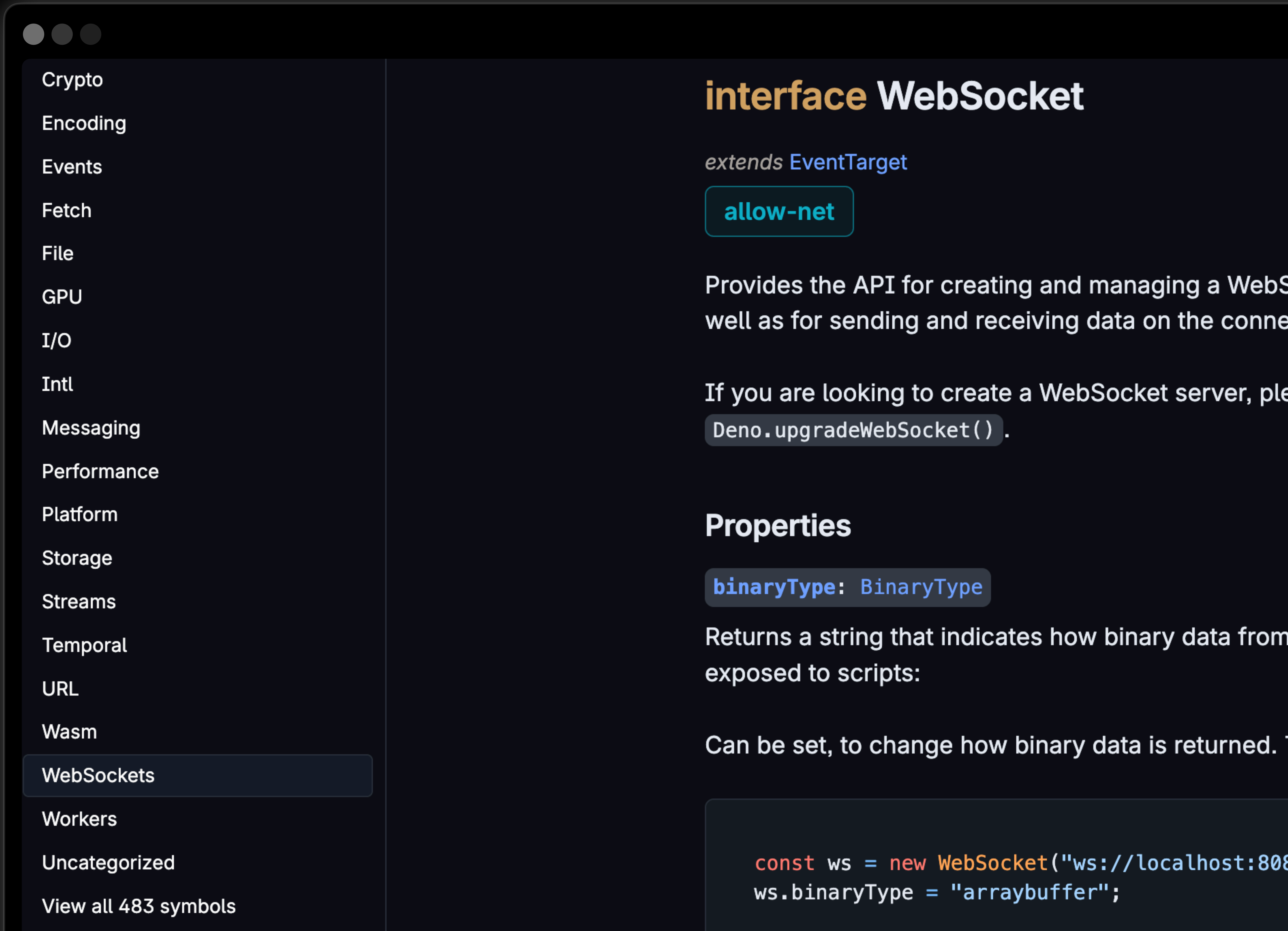
baseURL | hash | hostname | password | pathn

What is Deno?

docs.deno.com

Familiar APIs

Heavily based on web standards there is less re-learning required



Crypto
Encoding
Events
Fetch
File
GPU
I/O
Intl
Messaging
Performance
Platform
Storage
Streams
Temporal
URL
Wasm
WebSockets
Workers
Uncategorized
View all 483 symbols

interface WebSocket

extends EventTarget

allow-net

Provides the API for creating and managing a WebS
well as for sending and receiving data on the conne

If you are looking to create a WebSocket server, ple
`Deno.upgradeWebSocket()`.

Properties

`binaryType: BinaryType`

Returns a string that indicates how binary data from
exposed to scripts:

Can be set, to change how binary data is returned.

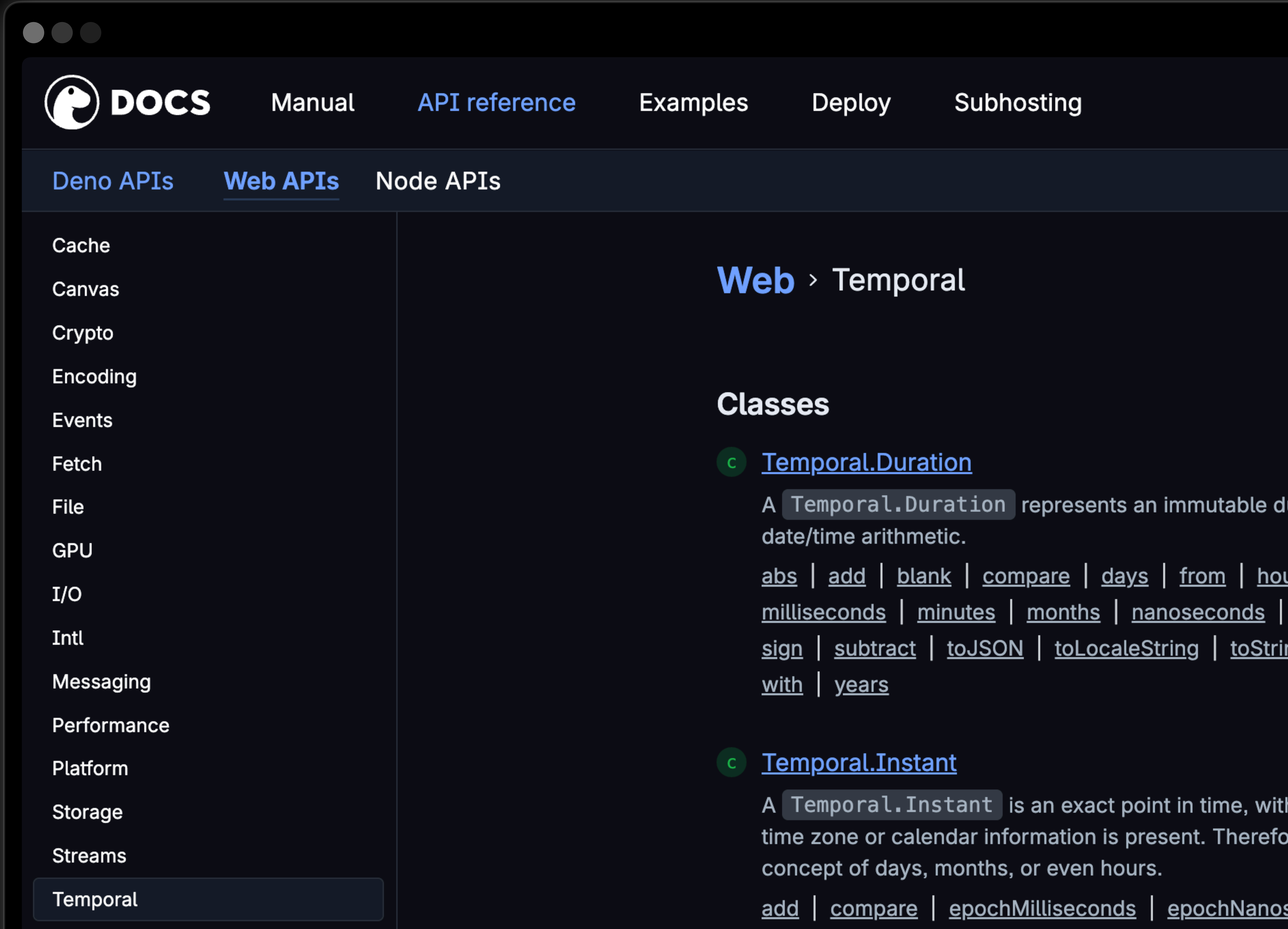
```
const ws = new WebSocket("ws://localhost:808
ws.binaryType = "arraybuffer";
```

What is Deno?

docs.deno.com

Familiar APIs

Heavily based on web standards there is less re-learning required


DOCS Manual API reference Examples Deploy Subhosting

Deno APIs Web APIs Node APIs

Cache
Canvas
Crypto
Encoding
Events
Fetch
File
GPU
I/O
Intl
Messaging
Performance
Platform
Storage
Streams
Temporal

Web › Temporal

Classes

c Temporal.Duration

A `Temporal.Duration` represents an immutable d
date/time arithmetic.

abs | add | blank | compare | days | from | hou
milliseconds | minutes | months | nanoseconds |
sign | subtract | toJSON | toLocaleString | toStri
with | years

c Temporal.Instant

A `Temporal.Instant` is an exact point in time, with
time zone or calendar information is present. Therefo
concept of days, months, or even hours.

add | compare | epochMilliseconds | epochNanos

Shallow on-ramp

A familiar entry point for those familiar with node

```
node main.js
```

A familiar entry point for those familiar with node

```
deno main.js
```

A familiar entry point for those familiar with node

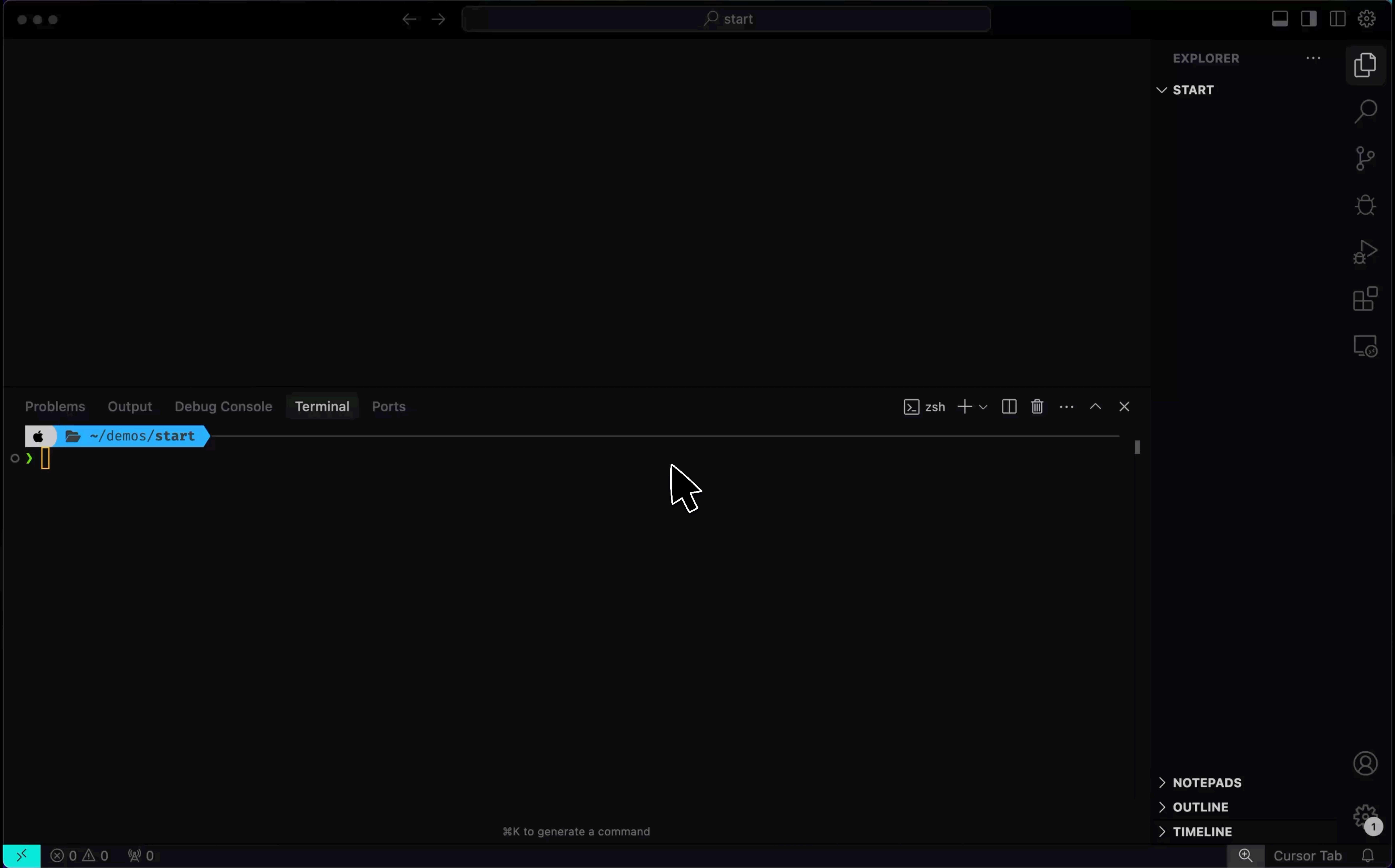
```
deno main.ts
```

A familiar entry point for those familiar with node

```
npm init
```

A familiar entry point for those familiar with node

```
deno init
```



```
 ~/demos/start
❯
```

⌘K to generate a command

Deno for Visual Studio Code

https://deno.co/vscode

A familiar entry point for those familiar with node

```
npm install
```

Deno first contact.

main.ts

```
import * as emoji from "node-emoji";

console.log(
  emoji.emojify(`:sauropod: :heart:  npm`)
);
```

Deno first contact.

main.ts

```
import * as emoji from "npm:node-emoji";

console.log(
  emoji.emojify(`:sauropod: :heart:  npm`)
);
```

Deno first contact.

main.ts

```
import * as emoji from "npm:node-emoji";

console.log(
  emoji.emojify(`:sauropod: :heart:  npm`)
);
```

Deno first contact.

```
deno run main.ts
```

🦕❤️ npm

Dependency management.

Dependency management.

Dependencies cached globally

Locally

Dependency management.

```
deno install
```

Dependency management.

```
deno add
```

Dependency management.

```
deno add npm:node-emoji
```

Dependency management.

deno.json

```
"imports": {
  "react": "npm:react@18.2.0",
  "vite": "npm:vite@^6.2.4",
  "@std/assert": “jsr:@std/assert@1.0.12"
  "node-emoji": “npm:node-emoji@^2.2.0",
}
```

Dependency management.

main.ts

```ts
import * as emoji from "npm:node-emoji";

console.log(
  emoji.emojify(`:sauropod: :heart:  npm`)
);
```

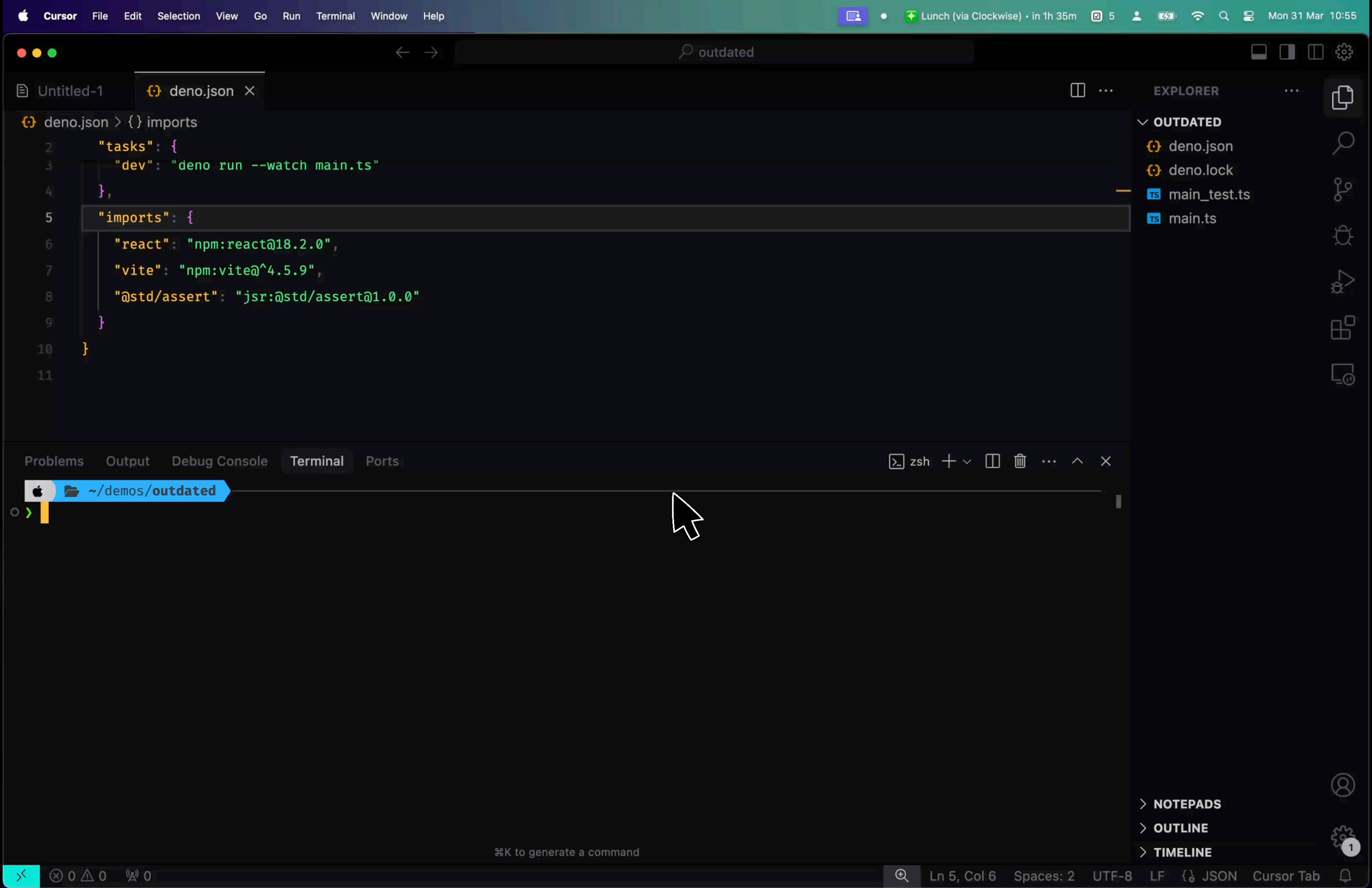
Dependency management.

main.ts

```
import * as emoji from "node-emoji";

console.log(
  emoji.emojify(`:sauropod: :heart:  npm`)
);
```

Dependency management.

```
deno outdated
```



```json
  "tasks": {
    "dev": "deno run --watch main.ts"
  },
  "imports": {
    "react": "npm:react@18.2.0",
    "vite": "npm:vite@^4.5.9",
    "@std/assert": "jsr:@std/assert@1.0.0"
  }
}
```

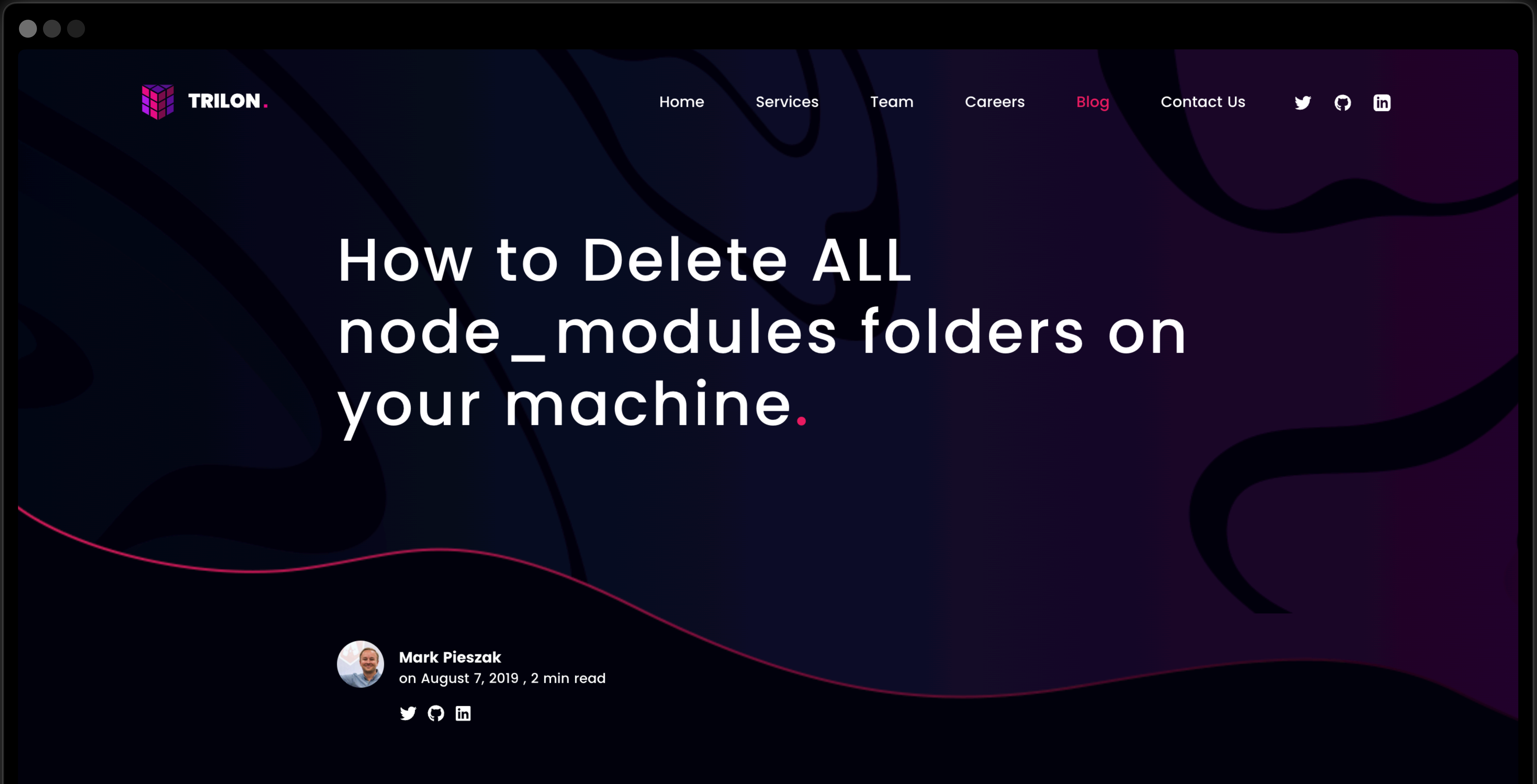
Dependency management.

Locally

Dependencies cached globally

Dependency management.

https://trilon.io/blog/how-to-delete-all-nodemodules-recursively



TRILON.

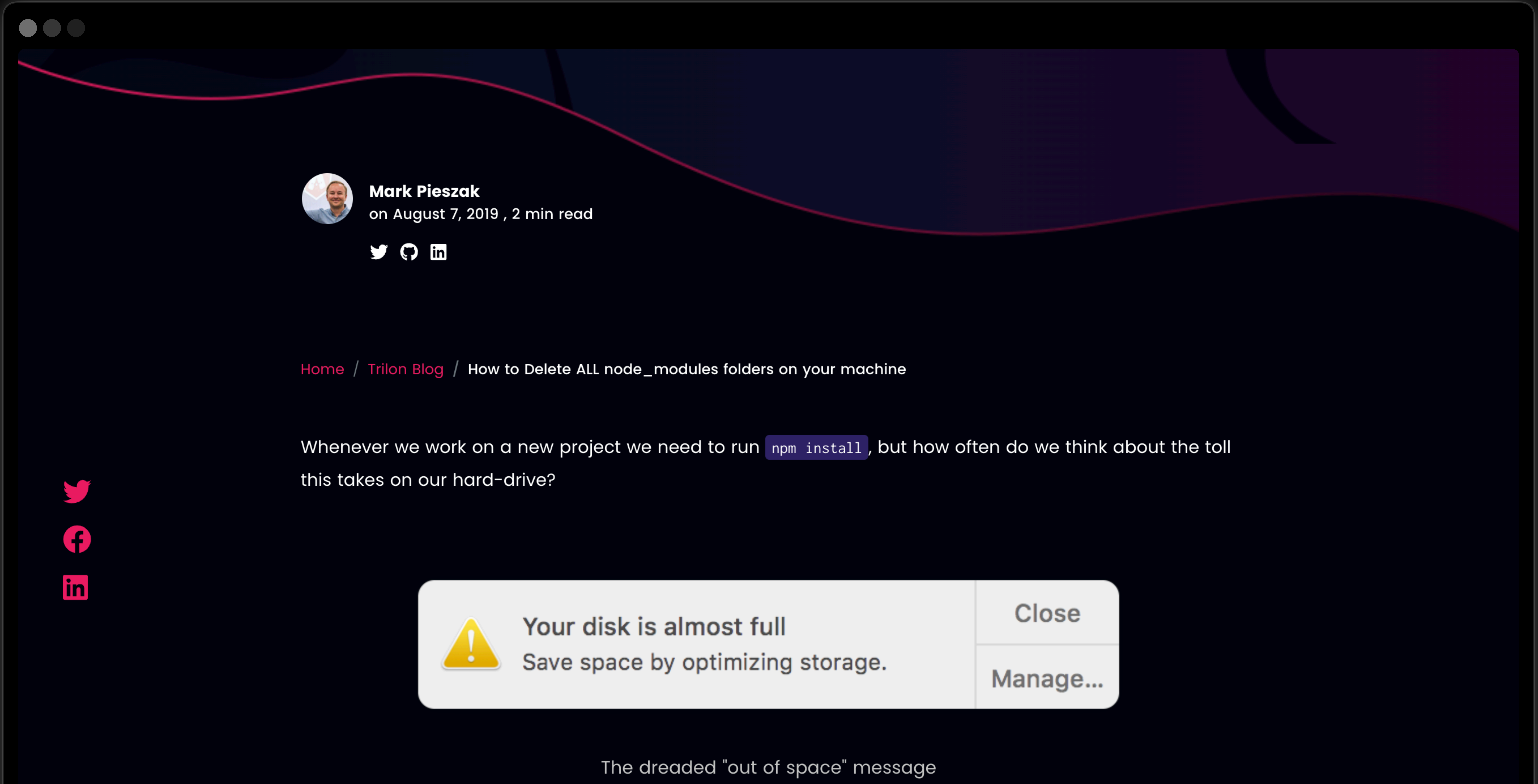
Home Services Team Careers Blog Contact Us

How to Delete ALL node_modules folders on your machine.

Mark Pieszak
on August 7, 2019 , 2 min read

Dependency management.

https://trilon.io/blog/how-to-delete-all-nodemodules-recursively



Mark Pieszak
on August 7, 2019 , 2 min read

Home / Trilon Blog / How to Delete ALL node_modules folders on your machine

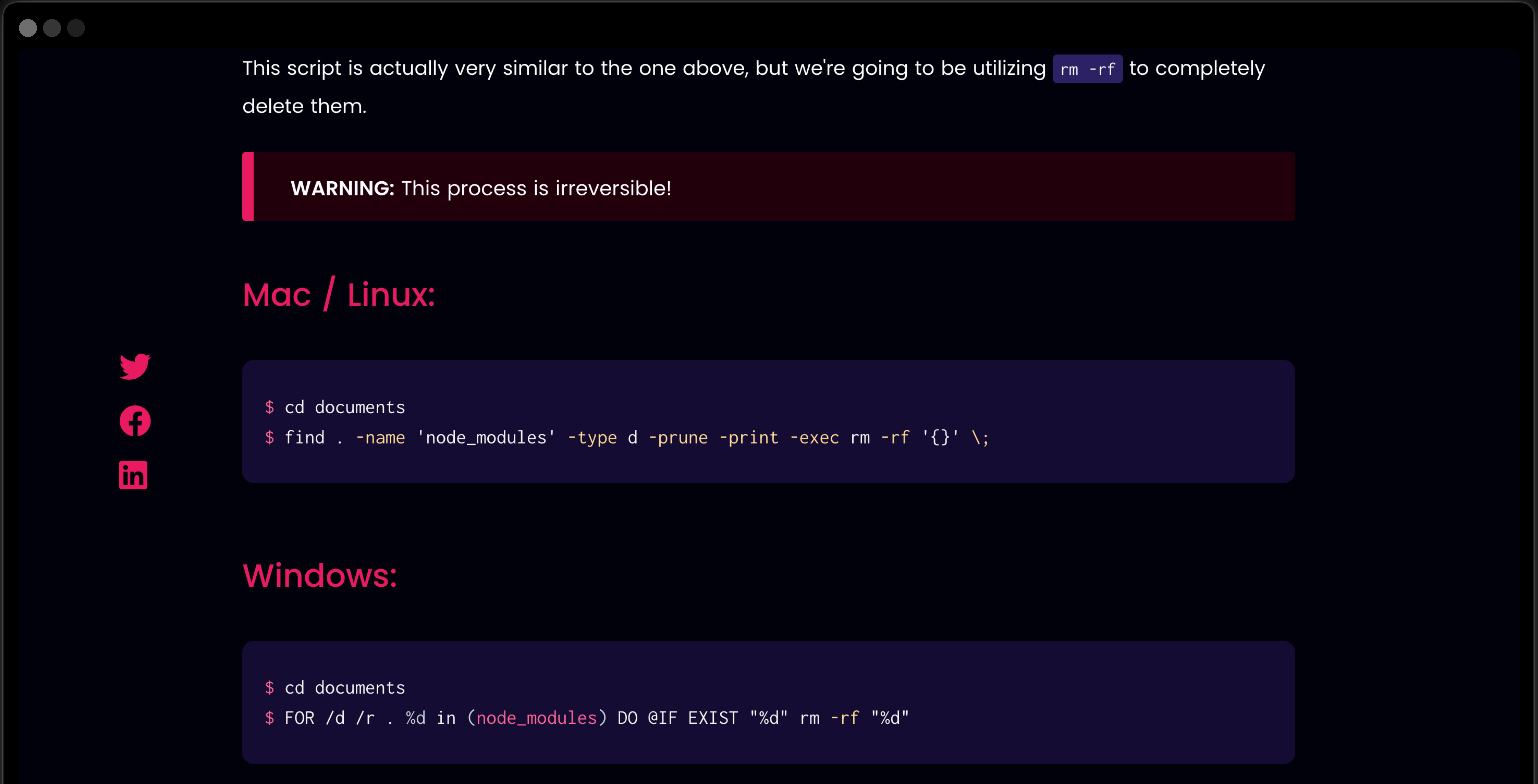
Whenever we work on a new project we need to run `npm install`, but how often do we think about the toll this takes on our hard-drive?

Your disk is almost full
Save space by optimizing storage.

The dreaded "out of space" message

Dependency management.

https://trilon.io/blog/how-to-delete-all-nodemodules-recursively



This script is actually very similar to the one above, but we're going to be utilizing `rm -rf` to completely delete them.

WARNING: This process is irreversible!

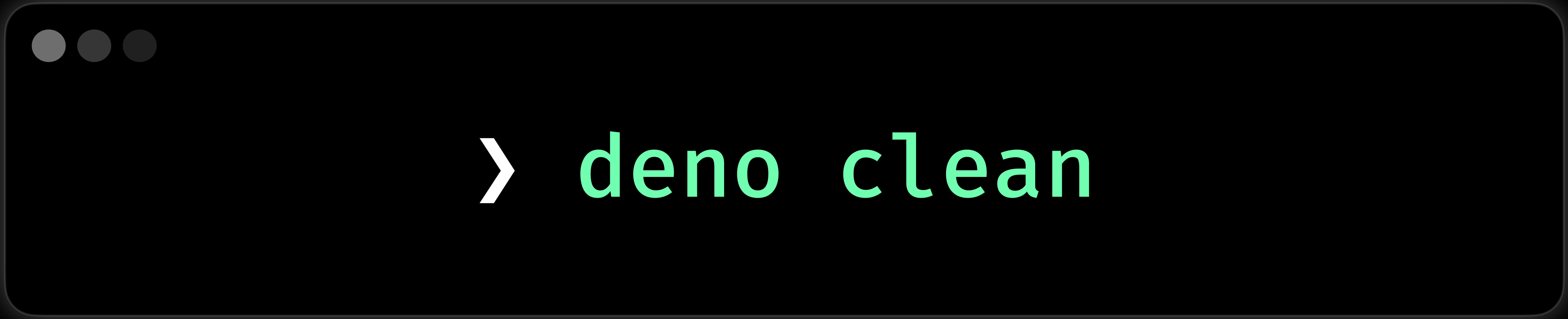
Mac / Linux:

```
$ cd documents
$ find . -name 'node_modules' -type d -prune -print -exec rm -rf '{}' \;
```

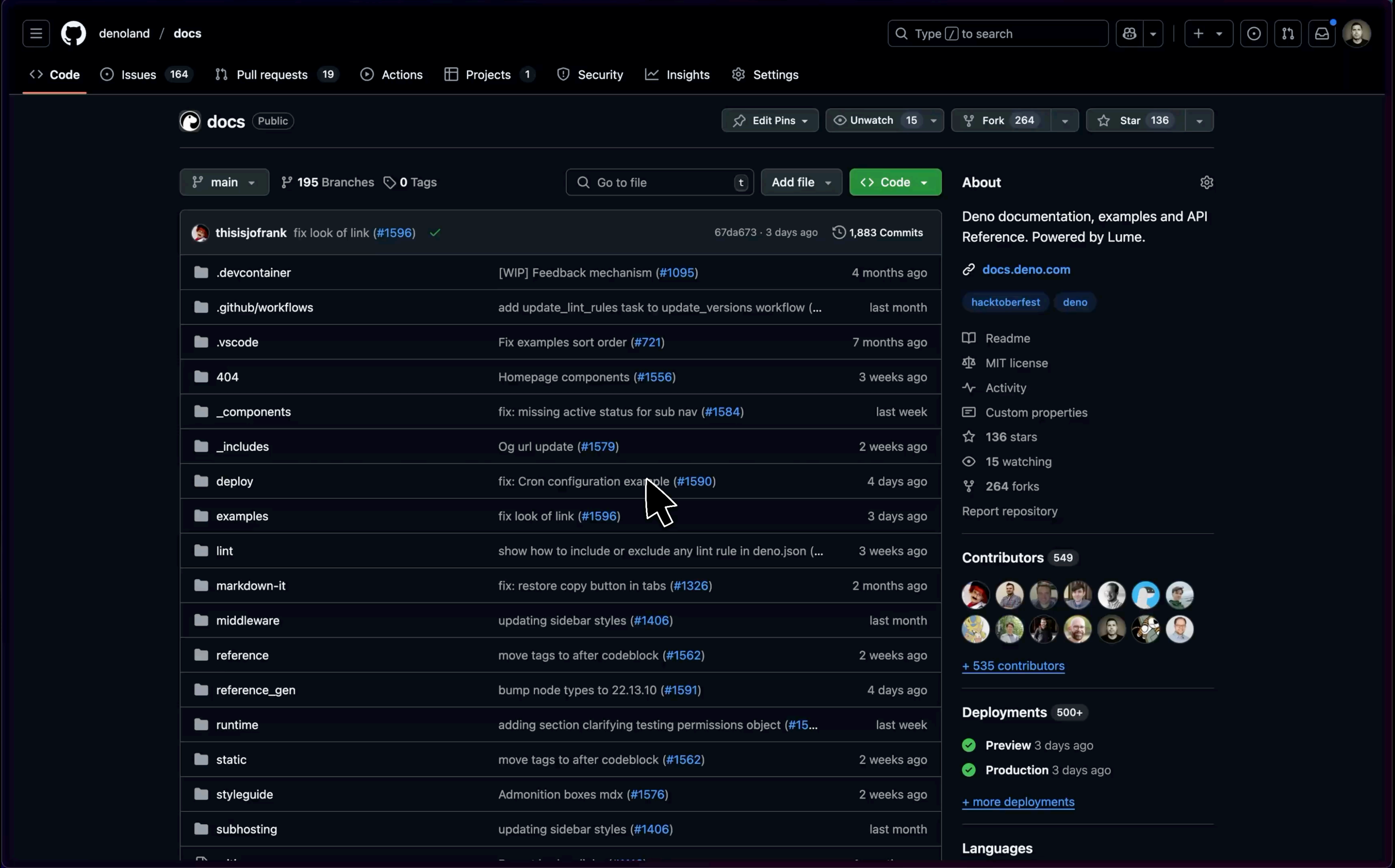
Windows:

```
$ cd documents
$ FOR /d /r . %d in (node_modules) DO @IF EXIST "%d" rm -rf "%d"
```

Dependency management.



```
❯ deno clean
```



docs Public

195 Branches 0 Tags

thisisjofrank fix look of link (#1596) 67da673 · 3 days ago 1,883 Commits

.devcontainer	[WIP] Feedback mechanism (#1095)	4 months ago
.github/workflows	add update_lint_rules task to update_versions workflow (...	last month
.vscode	Fix examples sort order (#721)	7 months ago
404	Homepage components (#1556)	3 weeks ago
_components	fix: missing active status for sub nav (#1584)	last week
_includes	Og url update (#1579)	2 weeks ago
deploy	fix: Cron configuration example (#1590)	4 days ago
examples	fix look of link (#1596)	3 days ago
lint	show how to include or exclude any lint rule in deno.json (...	3 weeks ago
markdown-it	fix: restore copy button in tabs (#1326)	2 months ago
middleware	updating sidebar styles (#1406)	last month
reference	move tags to after codeblock (#1562)	2 weeks ago
reference_gen	bump node types to 22.13.10 (#1591)	4 days ago
runtime	adding section clarifying testing permissions object (#15...	last week
static	move tags to after codeblock (#1562)	2 weeks ago
styleguide	Admonition boxes mdx (#1576)	2 weeks ago
subhosting	updating sidebar styles (#1406)	last month

About

Deno documentation, examples and API Reference. Powered by Lume.

docs.deno.com

hacktoberfest deno

Readme
MIT license
Activity
Custom properties
136 stars
15 watching
264 forks
Report repository

Contributors 549

+ 535 contributors

Deployments 500+

Preview 3 days ago
Production 3 days ago
+ more deployments

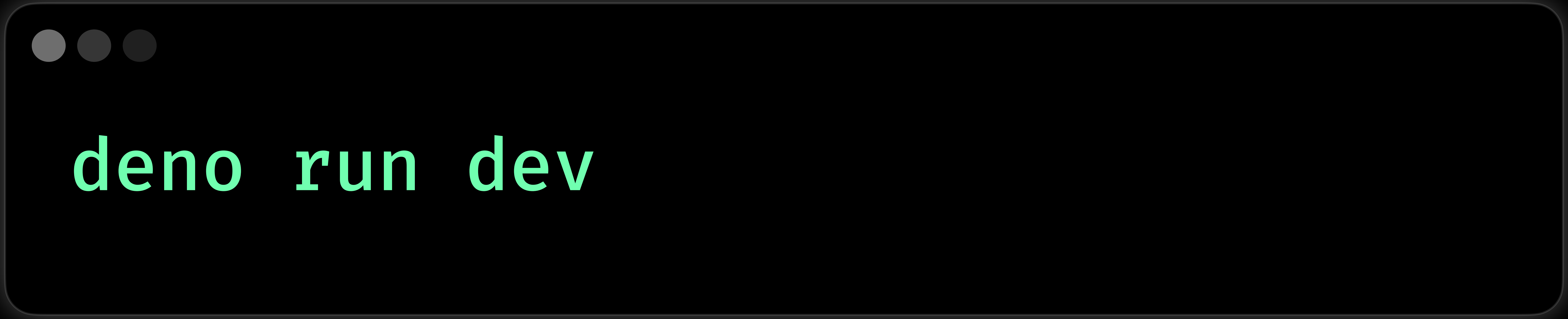
Languages

Task runner.

Running tasks.

```
npm run dev
```

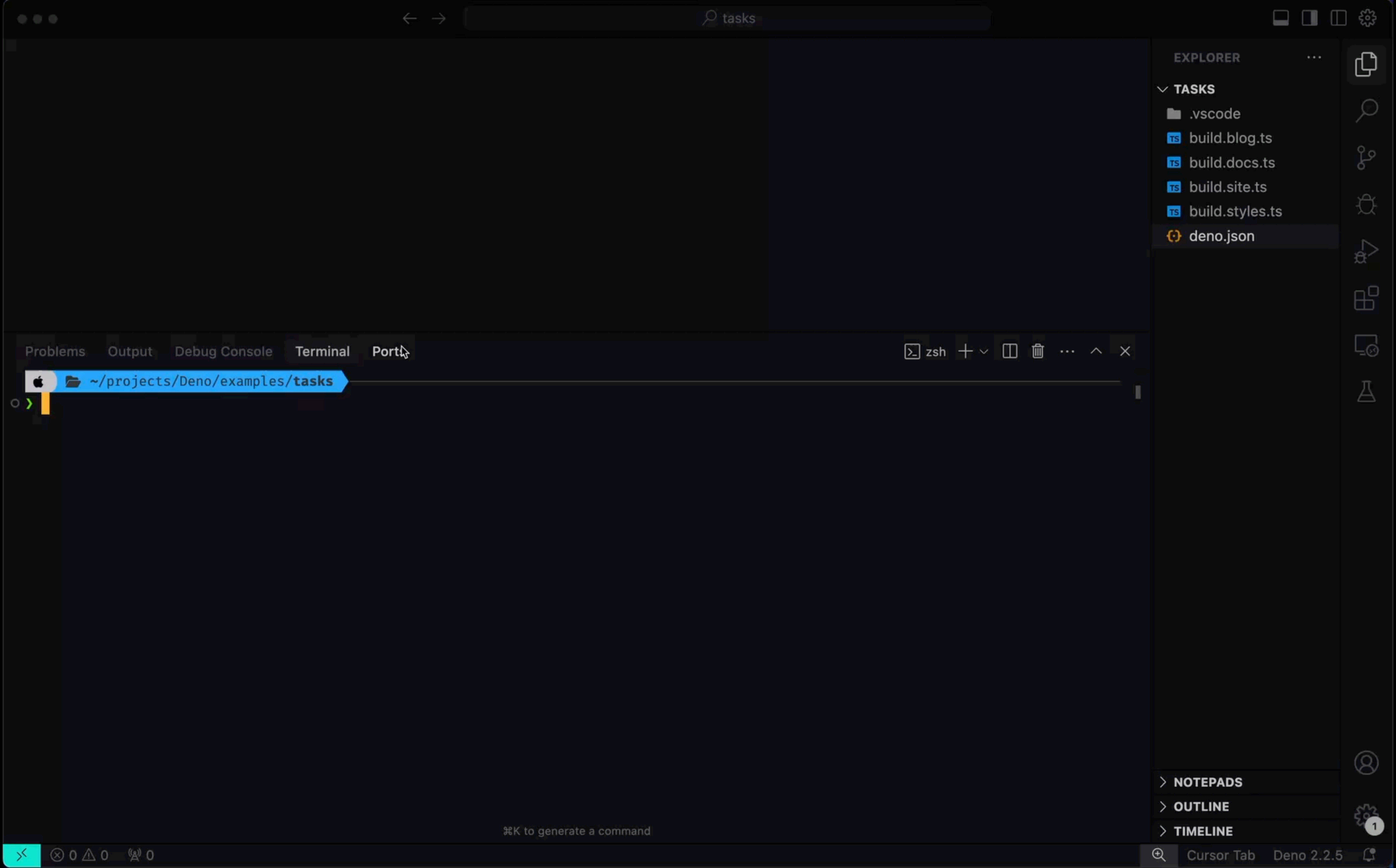
Running tasks.



```
deno run dev
```

Running tasks.

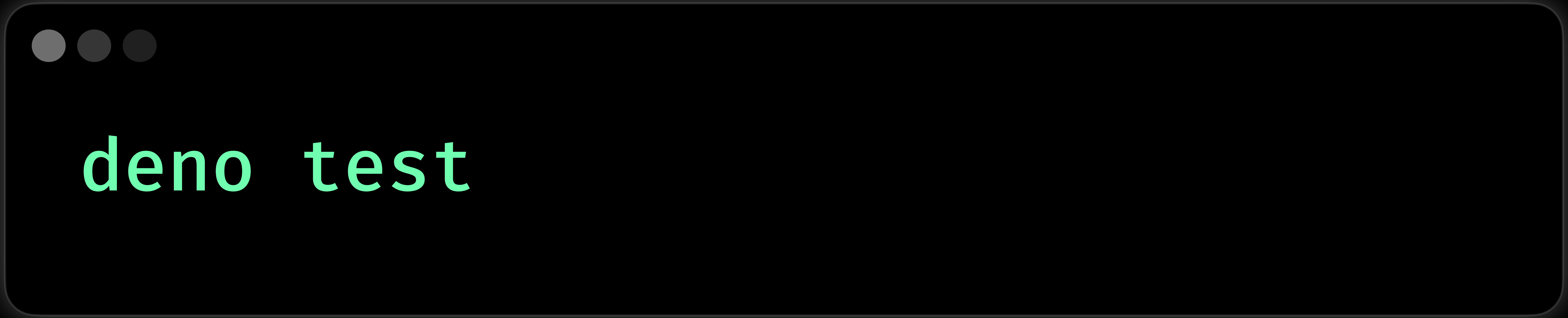
```
deno task dev
```



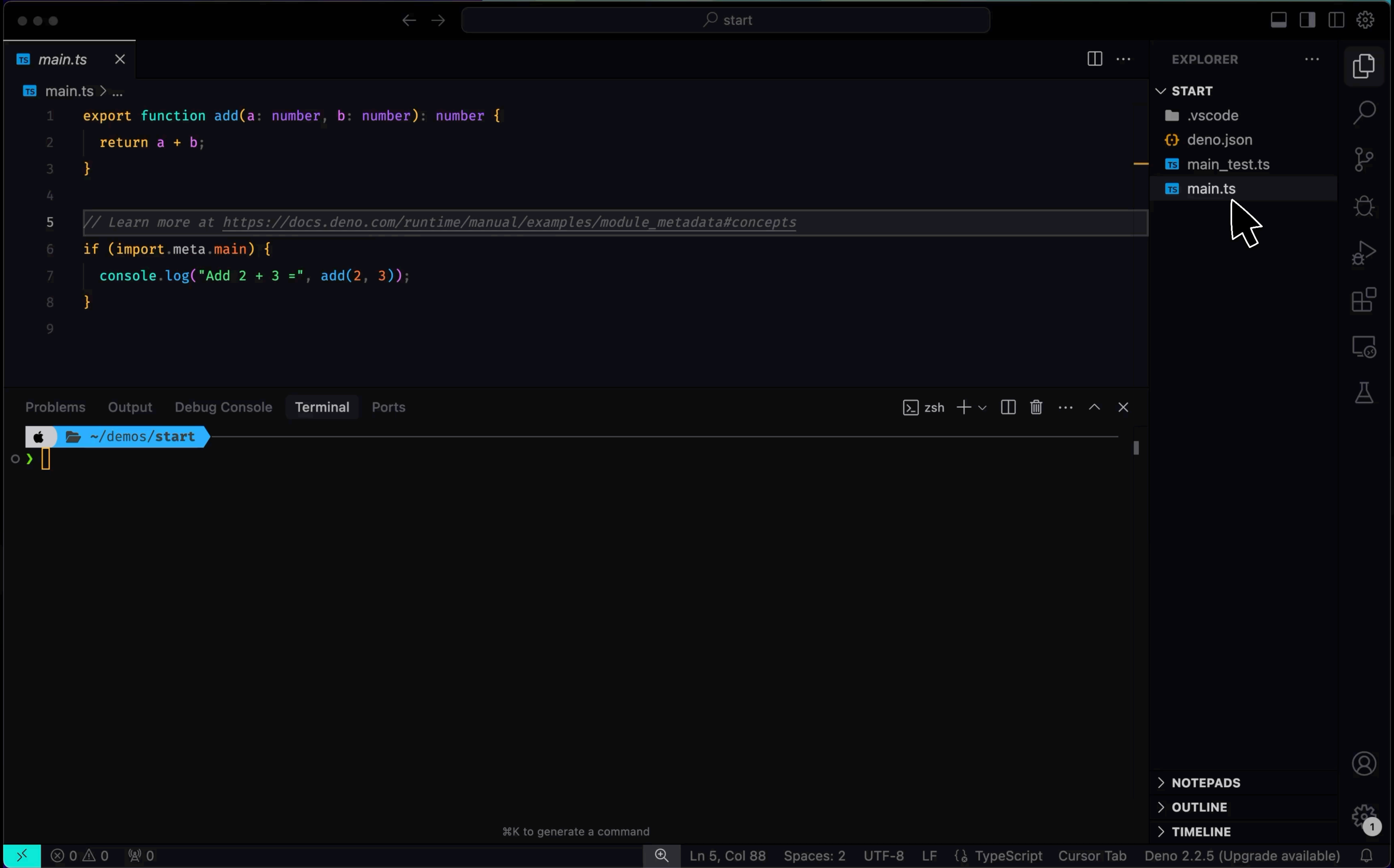
```
~/projects/Deno/examples/tasks
❯
```

Test runner.

Test framework.



```
deno test
```



```
export function add(a: number, b: number): number {
  return a + b;
}

// Learn more at https://docs.deno.com/runtime/manual/examples/module_metadata#concepts
if (import.meta.main) {
  console.log("Add 2 + 3 =", add(2, 3));
}
```

```
~/demos/start
❯
```

Test framework.

main_test.ts

main.test.ts

Test framework.

Mocking

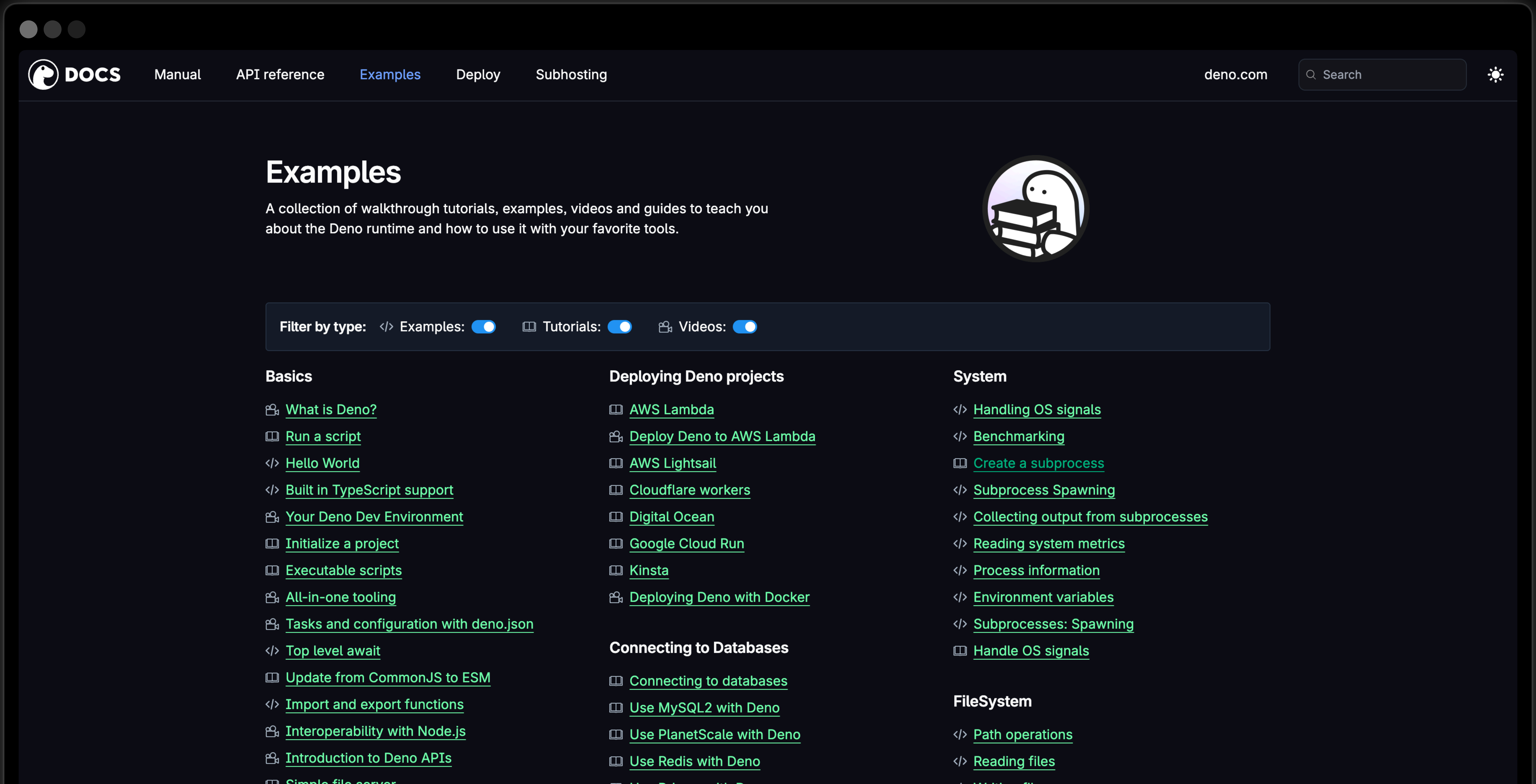
Async testing

Coverage

A variety of test styles

Test framework.

docs.deno.com/examples



Examples

A collection of walkthrough tutorials, examples, videos and guides to teach you about the Deno runtime and how to use it with your favorite tools.

Filter by type: Examples: on · Tutorials: on · Videos: on

Basics

- What is Deno?
- Run a script
- Hello World
- Built in TypeScript support
- Your Deno Dev Environment
- Initialize a project
- Executable scripts
- All-in-one tooling
- Tasks and configuration with deno.json
- Top level await
- Update from CommonJS to ESM
- Import and export functions
- Interoperability with Node.js
- Introduction to Deno APIs
- Simple file server

Deploying Deno projects

- AWS Lambda
- Deploy Deno to AWS Lambda
- AWS Lightsail
- Cloudflare workers
- Digital Ocean
- Google Cloud Run
- Kinsta
- Deploying Deno with Docker

Connecting to Databases

- Connecting to databases
- Use MySQL2 with Deno
- Use PlanetScale with Deno
- Use Redis with Deno

System

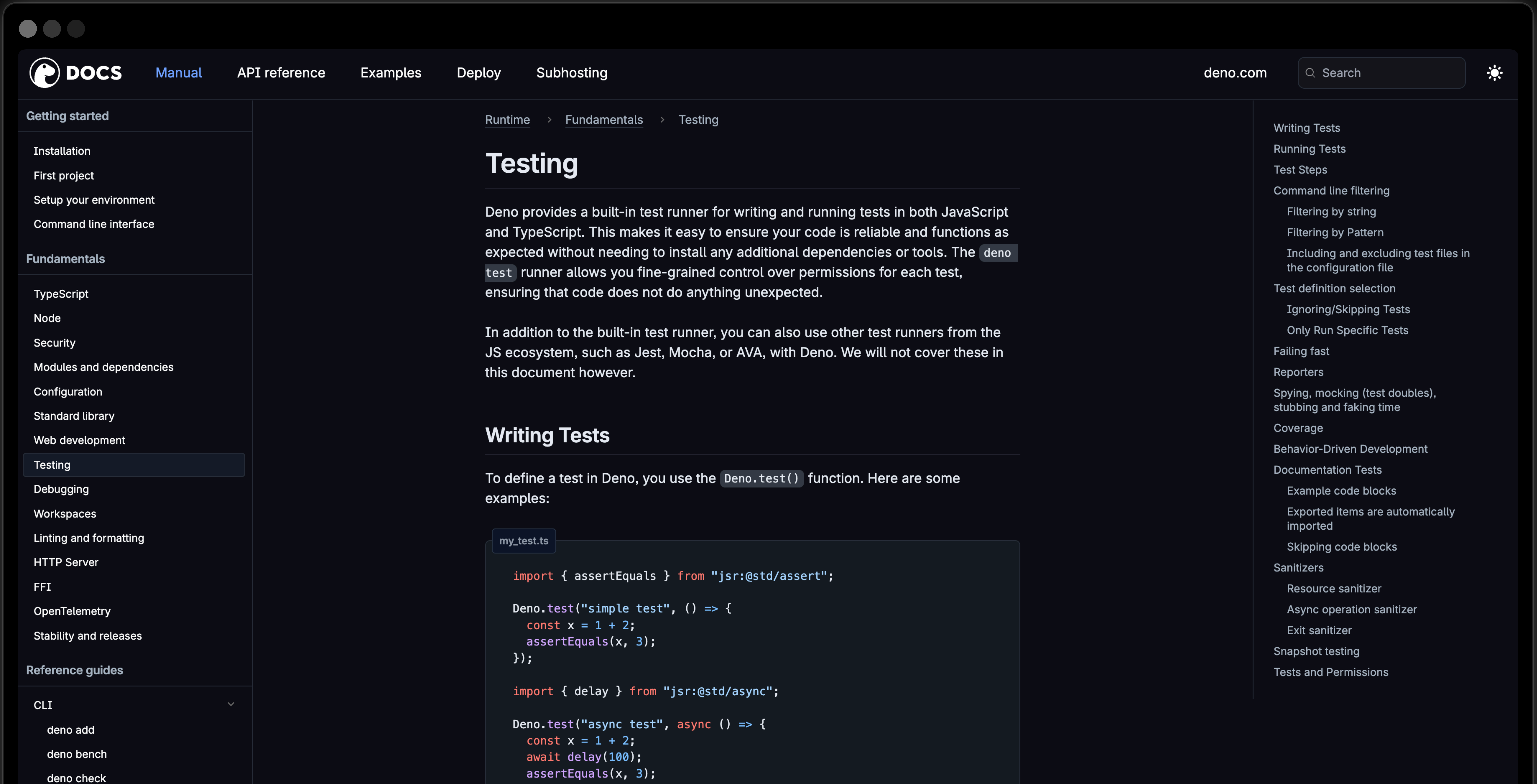
- Handling OS signals
- Benchmarking
- Create a subprocess
- Subprocess Spawning
- Collecting output from subprocesses
- Reading system metrics
- Process information
- Environment variables
- Subprocesses: Spawning
- Handle OS signals

FileSystem

- Path operations
- Reading files

Test framework.

docs.deno.com/examples/testing_tutorial



Runtime › Fundamentals › Testing

Testing

Deno provides a built-in test runner for writing and running tests in both JavaScript and TypeScript. This makes it easy to ensure your code is reliable and functions as expected without needing to install any additional dependencies or tools. The `deno test` runner allows you fine-grained control over permissions for each test, ensuring that code does not do anything unexpected.

In addition to the built-in test runner, you can also use other test runners from the JS ecosystem, such as Jest, Mocha, or AVA, with Deno. We will not cover these in this document however.

Writing Tests

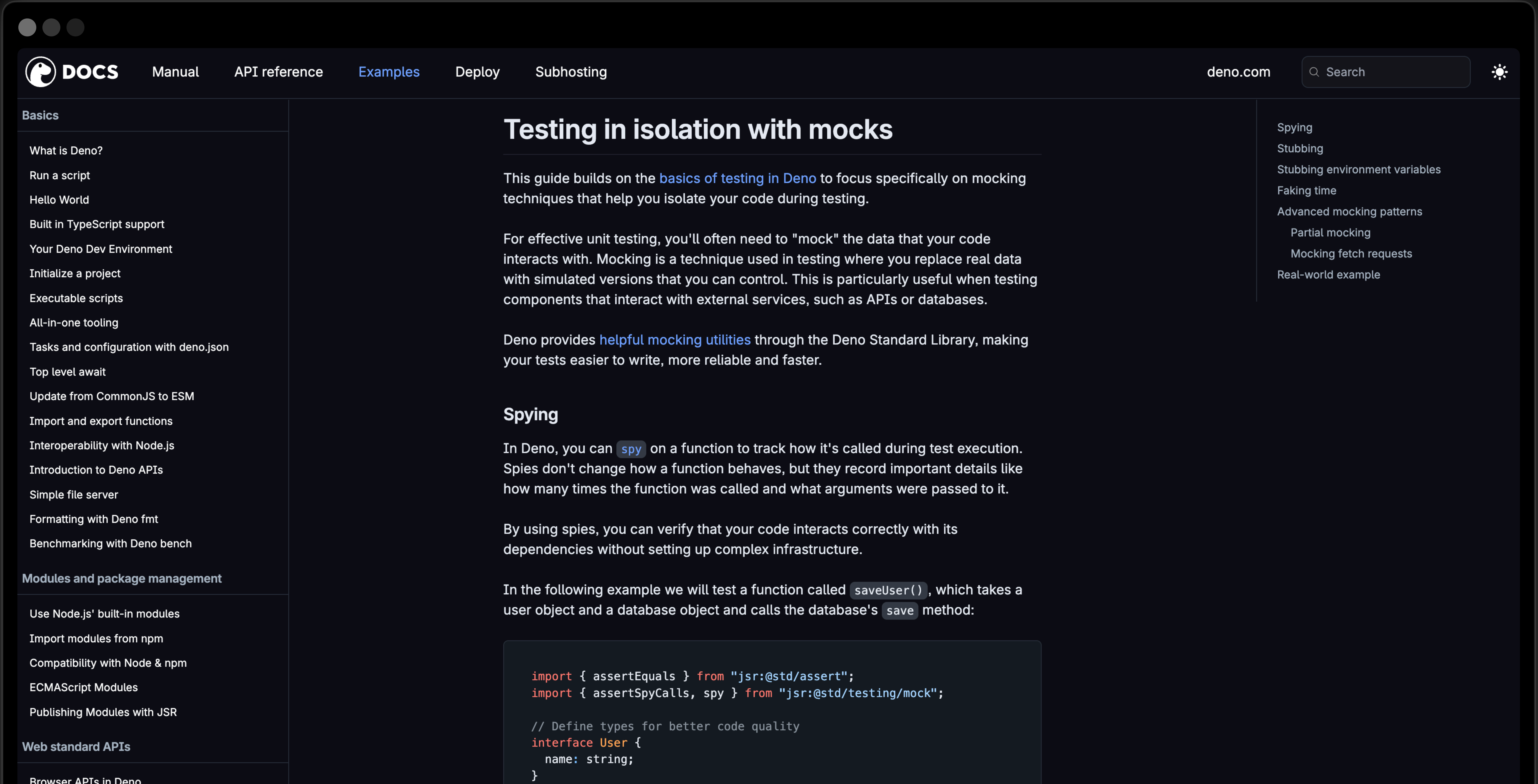
To define a test in Deno, you use the `Deno.test()` function. Here are some examples:

my_test.ts

```
import { assertEquals } from "jsr:@std/assert";

Deno.test("simple test", () => {
  const x = 1 + 2;
  assertEquals(x, 3);
});

import { delay } from "jsr:@std/async";

Deno.test("async test", async () => {
  const x = 1 + 2;
  await delay(100);
  assertEquals(x, 3);
```

Test framework.

docs.deno.com/runtime/fundamentals/testing



Testing in isolation with mocks

This guide builds on the basics of testing in Deno to focus specifically on mocking techniques that help you isolate your code during testing.

For effective unit testing, you'll often need to "mock" the data that your code interacts with. Mocking is a technique used in testing where you replace real data with simulated versions that you can control. This is particularly useful when testing components that interact with external services, such as APIs or databases.

Deno provides helpful mocking utilities through the Deno Standard Library, making your tests easier to write, more reliable and faster.

Spying

In Deno, you can `spy` on a function to track how it's called during test execution. Spies don't change how a function behaves, but they record important details like how many times the function was called and what arguments were passed to it.

By using spies, you can verify that your code interacts correctly with its dependencies without setting up complex infrastructure.

In the following example we will test a function called `saveUser()`, which takes a user object and a database object and calls the database's `save` method:

```
import { assertEquals } from "jsr:@std/assert";
import { assertSpyCalls, spy } from "jsr:@std/testing/mock";

// Define types for better code quality
interface User {
  name: string;
}
```

TypeScript toolchain.

JavaScript toolchain.

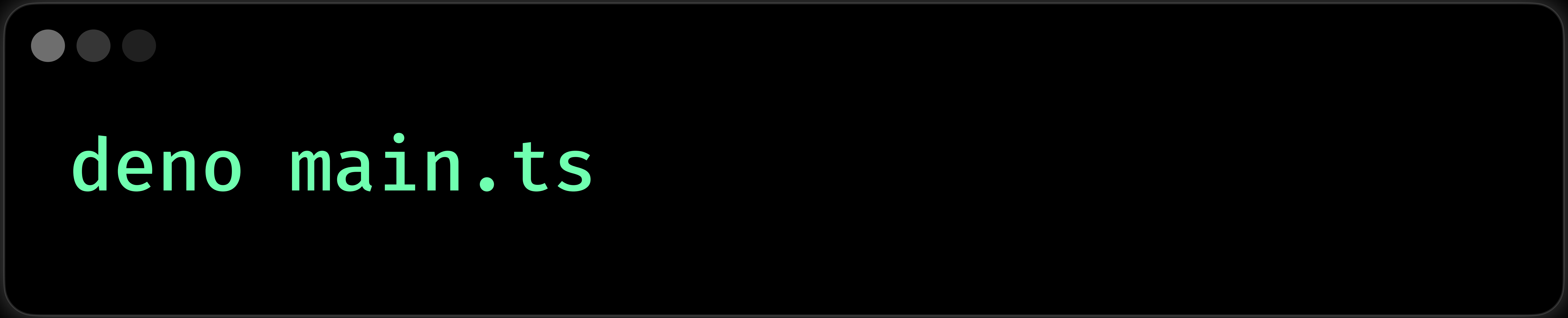
Development toolchain.

It’s already built in

Toolchain.

```
deno main.js
```

Toolchain.

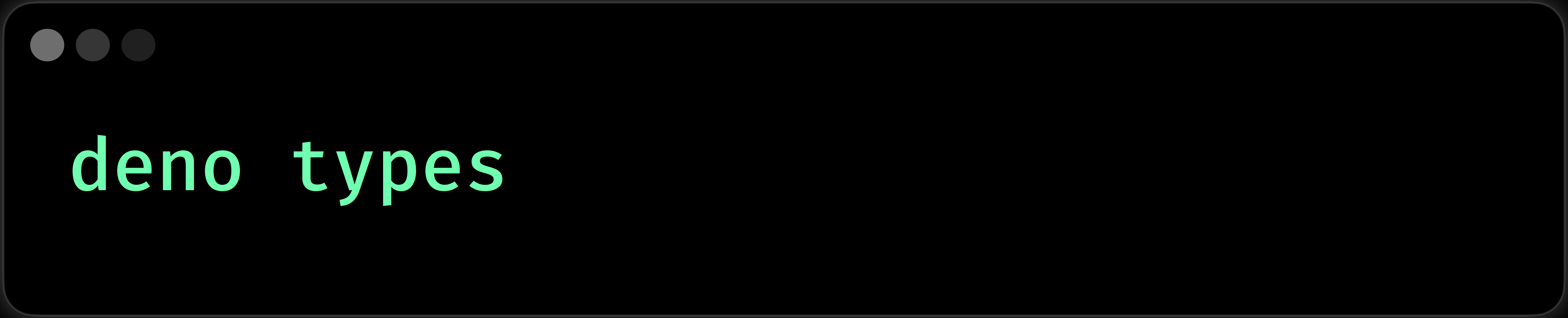


```
deno main.ts
```

Toolchain.

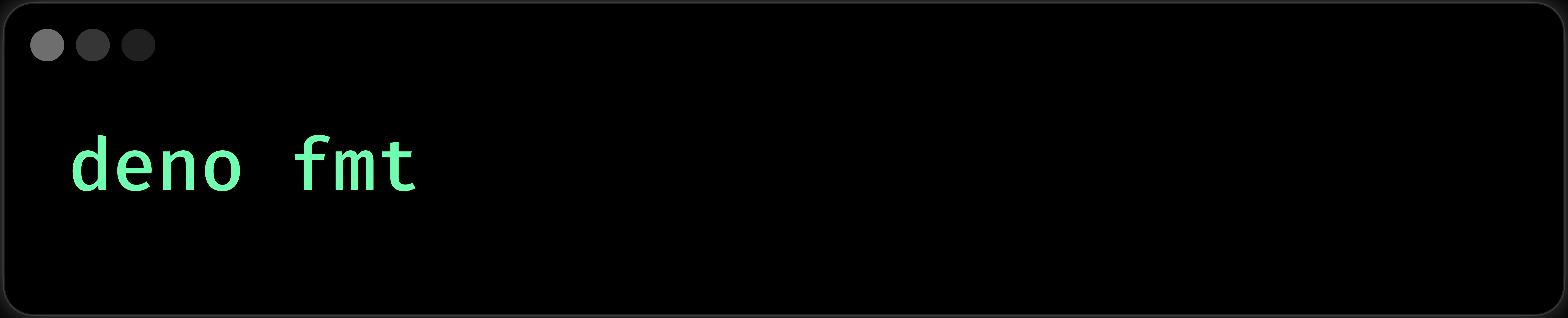
```
deno check
```

Toolchain.



```
deno types
```

Toolchain.



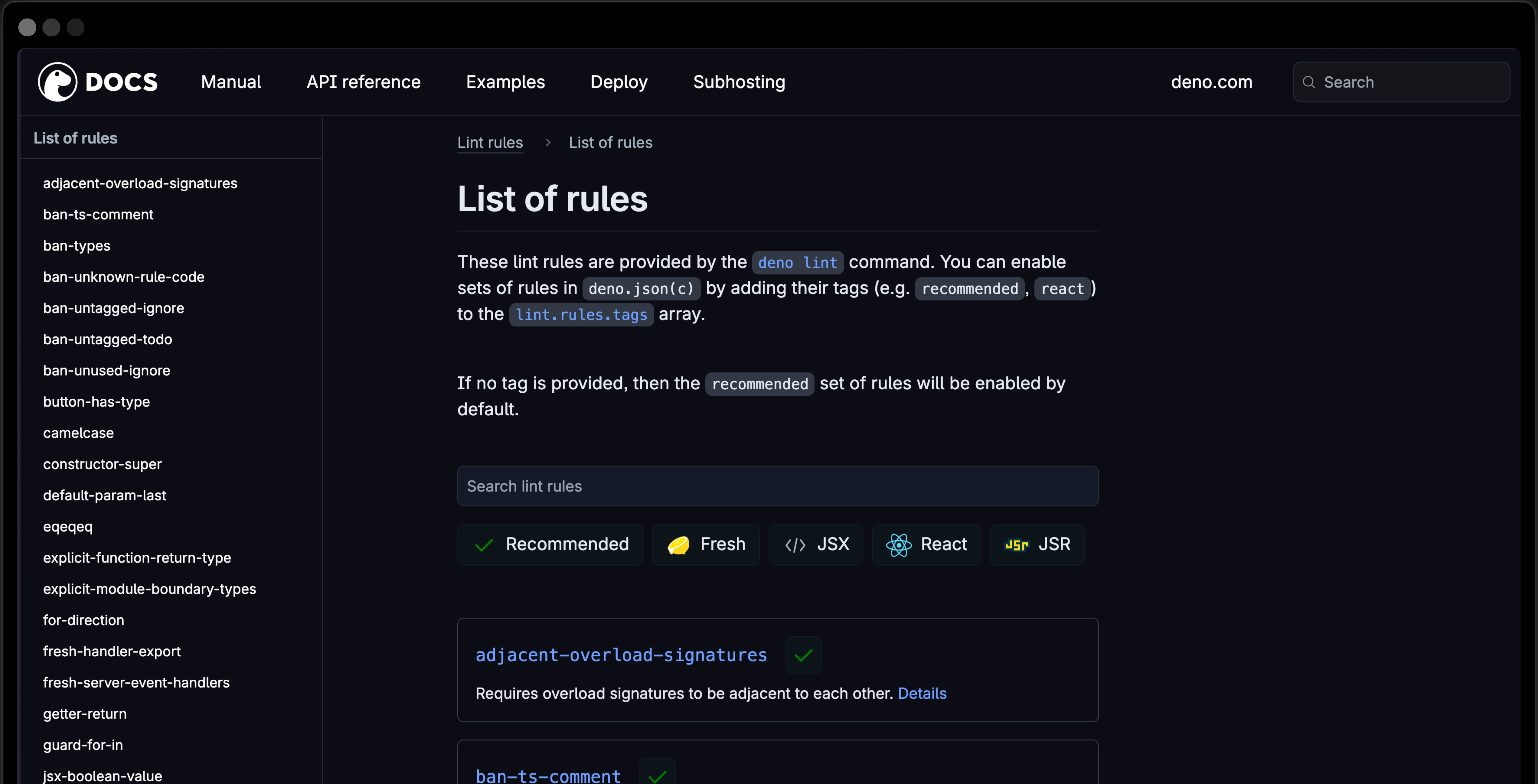
```
deno fmt
```

Toolchain.

```
deno lint
```

Toolchain.

docs.deno.com/lint



DOCS Manual API reference Examples Deploy Subhosting deno.com Search

List of rules

adjacent-overload-signatures
ban-ts-comment
ban-types
ban-unknown-rule-code
ban-untagged-ignore
ban-untagged-todo
ban-unused-ignore
button-has-type
camelcase
constructor-super
default-param-last
eqeqeq
explicit-function-return-type
explicit-module-boundary-types
for-direction
fresh-handler-export
fresh-server-event-handlers
getter-return
guard-for-in
jsx-boolean-value

Lint rules › List of rules

List of rules

These lint rules are provided by the `deno lint` command. You can enable sets of rules in `deno.json(c)` by adding their tags (e.g. `recommended`, `react`) to the `lint.rules.tags` array.

If no tag is provided, then the `recommended` set of rules will be enabled by default.

Search lint rules

✓ Recommended · Fresh · JSX · React · JSR

`adjacent-overload-signatures` ✓

Requires overload signatures to be adjacent to each other. Details

`ban-ts-comment` ✓

Why I’m excited.

Reasons to be excited.

Simplification and unification of tools

Reasons to be excited.

Regular improvements

Reasons to be excited.

I kinda sorta know most of it already

Lower friction and fewer quirks

Reasons to be excited.

Security and performance

Recommended resources.

docs.deno.com/examples

Deno examples and tutorials

discord.gg/deno

Deno Discord

bsky.app/profile/deno.land

Deno on Bluesky

www.youtube.com/watch?v=N04NrI37Ies

Deno 2 Deep Dive | Exploring the Deno Ecosystem

Thanks.

philhawksworth.dev

findthat.at/hello-deno

deno