



A short journey to **deno.land**

Workshop im Kurs Web-Technologien am 06.07.2021

Benedikt Engel

Ablauf

- Was ist und woher kommt Deno?
- Deno vs. Node.js
- Deno CLI & Try it yourself
- Packages & Modules
- Standard Library
- Runtime API
- Try it yourself

A secure runtime for JavaScript and TypeScript.

A short journey to deno.land
Benedikt Engel

Was ist und woher kommt Deno?

- Von Ryan Dahl
- Basiert auf der `V8-Engine`
- In `Rust` geschrieben
- Version 1.0 erschien am 13. Mai 2020
- Aktuelle Version: 1.11.4

Deno vs. Node.js

	Node.js	Deno
Support	JavaScript	JavaScript, TypeScript
Modul-System	CommonJS* <code>const module = require('module')</code>	ES-Modules <code>import { test } from "Path-toTest"</code>
Package-Manager	NPM / YARN	Kein Package-Manager nötig, in Deno integriert
Kontrolle über Dateisystem und Netzwerk	✗	✓
Top Level Await	✗	✓

Deno CLI

Subcommands

- ▶ `deno`
- ▶ `deno bundle`
- ▶ `deno cache`
- ▶ `deno compile`
- ▶ `deno completions`
- ▶ `deno coverage`
- ▶ `deno doc`
- ▶ `deno eval`
- ▶ `deno fmt`
- ▶ `deno help`
- ▶ `deno info`
- ▶ `deno install`
- ▶ `deno lint`
- ▶ `deno lsp`
- ▶ `deno repl`
- ▶ `deno run`
- ▶ `deno test`
- ▶ `deno types`
- ▶ `deno upgrade`

A short journey to deno.land
Benedikt Engel

Deno CLI

Subcommands

- ▶ deno
- ▶ deno bundle
- ▶ deno cache
- ▶ deno compile
- ▶ deno completions
- ▶ deno coverage
- ▶ deno doc
- ▶ deno eval
- ▶ deno fmt
- ▶ deno help
- ▶ deno info
- ▶ deno install
- ▶ deno lint
- ▶ deno lsp
- ▶ deno repl
- ▶ deno run
- ▶ deno test
- ▶ deno types
- ▶ deno upgrade

A short journey to **deno.land**
Benedikt Engel

Deno CLI

deno run

- ▶ `--allow-all` Erlaubt alle Zugriffe auf Netzwerk und Filesystem
- ▶ `--allow-net` Erlaubt alle Zugriffe aufs Netzwerk
- ▶ `--allow-net=<PATH>` Erlaubt Zugriffe auf angegebene URL im Netzwerk
- ▶ `--allow-read` Erlaubt Lesen des gesamten Fileyststem
- ▶ `--allow-read=<PATH>` Erlaubt Lesen des angegebenen Pfad
- ▶ `--allow-write` Erlaubt Schreiben im gesamten Filesystem
- ▶ `--allow-write=<PATH>` Erlaubt Schreiben im angegebenen Pfad
- ▶ `--watch` Startet Prozess bei Datei-Änderungen neu (Unstable)

Deno CLI

Subcommands

- ▶ `deno`
- ▶ `deno bundle`
- ▶ `deno cache`
- ▶ `deno compile`
- ▶ `deno completions`
- ▶ `deno coverage`
- ▶ `deno doc`
- ▶ `deno eval`
- ▶ `deno fmt`
- ▶ `deno help`
- ▶ `deno info`
- ▶ `deno install`
- ▶ `deno lint`
- ▶ `deno lsp`
- ▶ `deno repl`
- ▶ `deno run`
- ▶ `deno test`
- ▶ `deno types`
- ▶ `deno upgrade`

A short journey to **deno.land**
Benedikt Engel

Packages & Modules

Standard Library & Third Party Modules

- Kein Package-Manager
- Kein `node-modules` (bzw. `deno-modules`) Ordner
- Kein `package.json`
- Einbindung erfolgt direkt über URL als ES-Modules
- Werden unter `$HOME/Library/Caches/deno` bzw. `%LOCALAPPDATA%/deno` einmalig beim ersten Ausführen gecached

Packages & Modules

Einbinden & Nutzen

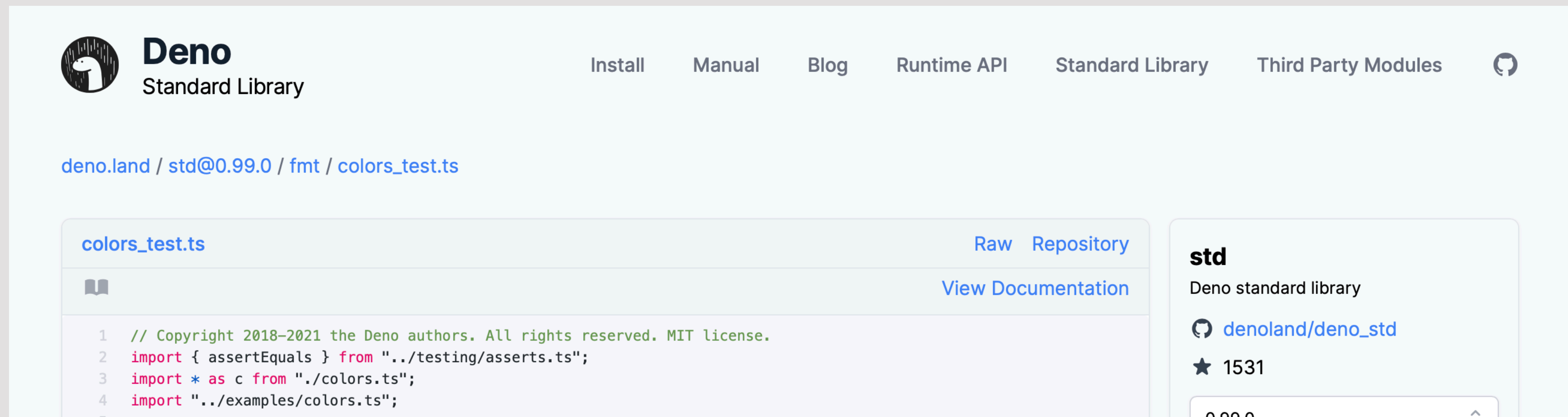


```
// Spezifische Version nutzen:  
import { serve } from "https://deno.land/std@0.100.0/http/server.ts";  
  
// Immer neuste Version nutzen:  
import { serve } from "https://deno.land/std/http/server.ts";
```

Packages & Modules

Informationen bekommen

- Link aus dem Importstatement im Browser öffnen, dies zeigt den Sourcecode
- In einem File auf ‘View Documentation’ klicken



Runtime API

RuntimeAPI-Doku

- Stellt viele hilfreiche Funktionen und Variablen zur Verfügung
 - Web Standards (wie fetch, setTimeout, etc.) - Nutzung direkt
 - Nicht Web Standards (wie read, chmod, etc.) - Nutzung durch Deno-
Namespace. Bsp: `Deno.cwd()` zum auslesen des aktuellen Verzeichnis

Standard Library & Runtime API - Try it yourself

Aufgabe 1

Schreibe ein Skript, welches CL-Argumente entgegen nimmt und in einer Konstante speichert.

Erweitere das Skript, sodass geprüft wird ob das Argument / die Argumente eine Datei, bzw. Dateien im aktuellen Ordner darstellen. Wenn die Datei vorhanden ist soll der Inhalt in Stdout ausgegeben werden, ist dies nicht der Fall soll die Datei mit dem entsprechenden Namen angelegt werden und die Bemerkung "Autogenerated File by our Deno-Program" hineingeschrieben werden.

Standard Library & Runtime API - Try it yourself

Aufgabe 2

Binde den http-Server aus der Standard Library in eine neue Datei ein und führe ihn aus.

Erweitere dein Skript, sodass die Files aus dem Ordner `./httpdocs` ausgelesen werden. Auf diesen ausgelesenen Namen sollen nun Routen im Server erstellt werden, sodass beim aufrufen von `/about` die Datei `about.html` zurückgeschickt wird.

Weiterführende Lektüre & Tutorials

Wenn du mit Deno richtig durchstarten willst

- ▶ Kostenloses Ebook Deno Succinctly
- ▶ FreeCodeCamp Tutorial Deno Course - Better than Node.js?
- ▶ Traversy Media Tutorial Deno Crash Course
- ▶ Awesome-List awesome deno
- ▶ Deno Dokumentationen RuntimeAPI, StandardLibrary



A short journey to **deno.land**

Workshop im Kurs Web-Technologien am 06.07.2021

Benedikt Engel