

Updated on 16/05/2023

Sign up

## Deno training

3 days (21 hours)

### Presentation

Deno is a command-line runtime for executing JavaScript and [Typescript](#) code. Like NodeJS, Deno is built on Chrome's [V8](#) JavaScript engine. It is written entirely in [Rust](#) (Nodejs is written in C and C++), which promises a high level of performance. Finally, to guarantee asynchronous I/O, the event loop used is [Tokio](#). This is Rust's equivalent of [libuv](#) in NodeJS.

Deno has been around for almost 1 year. It [is currently under intensive development](#). Its aim is to become a productive and secure JavaScript development environment with native Typescript support. It looks a lot like NodeJS in use. A completely rewritten NodeJS with less restrictive technologies. Basically, it avoids all the problems that can no longer be solved with NodeJS today. And, of course, there are plenty of new features.

As always, we'll be teaching you the latest version of the program, [Deno 1.11](#).

### Objectives

- Know how to use Deno and NodeJS in the latest versions
- Developing a web application with Deno

### Target audience

- Web developers

### Prerequisites

- Knowledge of JavaScript and a client-side framework would be a plus, or knowledge of another object-oriented programming language (java, php, etc.).

## Further information

- Take our [Node.JS training course](#)

## Deno training program

- Introduction
- Philosophy chapter
  - Goals
  - Non-goals
- Chapter Setup
  - Binary Install
  - Build from source
    - Prerequisites
    - Other useful commands
- API reference chapter
  - deno types
  - Reference websites
- Chapter Examples
  - An implementation of the unix "cat" program
  - TCP echo server
  - Inspecting and revoking permissions
  - File server
  - Permissions whitelist
  - Run subprocess
  - Linking to third party code
  - Using external type definitions
  - Testing if current file is the main program
- Chapter Command line interface
  - Flags
  - Environmental variables
  - Shell completion
  - V8 flags
  - Bundling
  - Installing executable scripts
- Chapter Proxies
- Chapter Import maps

- Chapter Internal details
  - Deno and Linux analogy
    - Resources
    - Metrics
  - Schematic diagram
  - Profiling
  - Debugging with LLDB
  - Deno Core
  - Updating prebuilt binaries
  - Continuous Benchmarks
  - Logos
- Chapter Contributing
  - Submitting a pull request
  - Changes to third\_party
  - Adding Ops (aka bindings)
  - Documenting APIs
    - Utilize JSDoc

## Comparison with Node.js

Deno and Node.js are both runtime environments built on Google's V8 JavaScript engine, identical to that used in Google Chrome. Both feature internal event loops and provide command-line interfaces for script execution and a wide range of system utilities.

In the meantime, Deno differs mainly from Node.js in the following respects:

- Uses ES Module as default module system, instead of CommonJS.
- Uses URLs to load local or remote dependencies, similar to browsers.
- Includes an integrated package manager for resource recovery, so no need for NPM.
- Supports out-of-the-box TypeScript, using an instant TypeScript compiler with caching mechanisms.
- Aims for greater browser compatibility with a wide range of Web APIs.
- Control access to the file system and network to execute sandboxed code.
- API redefined to use Promises, ES6 and TypeScript features.
- Reduces the size of the main API while providing a large standard library with no external dependencies.

## Companies concerned

This course is aimed at companies, large or small, wishing to train their teams in a new, advanced computer technology.

## Teaching methods

Practical course: 60% Practical, 40% Theory. Training material distributed in digital format to all participants.

## Organization

The course alternates theoretical input from the trainer, supported by examples, with brainstorming sessions and group work.

## Validation

At the end of the session, a multiple-choice questionnaire verifies the correct acquisition of skills.

## Sanction

A certificate will be issued to each trainee who completes the course.