

# Node.js 12 Runtime Programming Using TypeScript

## Feature Tour, Event Loop, Non-Blocking IO, VM, Crypto, Utilities, Streaming, N-API, Project

Node.js is a cross-platform easy-to-use runtime based on Google's high performance V8 JavaScript execution engine (as found in Chrome). Node also comes with a well crafted and substantial framework which covers many application areas.

Node has a number of desirable core characteristics – simplicity (it is very easy to get started using Node); modularity (everything is based on modules, which can grow over time); extensible (both by JavaScript / TypeScript code and by C/C++ code [using N-API]) and asynchronous / event-driven (non-blocking IO).

This course cover the latest edition – Node.js 12 – from the ground up for developers with little or no previous Node experience but a strong desire to rapidly become proficient in Node. We compare Node to other popular runtimes that they might know and see there are many similarities (language VM, package management, framework layout, tools, etc.) but also some differences.

Note this course does not cover the Node.js HTTP[S/2] modules – we have a separate detailed course covering these along with Express and PUG, which is an ideal follow-on course to this one.

<b>Contents of One-Day Training Course</b>	
<p><b>Target Audience</b> Experienced web developers who wish to get up to speed developing for the Node.js runtime using TypeScript.</p> <p><b>Prerequisites</b> No previous experience of Node.js required.</p> <p>General experience with web programming is required.</p> <p>All demos and lab exercises will be in TypeScript, so attendees need to know TypeScript.</p>	<p><b>Feature Tour of Node.js 12</b> Node.js 12 is the latest edition of this very popular server-side runtime for web, CLI and other workloads – let's see what Node has to offer</p> <p><b>Node And TypeScript</b> Most Node apps up to now have used JavaScript – we will use TypeScript Benefits of TypeScript for server coding Using Node from TypeScript (.d.ts files)</p> <p><b>Launching Node</b> Command line options for Node itself Environment settings Keeping a node app running</p> <p><b>Event Loop</b> Review of how the event loop works The important role events play in Node The Events module EventEmitter Preference for asynchronous Role of listeners</p> <p><b>Non-Blocking IO</b> Accessing the file system Accessing networks</p> <p><b>Network Programming</b> DNS UDP/Datagram TCP using the Net module IPC servers using the Net module</p> <p><b>V8</b> Google's V8 engine provides the core execution environment for node Using the V8 module</p> <p><b>OS Module</b> Portable access to capabilities/services of the operating system on which Node runs</p> <p><b>VM</b> The VM module can be used to compile and execute code in a language VM</p> <p><b>Utilities</b> Utilities module TTY Console Timers StringDecoder</p> <p><b>Crypto</b> Cipher Decipher Handling certificates Hashing</p> <p><b>Streams</b> Stream types – readable, writable, duplex and transform The readline module process.stdout</p> <p><b>N-API</b> Building portable C/C++ modules that work with Node using the new N-API</p> <p><b>Advanced Node</b> Process and child processes The cluster module Shared server ports and child processes</p> <p><b>CLI Project</b> Building a command-line app showing how to use many parts of the Node framework together in a realistic app</p>