



CORSO BASE/INTERMEDIO {FRONTEND}

# Node.js

## From zero to hero

Non ha mai utilizzato **Node.js**?

È arrivato il momento di imparare come usarlo per sviluppare applicazioni, specialmente in ambito Web.

Affrontiamo le basi di Node.js come il sistema di moduli, il package manager npm e i concetti fondamentali di programmazione asincrona, la standard library e l'ecosistema di librerie. Come lo faremo? Scrivendo un servizio REST API completa...e non solo!

### Livello

Base/Intermedio

### Durata

Corso da 24 ore

### Lingua

Italiano

## Cosa faremo

Durante il corso saranno realizzati numerosi esempi pratici di casi reali e più comuni che si trovano nel magico mondo di Node.js

L'approccio al corso sarà misto ed includerà quindi **sessioni teoriche** e di **live coding**.

**Numero massimo di partecipanti: 20**

## Skill minime necessarie

- Conoscenza dei concetti di OOP
- Conoscenza dei concetti API First
- Conoscenza dei più comuni pattern di programmazione
- Conoscenza Backend Development
- Conoscenza JavaScript

## Argomenti Trattati

### Introduzione Node.js

- Runtime e non linguaggio
- Storia
- Linguaggio JavaScript - importante ripasso su tutti i costrutti/sintassi
- Accenni a typescript - tipizzazione & generics & interface
- Differenze con browser
- CLI

### NPM, Moduli e Versionamento

- Installazione tramite [NVM](#)
- NPM
  - Creare pacchetti NPM
  - NPX

- Gestione dipendenze
- SemVer
- Modules
- Run Script

## Web Services (Parte 1)

- Differenza con Java e thread base webserver
- Event loop + lib uv
- Costruzione di un webserver in nodejs - http
- Costruzione di un webserver in nodejs - fastify & comparazione con http (log, validazione etc..)
- Accenni ad express
- Accenni a NestJS

## Node.js Core

- Callback promise async
- Async pattern
- Nodejs APIs
- Event Emitter
- Stream
- Global
- Timer
- Process

## Web Services (Parte 2)

- Docker per scaricarsi un mongo db in locale
- Collegamento da node.js
- Env vars
- API esterne con axios e node fetch
- Servire pagina statica usando fs (sync vs async vs stream)
- Template Engine
- Ecosistema fastify

## CI/CD

- Test
- Docker image
- Debugging
- Best Practices

## Cosa è necessario

- Il proprio computer portatile (con installato un environment che verrà preventivamente segnalato)
- Tanta buona volontà e voglia di imparare
- Connessione ad Internet

## Cosa è incluso

- Corso pratico con live coding
- Supporto setup environment
- Slide in formato PDF
- Repository del Progetto
- Attestato di Partecipazione
- Follow Up di fine corso
- Canale Slack dedicato ai partecipanti

## Dove si svolge

### *Full-Remote*

È possibile svolgere il corso in modalità *full-remote* con gli strumenti messi a disposizione da Devmy suddividendo, se lo si desidera, il tutto in sessioni da 4h.