

# JavaScript Node.js

---

Introduzione a Node.js con JavaScript

# Roberto Serra

- Software Developer e Solution Architect presso Etiqua SRL
- 10+ anni di esperienza nello sviluppo web
- JavaScript, TypeScript, C#, Clojure, PHP, Python

## Contatti:

- <https://github.com/obiSerra/>
- <https://etiqua.it/>
- roberto.serra@unito.it

# Riferimenti e materiale

Per informazioni o domande: [roberto.serra@unito.it](mailto:roberto.serra@unito.it)

Tutto il materiale sara' disponibile su

<https://github.com/obiSerra/laboratorio-nodejs>

# Di cosa parleremo

- Introduzione a NodeJs
  - Software consigliati e installazioni
  - Due parole su JavaScript
  - Node e Npm
  - Eseguire uno script node
- Module system
  - Cosa sono i moduli e perche' usarli
  - CommonJs vs ES6
  - Filosofia dei moduli e best practice
  - Moduli predefiniti, locali e Npm

# Installazione

- Node.js: <https://nodejs.org/>
- (Consigliato) NVM
  - <https://github.com/coreybutler/nvm-windows>
  - <https://github.com/nvm-sh/nvm>
- Git
  - <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>
  - (Consigliato) <https://github.com/>
  - (Consigliato) <https://desktop.github.com/>
- Altri consigliati
  - VS Code
  - ITerm (Mac OS) / CMDer (Windows)

# Installazione 2

Per verificare l'installazione

```
$ node --version
```

*V15.13.0*

```
$ npm --version
```

*7.7.6*

# Qualche nota su JavaScript

- Funziona su diverse runtime
- Il DOM non e' parte di JS ma del runtime
- Supporta le funzioni come First-Class Citizen

# Qualcuna anche su TypeScript

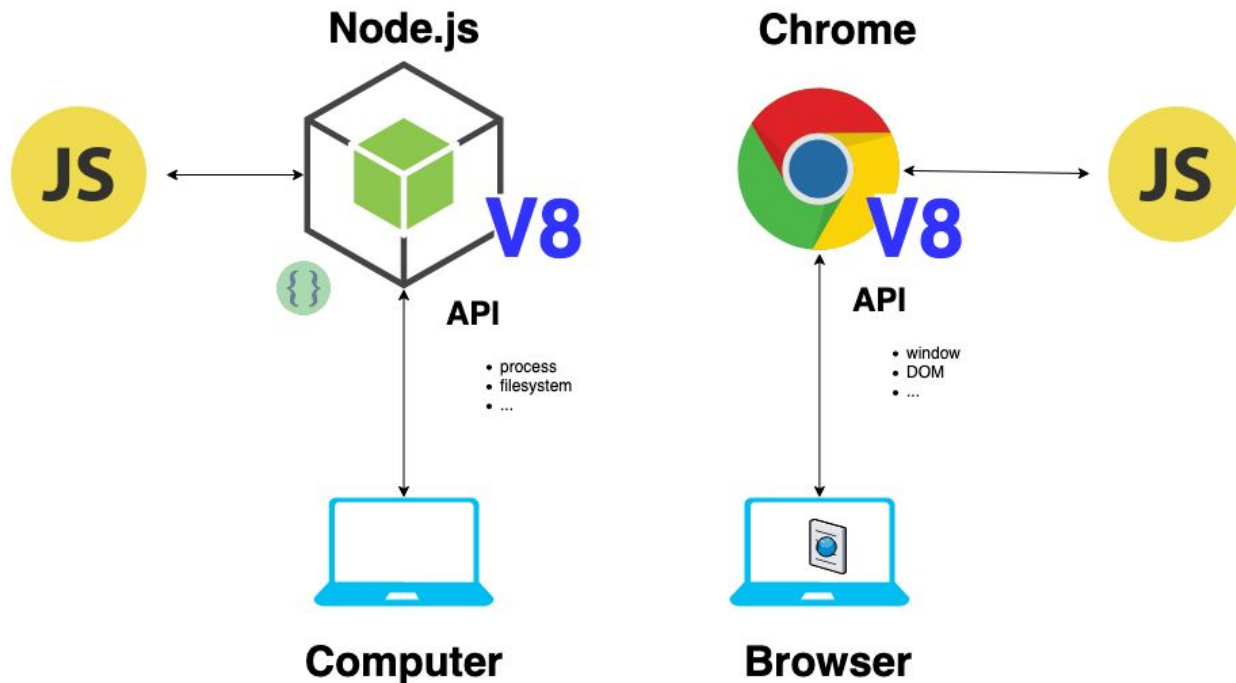
- Molto probabilmente non lo useremo...
- E' sviluppato da Microsoft
- E' un superset di JavaScript
- Ha typing statico
- Deve essere trasformato in JS per essere eseguito
- E' ottimo per grandi progetti



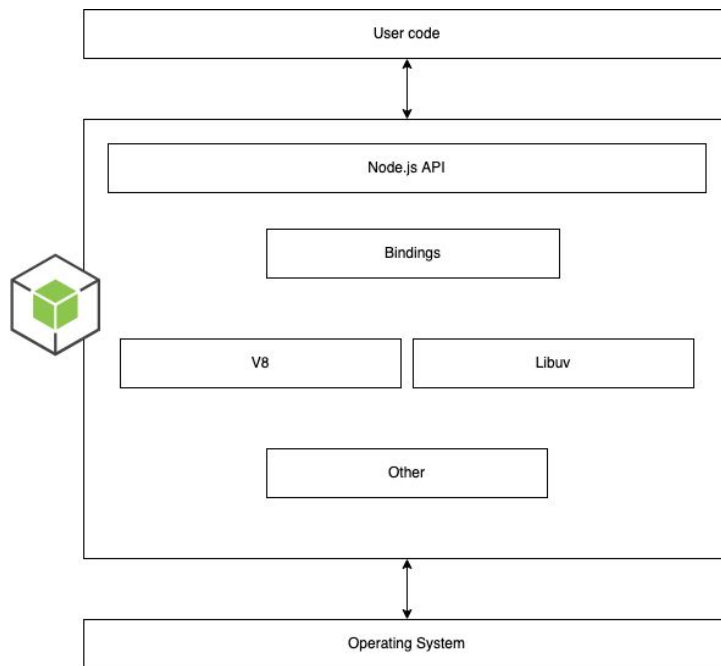
# Node.js

- Node.js (<https://nodejs.org/it/>) e' un runtime JavaScript
- Piattaforma per l'esecuzione di JavaScript
- V8 e' JavaScript Engine sviluppato da Google
- Sviluppato in C++
- Utilizza Npm come package manager
- Single Thread event loop

# Differenze tra Node.js e il browser



# Runtime di NodeJs



# Node.js 2

- Web Applications
- Tooling
- Desktop application
- IoT e altro

## Principali vantaggi:

- Semplicità' di utilizzo
- Npm
- Async programming

# Npm

- Node Package Manager
- CLI + package repository
- Utilizza json per definire i pacchetti
- Permette di aggiungere comandi

# package.json

```
{
  "name": "laboratorio-nodejs_lezione-1",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "typing": "node typing",
    "runtime": "node runtime"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "devDependencies": {
    "express": "^4.17.1"
  }
}
```

# Npm alcuni comandi utili

Inizializza un nuovo progetto

```
$ npm init
```

```
$ npm init -y
```

Aggiungere delle dipendenze

```
$ npm install --save <package1> <package-2>
```

```
$ npm install --save-dev <package1> <package-2>
```

# Npm alcuni comandi utili 2

Installare le dipendenze

```
$ npm install
```

Eseguire gli script di default

```
$ npm start | stop | build | ...
```

Eseguire uno script

```
$ npm run <script-name>
```



# Il primo progetto

1. Creiamo una nuova cartella ed entriamoci con il terminale
2. `$ npm init`
3. Apriamo la cartella con il nostro editor
4. Creiamo un file `index.js`
5. Scriviamo `console.log("hello")`
6. Possiamo eseguire il nostro script con
  - `$ node .`
  - `$ node index.js`

# I moduli in Node

- Single-responsibility-principle (SRP)
- E' possibile (e consigliato) creare tanti piccoli moduli
- Esistono alcuni moduli “di base” che permettono di interagire con l'ambiente circostante (fs, path, ...)
- Npm e' il piu' grande repository al mondo di moduli
- Attenzione a non abusare di tutto questo potere

# CommonJs

- Metodo di default di NodeJs
- Import dinamici
- Non e' lo standard di JavaScript
- require e module.exports

# ES6

- Metodo di default del frontend
- Attivabile tramite l'attributo "type" di un package.json
- Standard JavaScript
- Import statici
- import e export (default)

# Pattern per creare un modulo

- Named Exports
- Substack pattern
- Class export
- Instance export

# Facciamo una prova

- Struttura del progetto:
  - package.json
  - index.js - esegue il programma
  - src - contiene i moduli scritti per l'applicazione (almeno 1)
  - data - contiene i file di testo
- Consigli
  - Per leggere gli argomenti da linea di comando `process.argv` asda
  - Per leggere i file `findInFile`
  - <https://github.com/obiSerra/laboratorio-nodejs/tree/main/lezione-1>

# Requisiti

- Comando sul terminale `$ npm start foo` print il nome di file che contengono “foo”
- Utilizzare almeno un modulo per fare la ricerca
- Facoltativo: aggiungere un modulo per loggare le informazioni