Contents

[Class Objectives 2](#_Toc43563381)

[What You Need 2](#_Toc43563382)

[What is Node 3](#_Toc43563383)

[Resources 3](#_Toc43563384)

[Aspects of Node You Should Know 4](#_Toc43563385)

[ Asynchronous and Event Driven 4](#_Toc43563386)

[ Very Fast 4](#_Toc43563387)

[ Single Threaded but Highly Scalable 4](#_Toc43563388)

[ No Buffering 4](#_Toc43563389)

[ License 4](#_Toc43563390)

[Where to Use Node.js? 4](#_Toc43563391)

[Where Not to Use Node.js? 4](#_Toc43563392)

[Testing Node Install 5](#_Toc43563393)

[Testing Mongo Is Installed Right 5](#_Toc43563394)

Class : Learning Node For Everyone

Start Date : 06/20/2020

## Class Objectives

* Introduction
  + What You Need
  + What is Node
  + Resources
  + Contacting Instructors
  + Class structure
* Getting Started
  + Installing
  + Setting up
    - Node
    - Mongo DB
    - MySQL
    - Editors (IDE)
* Developing Code (all languages)
  + Introduction to Concepts and Tools
  + Best Practices
* How node works
* Async and Node
* DB with Node (Mongo & MySQL)
* Programming to handle errors
* Securing your code
* Modeling Data
* Using PUG
* Advanced Concepts

## What You Need

To do this class you will need a computer. A MAC or WINDOWS computer is fine. You can do NODE development on either OS with no difficulty at all. Nearly everything you see us do in Windows during the class is able to be done on a MAC laptop with no trouble. You will need at least 10GB of empty space on your drive. But I advise to have more if possible. Development takes time to learn and hone to a skill so you will have to invest time. Class time should be done on the computer you are developing your code on. Be prepared to do hands on coding and testing. You may on occasion need to share your screen but you are also welcome to handle trouble shooting in whatever way is easiest for you.

What is Node? What is this class all about?

Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux.

Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.

Node.js = Runtime Environment + JavaScript Library

# Resources

Online Node Editor / Tester

<https://www.tutorialspoint.com/execute_nodejs_online.php>

My Public Repositories

<https://github.com/coffee24By7?tab=repositories>

This Class Repo

<https://github.com/coffee24By7/FullStackNode>

Downloads

Node : <https://nodejs.org/en/download/>

VS Code : <https://code.visualstudio.com/download>

Mongo : <https://www.mongodb.com/try/download/community>

MySQL : <https://www.mysql.com/downloads/>

## Aspects of Node You Should Know

* Asynchronous and Event Driven − All APIs of Node.js library are asynchronous, that is, non-blocking. It essentially means a Node.js based server never waits for an API to return data. The server moves to the next API after calling it and a notification mechanism of Events of Node.js helps the server to get a response from the previous API call.
* Very Fast − Being built on Google Chrome's V8 JavaScript Engine, Node.js library is very fast in code execution.
* Single Threaded but Highly Scalable − Node.js uses a single threaded model with event looping. Event mechanism helps the server to respond in a non-blocking way and makes the server highly scalable as opposed to traditional servers which create limited threads to handle requests. Node.js uses a single threaded program and the same program can provide service to a much larger number of requests than traditional servers like Apache HTTP Server.
* No Buffering − Node.js applications never buffer any data. These applications simply output the data in chunks.
* License − Node.js is released under the [MIT license](https://raw.githubusercontent.com/joyent/node/v0.12.0/LICENSE).

## Where to Use Node.js?

Following are the areas where Node.js is most efficient

* I/O bound Applications
* Data Streaming Applications
* Data Intensive Real-time Applications (DIRT)
* JSON APIs based Applications
* Single Page Applications

## Where Not to Use Node.js?

It is not advisable to use Node.js for CPU intensive applications.

## Testing Node Install

Create a js file named main.js on your machine (Windows or Linux) having the following code.

/\* Eugene’s First Application In Node \*/

console.log("Hello, World!")

Now execute main.js file using Node.js interpreter to see the result -

<prompt>: *node main.js*

If everything is fine with your installation, this should produce the following result -

Hello, World!

## Testing Mongo Is Installed Right

Verify this is the correct location for your install.

Then type in the following commands:

c:

cd\

md data

cd data

md db

cd\

cd "C:\Program Files\MongoDB\Server\4.2\bin"

mongod

If everything is right, you should see a mongo screen that shows running status