Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Node.js' package ecosystem, npm, is the largest ecosystem of open source libraries in the world.

> Node js Builded after extending Chrome's V8 javascript Engine.

>V8 is Google's open source high performance java script engine.

>A JavaScript engine is a program or interpreter which executes JavaScript code. it is most commonly used in Web browsers.

>php implements over apache, java implements over JVM (java virtual machine), so same as V8 is like JVM/apache and Javascript code is like java/php languages.so if u will install V8 in any machine you can put javascript code on that machine.

> Code of V8 is written in c++ to improve performance.

> Google Chrome executes over V8 Engine.

> V.imp: a person pick 'opensource V8 Engine code' and modified it and made Node js.

**Node js is single threaded architecture because it runs over google’s v8 engine and it is single threaded. Single threaded architecture provide much better performance than one thread per request architecture.**

Some Extra thing in Node js over V8 :

1: Event Driven : [Event Emitter generate Event/handle multiple event at a time] in Node event will generate and execute (may be in queue). [Ex: Event = Button Click] so creating event and getting output is known as event driven.

2: Non Blocking I/O: we know multitasking and multithreading with Thread concept used in OS.

(Thread is used to execute multiple task at a time[single process devided into multiple thread and execute that single process in OS])-Ex:download file from internet

> only one thread can save that file at that time (because hard drive is locked by one thread), so deadlock condition may be occor ,

> So in Node js No any thread will wait of other thread to execute first, so there is no chance of dead lock , so it is Non Blocking (Nonblocking = Asyncronous)

> so it is lightweight and efficient.

======================================================================

Node js is event driven asyncronous framework, design to build scalable network application.(www is also a network).

Node js is thread based but you do not worry about deadlock condition like in other thread based languages.

Node js becomes popular because code is written in very common languages that is JAVASCRIPT.

node is best for fast response like chat application.

https://www.youtube.com/watch?v=2LvGCUe9XFs&list=LLrJweUPOm\_WflNXZWfeflsA&index=10 - END

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

because V8 engine is single threaded archetecture so node is also single threaded . (we have a main loop keep on listening)

> https://cdn-images-1.medium.com/max/1600/1\*LgvhkHtSWpv4IVx8jgc7Ag.png (image URL to understand node functionality)

<https://codeburst.io/how-node-js-single-thread-mechanism-work-understanding-event-loop-in-nodejs-230f7440b0ea>

Yes! you heard it right, Node is single threaded and it is doing magical things with this model. Some of the popular server side technology like PHP, ASP.NET, Ruby & Java Servers all follow Multi-threaded where each client request results in the instantiation of a new thread or even a process, but Node. js, requests are run on the same thread with even shared resources. All Node JS applications uses “**Single Threaded Event Loop Model**” architecture to handle multiple concurrent clients. So Yes NodeJS is single threaded, but this is a half truth, actually it is **event-driven and single-threaded with background workers.**The main event loop is single-threaded but most of the I/O works run on separate threads, because the I/O APIs in Node.js are asynchronous/non-blocking by design, in order to accommodate the event loop.

npm install nodemon --g

What Can Node.js Do?

Node.js can generate dynamic page content

Node.js can create, open, read, write, delete, and close files on the server

Node.js can collect form data

Node.js can add, delete, modify data in your database

What is a Node.js File?

Node.js files contain tasks that will be executed on certain events

A typical event is someone trying to access a port on the server

Node.js files must be initiated on the server before having any effect

Node.js files have extension ".js"

==============================================================================

What is a Module in Node.js?

Consider modules to be the same as JavaScript libraries.

A set of functions you want to include in your application.

Common use for the File System module:

* Read files
* Create files
* Update files
* Delete files
* Rename files

==============================================

All requests are run in single thread with even shared resources.

All Node JS applications uses “Single Threaded Event Loop Model” architecture.

NodeJS is single threaded, but this is a half truth, actually it is event-driven and single-threaded with background workers. The main event loop is single-threaded but most of the I/O works run on separate threads, because the I/O APIs in Node.js are asynchronous/non-blocking by design, in order to accommodate the event loop.

imp: see archetecture of normal and node : https://www.journaldev.com/7462/node-js-architecture-single-threaded-event-loop

++++++++++++++++CS GREEKS 18 VIDEOS+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

V1: you can execute node code in CMD directly, type node then enter and console.log("abcd");

V2: undefined function is : no parameter and no return value.

ananomous function : function with no name. var prints = function () {}

v3 : Database, server, request > there are 3 things in node to handle request. use send request to server and server pass request to database(may take 5 second to handle request). will take next request withoout completing previous request

v4:in node everything is a refrence , if you are coping a object into new object, it will not copy, actually it will keep the refrence of that object.so if you will change value in new object it will be change in original/old object.

V5: everything in js is object, 'this' keyword inside a object points the same object, and if this is not inside the object(like in a function), so this will point a global 'this'.

V6: Prototyping: we can add method and properties into the class and objects.

Note: ECMAScript is a Standard for a scripting languages. Languages like Javascript are based on the ECMAScript standard. ECMA Standard is based on several originating technologies, the most well known being JavaScript (Netscape) and JScript (Microsoft). ECMA means European Computer Manufacturer's Association.

You can use the protype property after creating a object using 'new' keyword.

V7: module.exports.a = name; and

var m = require('./movie');

m.a();

to export module or js file.

V8: same above code in V7 can be written in advanced form, see video.

module.exports={

name:function(){console.log("Don");}

}

and

var m = require('./movie');

m.name();

V9:Modules share common memory, thay shere the memory.(called default state of Node)

all chield module change value in common parent module.

so if you export a module / include a module in a js file and set the value of that module, after that you include same module in other file again, so you will get that same value which you have set before while including that module in second file.because it have the original copy of that module.

V10 : Create Module without sharing a common memory: (using object factory)

module.export = function(){ return name:""}

correct video is : https://www.youtube.com/watch?v=I50lByKfgA4

V11:Core modules = pre defined modules.path related concept , repeated output of node in set interval.