(*) → After installing Node.js, we can use it to basically use the extended V8 version to run JS scripts on computer.
↳ then suddenly can access these features as they don't run on browser, but directly executed through Node.js runtime.

Saathi

Date __/__/____

(1) Section 1 : Introduction

What is Node.js ?

Node.js is a javascript runtime.

js - Programming language used in the browser to manipulate DOM (ie page loaded in browser)

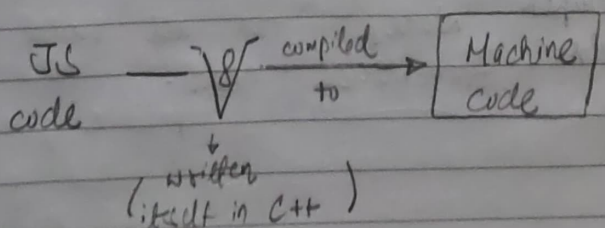
JS is a language that runs in the browser, and allows us to interact with the page after it was loaded - ∴ essential for building interactive UIs.

Node.js is sort of different version of JS.
↳ built on JS - adds some features & has some less features than original JS's DOM manip
↳ Puts JS into a different environment.
Node allows us to run JS code on server.

Allows us to run JS, not only on browser, but like a normal programming language - like normal programs on computer.

∴ Node → used to run JS code out of the browser
Node uses a V8 engine
↳ JS engine made by Google that runs JS on browser.

What V8 does in the browser ?



Node.js takes the V8 codebase (written in C++), and adds certain features like - working with local file system [opening / reading / deleting]. Such features are not possible in browser. (for security purposes)
Node.js adds these features to the "vanilla" V8 engine.

Page No.

(*)

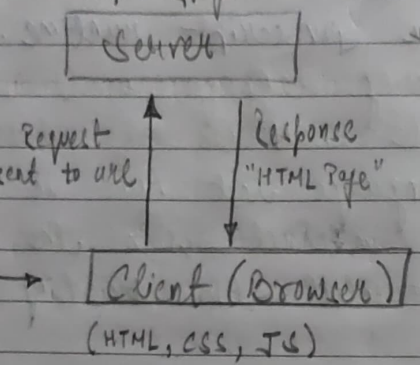
REPL → special input mode where we can execute code, and have it parsed by node on the fly.

Saathi

Date ___/___/___

Role and Usage →

Server → some computer running on the internet, which for the IP associated with that domain (automatically resolved) http://myfoge.com



On the server, we execute some code that does something with incoming request & returns a response (HTML page with CSS & JS on browser side)

On the server, we typically do tasks that we can't or don't want to do from inside the Browser for performance or security reasons.

ex → connect to databases (fetch and store data)

Node.js
- user-authentication (cbv done on a place user can't access)
- input validation
- business logic (things that take too much time to run for fast UX)

∴ We'll use Node.js to write code on the server that returns data our client can work with.

Node.js is not limited to the server!

Node

- utility scripts
- build tools

React/Angular/Vue - individually use Node.js for all build process these frameworks need.

Role

Run Server -

create server and listen to incoming requests.

- we don't just write code running on the server, but we write server ourself.

In PHP, we use Apache/Nginx to run server that listen to requests, and execute PHP code.

Responses

To return responses (rendered HTML, Icon, files ...)

- both code that takes incoming request & routes them to other code.