**Day 1 : 24 Aug 2024**

Node JS

Html -🡪 web pages

Css or bootstrap 🡪 apply formatting style

JavaScript : ES5 -🡪 use to do validation on client side.

<http://www.google.com> URL

req(http/https)------🡪

Client Server

🡨-Res(http/https)

Html/html5🡪

CSS/css3-🡪

Bootstrap

JS (JavaScript )

To do validation on client side.

Using JavaScript we can do programming on web page without server.

Read, write and Update DOM.

jQuery

Angular

React Js

Vue JS

Client Side technologies

Backend technologies

Java (Servlet, JSP,EJB), Spring framework / spring boot

Asp.net

Php

Python with Django

Node JS: Node JS is not a library or not tool or not a framework. Node JS is run time environment for JavaScript program or library or framework.

Before Node JS JavaScript is known as client side scripting language. To run this program or js program we need browser environment. But after node js we can run JavaScript program outside browser environment. After node js we can say JavaScript can be client side as well as server side scripting language.

MEAN Stack Mongo DB/ MySQL Express JS Angular Node JS

MERN Stack Mongo DB/ MySQL Express JS React JS Node JS

MEVN Stack Mongo DB/ MySQL Express JS Vue Node JS

V8 Engine

JsVM

Node JS provided lot of modules. Which help to do all server side programming.

Like

1. Handling IO operation using JS
2. Handling OS resources
3. Accessing the RDBMS as well as No SQL Database.
4. Networking

Etc

Open the command prompt

First check node --version

Write the node and hit enter key : it open REPL terminal Read Eval Print Loop.

Client Side JavaScript provide BOM and DOM Hierarchy

Brower Object Model

Document Object Model

Object -🡪 property (variable)

Behaviour (function / methods)

Object ---🡪 property

Behaviour

Object -🡪 property

Behaviour

Object ---🡪

document.write(“Welcome to Client side JS”)

window.document.write(“Welcome to JS”);

In Node JS program no BOM and DOM.

Node JS provided one of the pre defined global object. ie console.log(“Welcome to Node JS”);

Synchronous communication

Statement level

console.log(“first statement1”);

console.log(“second statement2”);

console.log(“third statement3”);

function call

fun1()

fun2()

fun3();

Asynchronous communication

Statement level

console.log(“first statement1”);

using some function we make this statement as asynchronous console.log(“second statement2”);

console.log(“third statement3”);

function call as asynchronous

setTimeout()

setInterval()

clearTimeout();

fun1() client 1

fun2() client 2

fun3(); client 3

AJAX : Asynchronous JavaScript and XML

XMLHttpReqeust or ActiveXObject

Event loop

Running node js program using script file.

Callback : passing the function name or function body or function itself to another function as a parameter is known as callback.

Synchronous : the logic or statement are executed in sequence. When the current statement or function call or request depends upon previous statement execution.

Asynchronous : the logic are executed independently without blocking the primary program or statement or function call execution.

Tot create custom asynchronous functions or statement

Client side as well as server side JS provide three function ie

setTimeout

setInterval

clearInterval

Node JS Modules Node js modules is like a package or namespace. Using modules we can encapsulates related functionality. Allowing to organize and reuse our code effectively. Node js provide modules concept help which help to break down our application into smaller parts. Each modules are divided into different files and folder and those file we connect using import (ES6 style)/require (ES5 style) and exports

Mainly modules are divided into 3 types.

1. Core modules (by default present with node js run time environment)
2. User defined modules
3. External or third party modules

Syntax to use core module or user defined modules or external modules

let/var referenceName = require(“modulename”); it is use to load the modules.

Core modules

OS module : if we want to get OS related functionality we can use os modules.

Fs module : fs : file system : provide files related operation which we can do synchronous as well as asynchronously. Ie read, write, copy and append.

In node JS or JS we can create object using 3 ways

1. Literal style
2. Using function style in ES5 style
3. Using class style from ES6 onward

What is Node JS