Node.js and Express

1. What is Node.js?
   1. Used to programme desktop apps on server. It’s speciality is that it is written in JS so no need to learn another language like php etc.
   2. Separate processing of code execution from the client side. No need to run development files on a browser, instead we can create a dedicated environment for it.
   3. REPL -> Read Evaluation Print Loops. The same as python console or JS Console in chrome.
2. Native node modules usage
   1. Require is same as import. Here we store the libraries as objects or constants. Go through the documentation and see for yourself. Example used was of FileSystem or fs module.
3. NPM manager
   1. External libraries made by someone else which we can incorporate in our project.
   2. It is very similar to pip or conda for python.
   3. Npm create a file package.json which creates JS object like list of all the packages that we install in the current project.
4. [Difference between node and express.](https://www.geeksforgeeks.org/node-js-vs-express-js/) Do read. <3
5. What is express?  
   As jQuery is to js , so is, express to node js. A framework that provides broad features for building web and mobile apps. It is built on top of Node js so we can manage servers and do routing.
6. Server basics – creation, requests, response, parsing and data type, simple calculations.
7. [MUST READ FOR EXPRESS](https://www.freecodecamp.org/news/express-explained-with-examples-installation-routing-middleware-and-more/)
8. [Different type of Requests in HTTP which NodeJS uses to communicate between server and client.](https://www.geeksforgeeks.org/different-kinds-of-http-requests/) [Sample of these requests](https://expressjs.com/en/starter/basic-routing.html)
9. Some code to study:

// Importing the libraries/frameworks and storing them as variables.

const express = require('express');

// body-parser is used to parse through the request sent to us by the website at user end.

const bodyParser = require("body-parser");

// making an object of express app.

const app = express();

// Use body-parser to parse.

app.use(bodyParser.urlencoded({extended: true}));

// The first parameter "/" specifies the route at which we must receive any request objects and the second is a call-back function which tells the server what to do when that request happens.

// This function is used to do something with the request and send a response to a browser and it renders it on screen.

app.get("/", function(request, response){

    // console.log(request);

// \_\_dirname is a predefined constant of pwd in node

    response.sendFile(\_\_dirname +"/index.html");

// Send file function is used to send HTML files as response.

});

// Dealing with the data in form.

app.post("/", function(req,res){

var num1 = Number(req.body.num1); // WE need to convert the input received into correct datatype because the request is all strings.

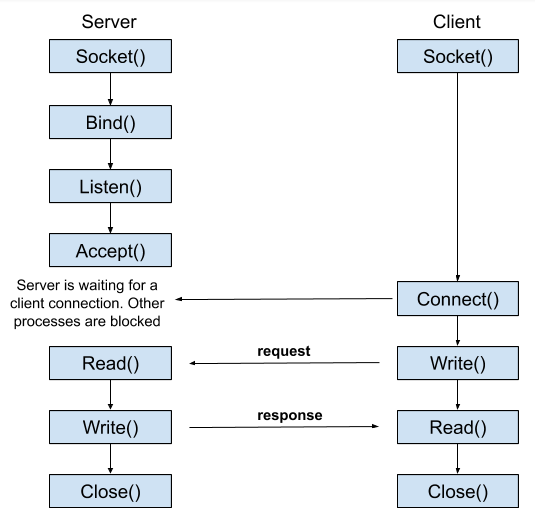
    var num2 = Number(req.body.num2);

    var result = num1+num2;

    res.send("The sum is "+ result);

});

app.listen(3000,function(){console.log(“server running”);) // [Refer this](https://stackoverflow.com/questions/70384927/what-does-the-listen-method-in-express-look-like)

1. Processing POST REQUESTS:
   1. The method in HTML form is usually POST, which means it is sending the data to somewhere usually current page route.
   2. The POST request is routed to the same page, but since it is a different kind of request, we receive data from this reqest using body-parser and pick up the required information with the following command - var num1 = Number(req.body.num1).
   3. Then just like sending response to any other request we would revert back.
2. 

**Section 20 lil bit**

1. Making get requests with the Node HTTPS module:
   1. CLIENT 🡨🡪 OUR SERVER
   2. OUR SERVER 🡨🡪API SERVER
   3. OUR SERVER 🡨🡪 CLIENT SERVER
2. // We can have only one send method in an one app.get method. THIS IS LIKE THE RETURN STATEMENT IN FUNCTIONS. We can have multiple res.write() methods.