Node.js

-a JS platform that executes JavaScript code outside a web browser

Express.js

- is one of the most depended-upon nodejs packages

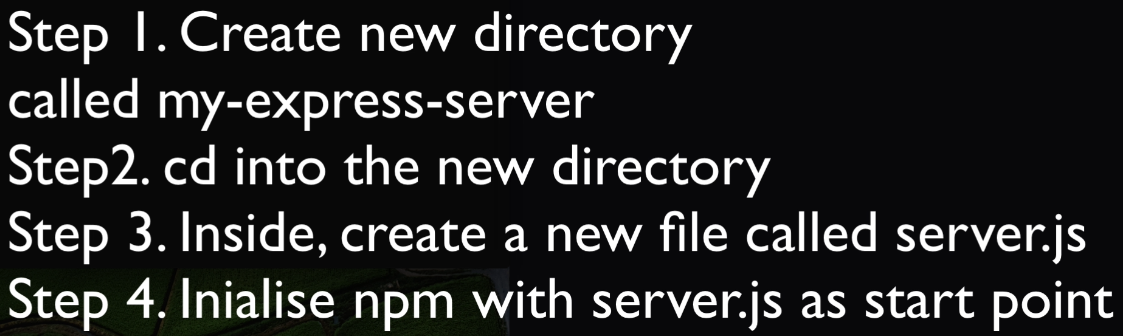
- is a nodejs framework that serves nodejs as jquery does to java script

- designed to make you write less repetitive code

atom .

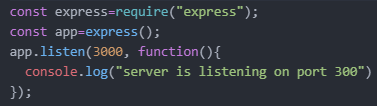
- will display the directory inside atom

The steps you go through every time you create a server



Step 4: type npm init

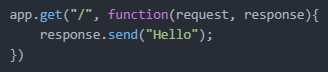
Step 5. npm install express

Step 6: write  in server.js

-https://expressjs.com/en/starter/hello-world.html

-our server is listening on port 3000 for any browsers that are trying to get in touch with it.

Step 7: Add this to your server.js file

- 

- this will display hello on your browser. The response sent to your browser will be hello

- if we console.log(request): we will see the info about the request.

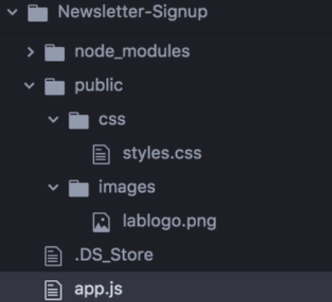
Step 8: type npm install -g nodemon

Step 9: type nodemon server.js

Step 10: Add this to your server file



* It specifies a static folder named public for your static files to consider as root
  + Classic files means files like images and css you add to your page.
* this will allow you to add static images and CSS to your page through a folder called public in your directory. Move your css file and public file into this.



* the public folder will be considered as the root for static files and we can use it to access static files.

How to avoid typing node server.js every time there is a change in your source code?

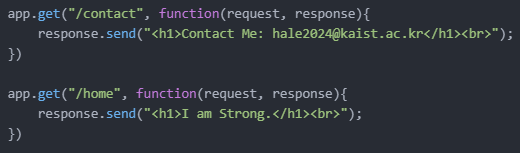
1st type npm install -g nodemon

2nd type nodemon server.js

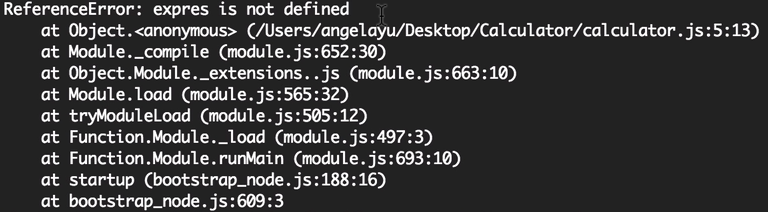
* it will look for changes in your terminal and restart your server

localhost:3000/

* Is the home route of your web page.
* When we load a page our browser sends a get request to the server to get some data for the url location, the server then returns a page



* you can add other routes like the above.



* If there are errors in our code, our terminal will tell us when we nodemon or node server.js



* \_\_dirname
  + Means the directory path of the file
* sendFile
  + allows us to send a whole file as a response

You can check the post message sent by:

-Opening the site in mozilla+Inspect element+Network+All

- 

HTTP Status Code meanings

-100 means hold on sth is going to happen

-200 means here you go. It is usually a successful request code

-300 means go away there is some security involved

-400 means you screwed up

-500 means the browser screwed up

-You can check more on: <https://en.wikipedia.org/wiki/List_of_HTTP_status_codes>



-for receiving the post message sent

Receiving post Messages and parsing them properly.

Step 1: write this in calculator.js (server file) for receiving the post message & giving out a response



Step 2: write “npm install body-parser” in terminal

Step 3: write this in calculator.js (server file)



Step 4 write this in calculator.js (server file)



-bodyParser will allow us to parse the HTTP request & to tap into request.body

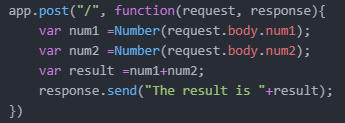
-u could console.log the value if you want

-urlencoded: - is used whenever you're trying to grab the information that gets

posted to your server from an HTML form

-extended:true:- allows us to post nested objects.

Step 5: Tap into request.body.num1 inside the app.post



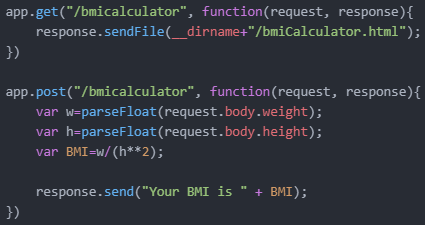
app.get

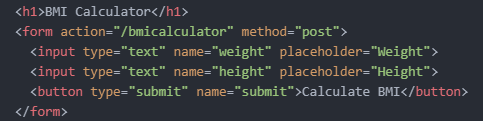
-used to show the page when the browser requests for the page.

app.post

-used to give a response to the post message sent to the server.

Creating a BMI calculator

-  in server file

-  in html file