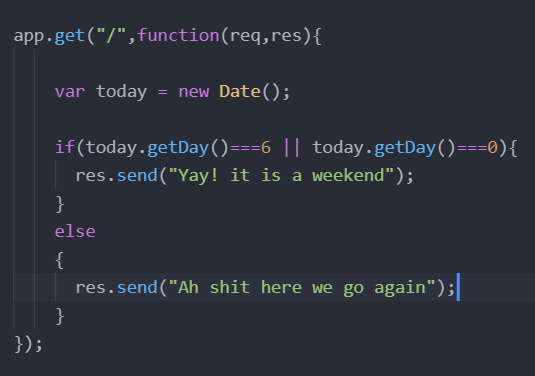
**TODOLIST**

Get date method: belongs to JS in built library



**Sunday-Saturday: 0-6**

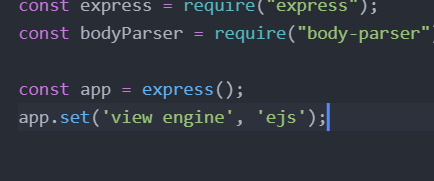
**Templating**

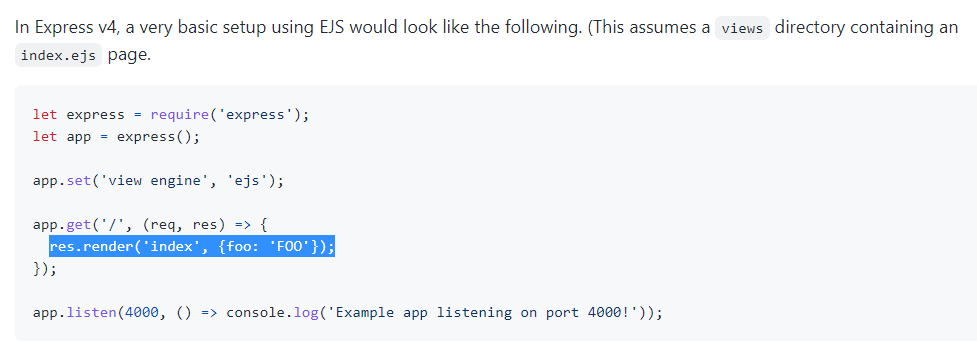
A method to avoid writing a lot of code over and over again just like the functions in c++.

<https://ejs.co/> : EJS is a simple templating language that lets you generate HTML markup with plain JavaScript. No religiousness about how to organize things. No reinvention of iteration and control-flow. It's just plain JavaScript.

**EJS**

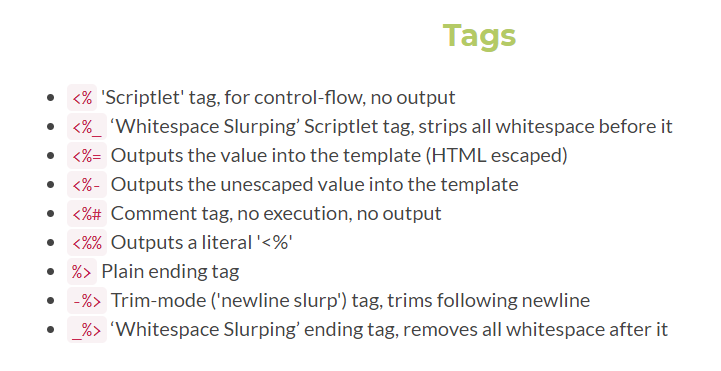
It is mandatory to put ejs view engine below app because if you would put it above the app declaration then that means that you have set the view engine before declaring the app. <https://github.com/mde/ejs/wiki/Using-EJS-with-Express> page to guid about setting up the ejs.





It is advisable to use different variable names for key: value pair when we are passing the values of the ejs modules as it will get really confusing. Therefore it is highly recommended to use different modules in the server and the ejs file.

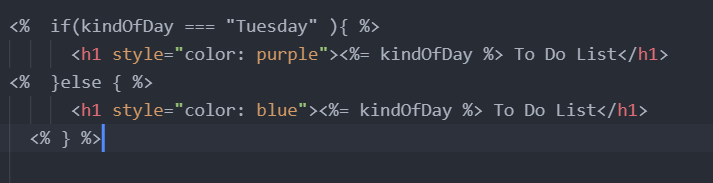




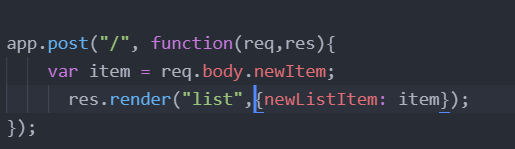
These are different tags for the user usage basis.

The first tag is a control flow tag that doesn’t support output rather only the control flow. The major thing about this is that it used when we want to modify our code when there is a situation to change our code for some instance.

This tag <% code %> is for each of the line that operates on the control flow of the program.



<https://stackoverflow.com/questions/3552461/how-to-format-a-javascript-date> a method to print day time without switch statements



This is the error we will get if we simply just trigger the newListItem as the item and post it to the home route. Therefore this is not a feasible method to do it.

* We had few bugs in the code as we moved on with the process of obtaining information with this entire process of sending data to our server and the server response of our data.
* We need to declare all the EJS of the list.ejs file in one place where we are rendering it. But when we want it on different places then we do it using the redirect method.



Here we can see that we created the items array which is initialized with three items in the beginning and as we will enter new items using the input boz oon the page, we will catch it here on our server using the post route and then we will add that new listed itme into the items array and in return the for loop in the ejs file will create a simultaneous li(list items) until the for loop terminates.

All this process is made easy using the templating method.

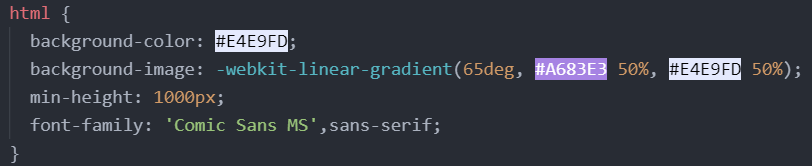
SCOPE: is basically the accessibility of a variable/costant/let. Like as when we declare a variable inside a function but we are trying to access it outside the function but as we know scope is restricted to that particular function hence it cannot be accessed outside the function

* **Use let/const not var. using var is very rare.**

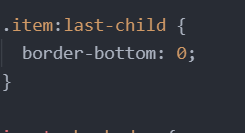
CSS inclusion

* When we try to link the file of css to our list.ejs it simply don’t work. This happened because we are not dealing with static websites anymore so instead we need to tell our express to use a folder called PUBLIC and use whatever inside of it. Be it a styles file or a simple html.
* We put everything inside that public folder. Our css folder and that will be included inside the lsit.ejs file using the link tag passing the href as 

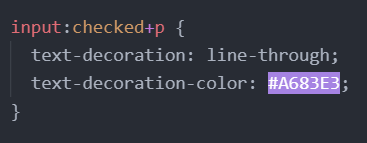
1. Creating a gradient inclined line



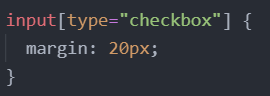
1. Targeting the last child of the class item



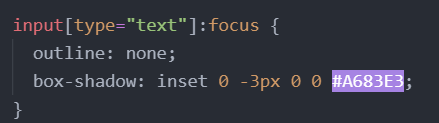
1. Creating a crossed line through the paragraph tags



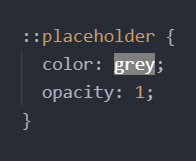
1. Input of the type checkbox



1. Input of type text with focus



1. Pseudo selector

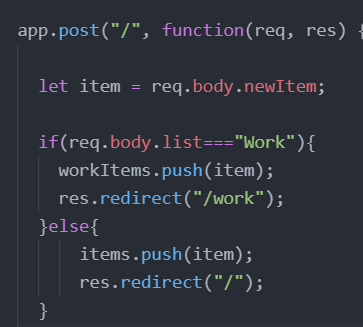
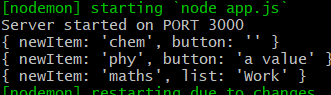
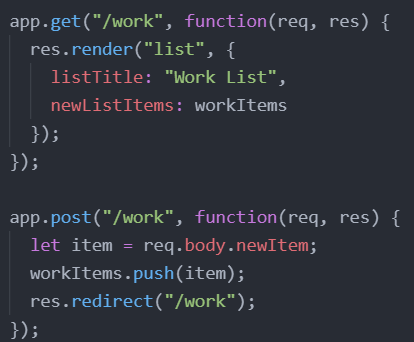


1. Remove the autocomplete from the input bar



What we have done so far is that creating a file that makes up the /work route to get to the route page rather adding the items in the previous array of items. We wanted to add into the newly created array workItems.

We first console logged it about what the body was returning using the console.log(req.body) in the post request that we have created. The next thing is we provided a name to the submit button as the list button. And then created a dynamic tag of the ejs into the list.ejs file. And in the post request we used the if else statements in which it states that when the input button list as req,body.list has the value of WORK on that case redirect the post route to the /work route otherwise the home route/

<https://www.w3schools.com/html/html_symbols.asp> to implement different signs

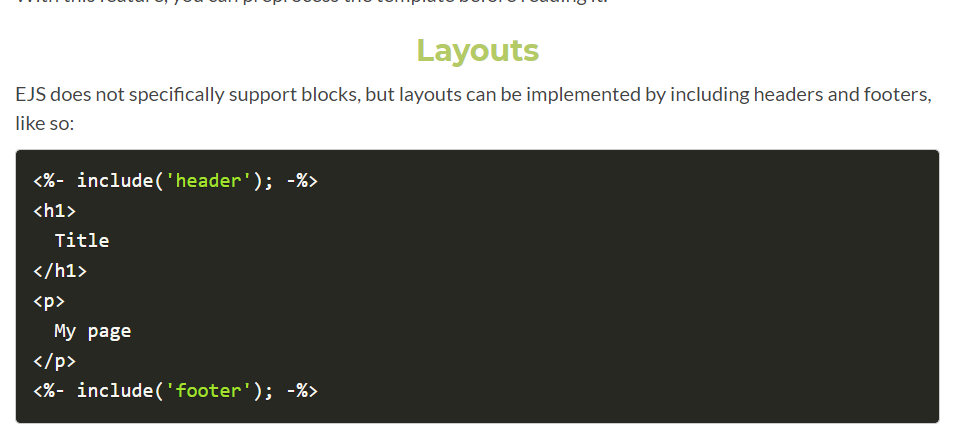
**LAYOUTS in EJS**

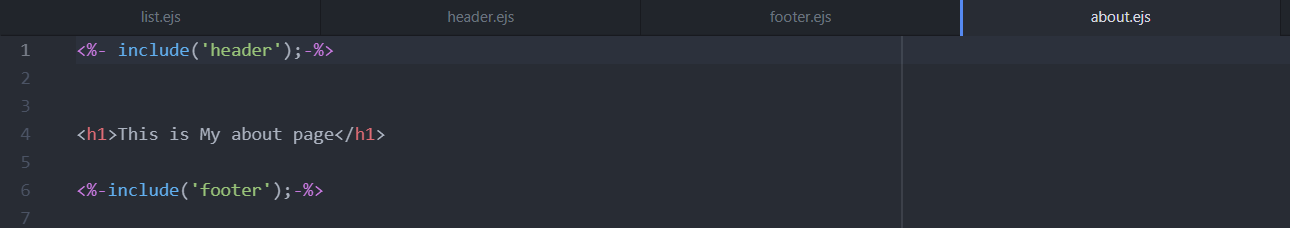
**Use?**

When we are dealing with a large website (So many pages) we need to create multiple pages write their html and then incorporate the CSS the way we want it to be. Layout in EJS helps in creating an their incorporation into the server much easier rather than sending the file each time we want a particular page. So it is better to use the layout rather creating single page and sending them each time.

**How to implement?**

1. Create two ejs files named header and footer and in those file put the header and the footer of the parent html file.
2. The use this to incorporate other pages into the same files onto the server

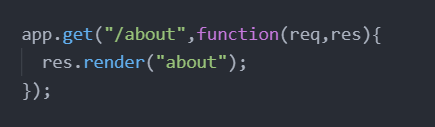




My about page using the header and footer and now comes the sending part of this about page.

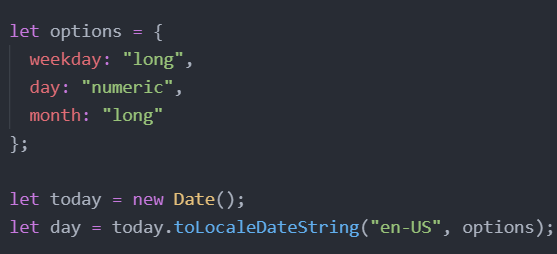
**How to incorporate into the server?**

Just create the function and then render the respective file.

****

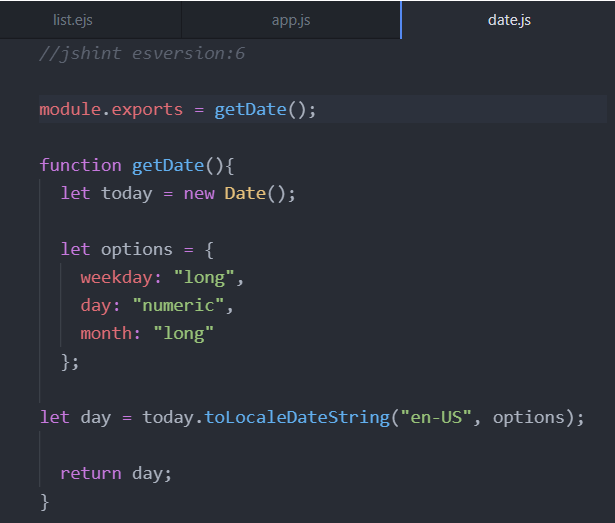
**Also this page is completely different page from the to do list page which is basically has checkboxes and other options to add. The about page doesn’t have it. It is as if like all the page of the website is being split into the different sections rather creating a whole bunch of the file into the same page.**

This block of code doesn’t belong in here as it is a calculation of the date, so what we can do is. We can simply create a function inside another file called as date and then use that custom made js file as a module. And export it into the server file in app.js.

****

**HOW?**

1. **Create the file of your requirement and do the return values and everything that is required.**



**Incorporate?**

1. **Simply require the module as we used to do with the other modules like express and body parser.**
2. **Then we let a variable called day and assign it the value of the date const that module we required earlier.**

****

**NOTE:**  There were points when I declared them as a function a module is not a function so we cannot pass it the value rather assing it the value of the function as we did. You can see that above.

<https://nodejs.org/api/modules.html> docs on MODULES

How to add multiple modules instead of just one?



REFACTORING THE CODE OF MODULE

 More refactoring the code

The reason we assigned a const to the array is because we can protect the parent array t be passed into another array and creating a new version of it.

We can protect the array from modification but we cannot protect the things inside of that array.

