#### Aula 01 (27/06/2016)

Node.js: interpretador de javascript.

Terminal: interface textual com o sistema operacional.

Navegação no Console:

CD - navega entre as pastas;

PWD - mostra em que pasta se encontra;

Control + L - limpa o terminal;

Control + D - sai do node;

LS - mostra quais arquivos estão na pasta;

open . - abre um arquivo.

node + arquivo.js - executa o arquivo.

mkdir - cria uma pasta.

#### Aula 02 (28/06/2016)

Muitos exercícios para fortalecer a lógica de programação.

Require: puxa funções que estão em outro lugar.

Constante: se comporta como uma variável, mas só pode ser definido uma vez.

Código para criar uma porta:

var http = require('http');

const PORT=8080;

function handleRequest(request, response) {

response.end('UHUUUU!! FUNCIONA!! YEAAHHH!!');

}

var serve = http.createServer(handleRequest);

server.listen(PORT, function(){

console.log("Servidor ligado, escutando porta", PORT);

});

Endereço local: 127.0.0.1..8080

#### Aula 03 (29/06/2016)

De manhã: teatro explicando como funciona a relação entre servidor e browser.

De tarde: instalar express e dinâmica de duplas para fazer o exercício do site de yoga usando ejs.

Paginas estáticas usam .html, página dinâmicas são ".ejs".

**Resolução explicativa do exercício da página de Yoga**

express js

<http://expressjs.com/pt-br/starter/installing.html>

<https://scotch.io/tutorials/use-ejs-to-template-your-node-application>

1 – Criar a pasta “Site”

2 – Criar as sub-pastas “Public”(para arquivos estáticos) e “Views” (para arquivos dinâmicos)

3 – Colocar todos os arquivos que o professor enviou na sub-pasta “Public”

4 – Mover somente o arquivo “Instructors.html” para a sub-pasta “Views” (mais pra frente este arquivo deverá ser substituído pela versão com extesão “.ejs”)

5 – Criar a pasta “myapp” dentro da pasta “js” (que foi copiada para a sub-pasta “Public” junto com os arquivos do professor)

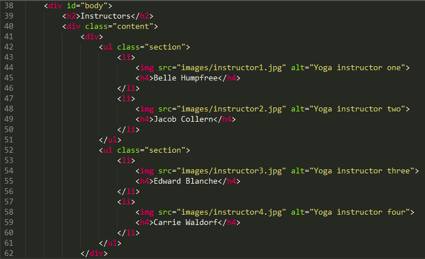
6 – HORA DE INSTALAR OS NPM!

* Abra o terminal/prompt de comando
* Vá até a pasta desejada
  + Neste caso, vamos repetir estas instalações para as pastas “myapp”, “site” e “views”
  + No windows, para acessar uma página digite “cd (nome da pasta)” + enter
  + Se você não sabe quais são as pastas, você pode digitar “dir” + enter e todas as pastas serão listadas
* npm init
* npm install express --save
* npm install ejs --save
* npm nodemon --save
  + lembrando que tem um espaço antes de “--save”!

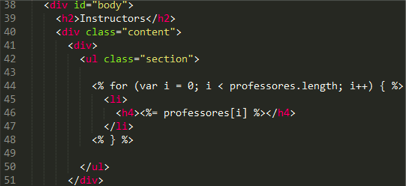
7 – No Sublime, crie o código com javascript abaixo e salve-o como “app.js” na pasta “site”

8 – No Sublime, abra o arquivo “instructors.html” e substitua o conteúdo da div de instrutores pelo código javascript enviado pelo professor, salve-o como “instructors.ejs” e delete aa versão “instructors.html”.

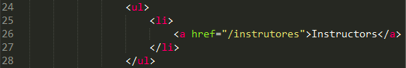
De...



Para...



9 – No Sublime, edite o endereço do menu de todas as páginas “.html” de “instructors.html” para “/instrutores” Dessa forma, os links dessas páginas irão funcionar quando você for executar o código ☺



10 – No terminal/prompt de comando, ir até a pasta “site” e execute o arquivo digitando “node app.js”

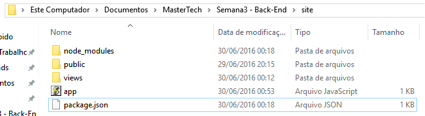
11 – Abra o browser (Chrome, por exemplo) e digite no endereço “localhost:3000” + enter

* Navegue pelas aba “Classes” > “Instructors” para verificar se o código está ok
* Faça o teste digitando “localhost:3000/instrutores”

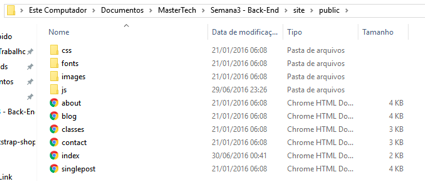
12 – teoricamente é para funcionar ☺

13 – Para finalizar a execução no terminal/prompt de comando, abra ele e aperte as teclas “Ctrl” + “C” e feche-o.

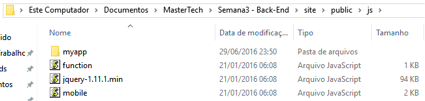
No final, a sua pasta site ficará assim:



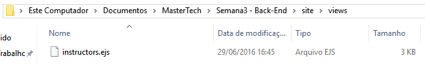
... a sub-pasta Public ficará assim...



.... a pasta “js” em “public” ficará assim...



...a sub-pasta “views” ficará assim...



AGRADECIMENTOS MAIS QUE ESPECIAIS PARA

O VAGNER, PARA O JOHN E PARA A NINA!

#### Aula 04 (30/01/2016)

**Passo a passo para criar um site:**

1 - Cria uma pasta global para o projeto. Todo e qualquer comando de node e ejs será feito na pasta raiz. No terminal, usa: mkdir nome-do-projeto

2 - Abre a pasta. No terminal, usa: "cd nome-do-projeto"

3 - Rodar o npm para que possa usar o npm no projeto. No terminal, usa: "npm init ." e dá muitos enters.

4 - Configura as dependências. Ou seja, instala o express, que permite criar aplicativos web. No terminal, confira se está na pasta raiz. Se sim, instale usando "npm install express --save". Se não, vai pra pasta raíz.

5 - No sublime, cria um novo arquivo e salva como "app.js".

6 - Abre o link "[http://expressjs.com/en/starter/hello-world.html"](http://expressjs.com/en/starter/hello-world.html%22e) copia o código e cola no arquivo app.js. Essa é a estrutura básica de qualquer projeto web comum, você pode editar a rota/caminho das páginas do site chamando a função:

app.get('/nomedapagina', function (req,res) {

res.render('arquivo.html')

});

7 - Dá um start no servidor. No terminal, usa "node app.js".

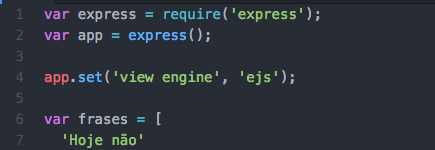
8 - No browser, escreve: "localhost:3000" para acessar seu arquivo.

9 - No sublime, cria um novo arquivo javascript e salva como "nome.js". Nele escreve o algoritmo que vai usar. Depois copia ele para o arquivo "app.js". Fica assim:



10 - Instala o EJS na pasta raíz. No terminal, usa: "npm install ejs --save".

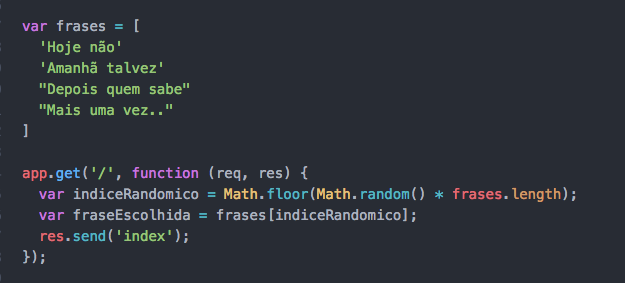
11 - No sublime, no arquivo app.js configura o ejs, usando "app.set('view engine', 'ejs')". Fica assim:



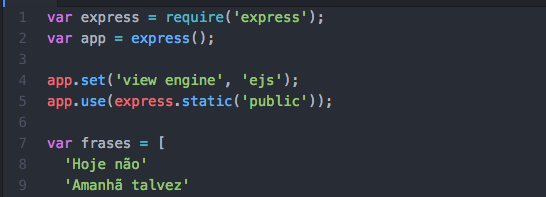
12 - Na pasta raiz, cria uma pasta chamada "views" e uma chamada "public". Em views, salvará os arquivos dinâmicos (ejs). Em public, os estáticos (html, css, fonte, javascript, imagens).

13 - No sublime, cria um novo arquivo chamado "index.ejs" e salva na pasta views.

14 - No app.js, edita a rota para chamar "index" usando "res.send". Fica assim:

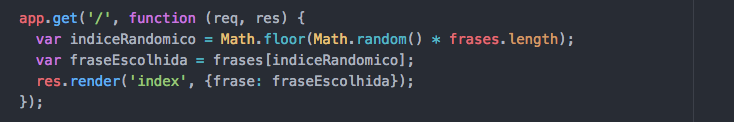


15 - No app.js, cola o código: app.use(express.static('public')); Fica assim:

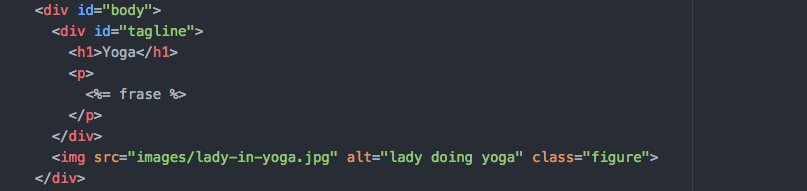


16 - Abre o index.ejs e edita o caminho entre ele e o app.js. Considere que a manipulação de dados principais, tudo que é lógica, deve ficar no "app.js". Só as lógicas de apresentação de conteúdo, as respostas, ficam nos arquivos ejs.

Exemplo: Em app.js:



Em index.ejs:



--

Action: rota por qual o formulário faz um request.

Método ".post" : determina que as informações vão só no corpo do conteúdo, não na url. Ou seja, formulário sempre usa "app.post".

Formulários

1 - instala dependência no terminal usando: $ npm install body-parser --save

2 - altera no app.js para "req.body.nome"

no contato.ejs:

#### Aula 05 (01/07/2016)

Cada grupo teve que desenvolver a essência do seu projeto, usando o que foi aprendido durante a semana e algo a mais (ex: integrar como a api do google).

**Anotações do Codeschool**

JavaScript is a programming language used to add interactivity to a web page. jQuery simplifies JavaScript syntax and makes it easier to build interactive web pages that work across multiple browsers.

jQuery enables us to do three main things on a web page:

1. Events. Respond to user interactions.
2. DOM Manipulation. Modify HTML elements on the page.
3. Effects. Add animations.