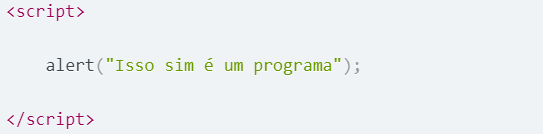
**FRONT END**

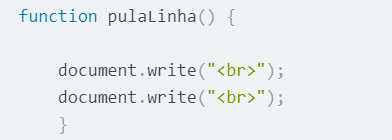
**LÓGIGA DE PROGRAMAÇÃO I**

A partir da versão 55 do chrome e versões mais recentes de alguns navegadores já é detectado automaticamente o encoding dos arquivos, a tag <meta charset=”UTF-8”>. No entanto para uma boa prática é sempre recomendado usar, pois alguns navegadores ainda pode não detectar automaticamente.

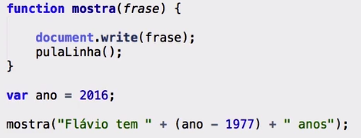
Todo texto em javascript deve estar entre aspas “exemplo”, e para que o código seja identificado como javascript devo sempre identificar ele com a tag <script> texto </script>. Obs: Toda linha js termino ela com o ponto e vírgula e para passar o texto para algum comando eu coloco ele entre parênteses, exemplo:



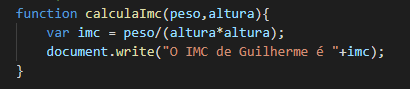
Para criar uma função eu uso o function da seguinte maneira ‘function nomeFuncao() {}’, o nome da função sempre será um verbo, além disso devo sempre identar meu código para melhor visualização e todas funções vem sempre no início da tag <script> Exemplo:



Podemos também passar parâmetros para a função da seguinte maneira, ‘function mostra(frase){}’ onde passei uma variável como parâmetro, então se em algum momento eu utilizar essa função passando algum tipo de parâmetro ele irá fazer frase=”algo que passei como parâmetro” e irá retornar na função, exemplo:

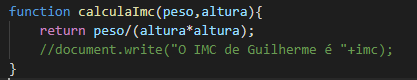
  
Na tela ele vai mostrar frase= “Flávio tem “ + (ano – 1977) + “anos”) e irá exibir na tela o valor disso, que será *Flávio tem 44 anos*.

Também posso passar mais de um parâmetro caso queira fazer algum cálculo, por exemplo; function calculoImc(altura,peso); e calcular esses dois valores com uma variável recebendo eles e depois imprimindo ou já imprimir calculando. Exemplo:



Depois é só chamar a função calcularImc(72,1.74);

**Return**, o return faz com que minha função retorne o valor que eu indiquei que ela retornasse, exemplo:



Nessa função vamos retornar como resultado final o valor calculado e recebido pela variável imc, então pegando como exemplo abaixo:



Irá retornar o valor do return para o calculo do imcGuilherme e imcLuisa.

**LÓGICA DE PROGRAMAÇÃO II**

**Tag’s HTML (Linguagem de marcação ‘estática’)**

**<br>** - tag para dar quebra de linha dentro do texto html

**<h1></h1>** - tag head, cria o primeiro título da página, exemplo **<h1>** Este é um título **</h1>**

**<a>** - tag âncora, serve para criar links na web, exemplo; <a href=<http://www.exemplo.com.br>**>**aqui**</a>** onde o aqui vai virar o link

**<hr>** - essa tag define uma quebra temática entre os elementos de nível de parágrafo (exemplo uma mudança de cena de uma história ou mudança de tema) e essa quebra fica tracejada da seguinte forma \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**<big>** - aumenta o tamanho do texto.

**<script></script>** - tag script, ela identifica que todo conteúdo dentro da tag é um código javascript

**Tag’s JAVASCRIPT (Linguagem de programação ‘dinamica’)**

**INSTRUÇÕES:**

**alert** – A instrução alert recebe um texto e este texto é definido entre aspas

**document.write()** – Instrução para escrever no documento (html), sendo assim tenho a possibilidade de escrever qualquer texto DINÂMICO no meu html, então posso dar codições para o texto ser escrito por exemplo.

**var** - declaração de variável, exemplo var num=0;

**Math.round()** – calcula a média do número passado como parâmetro no parênteses, *Math* é matemática e *round* média.

**//**- é usado para comentar linhas em javascript

**function (){}**-

**prompt** – o navegador abre uma janela para que o usuário insira um dado solicitado

**parseIn()** – converte uma string para int (número inteiro).

**If()** – condições, exemplo if(18>19){} ou if(18>19 && 18>15){} e etc.

**Math.random()** – sorteia e entrega um número aleatório, podendo ser usado com o math.round(), que arredonda o número repassado.

**Math.round()** – arredonda o número.

**for()** – é um laço de repetição, exemplo for(var num=0; num<numSolicitado; num++){};

**break** – força a quebra de um laço de repetição