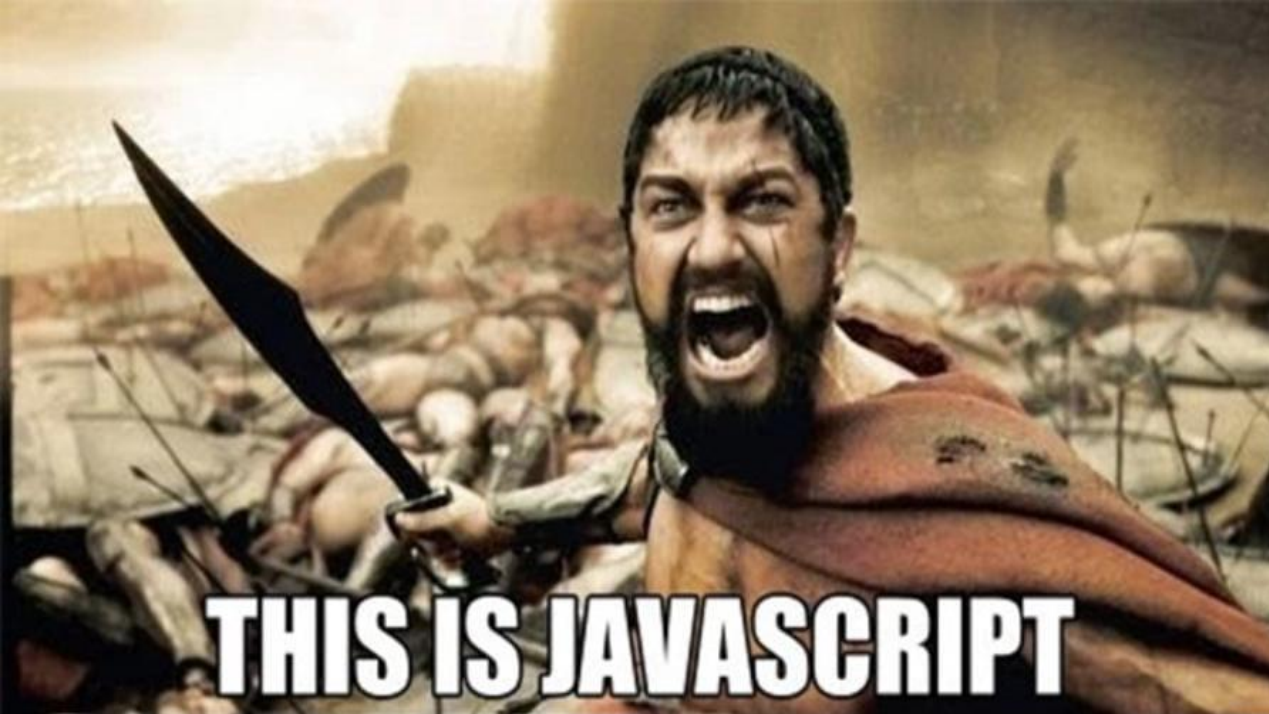


Front-End Nível II

Revisão JavaScript



THIS IS JAVASCRIPT

**WASTES 2 HOURS DEBUGGING
BROKEN JAVASCRIPT**



**MISSING A CURLY
BRACE**



IT'S NOT BECAUSE EVERYBODY USE IT

THAT IT'S EASY TO USE

javascript...



WTF AM I READING

Vamos revisar nosso aprendizado em **JavaScript**?

Construiremos o projeto do site

Aparecida Nutrição.

Vamos começar a manipular os dados do site da Aparecida. A página deixará de ser estática e ganhará mais interatividade. O índice de massa corporal (IMC) será calculado automaticamente, e conseguiremos adicionar um novo paciente sem alterarmos o HTML. Isso dará dinamismo e deixará a página mais moderna.

Vamos Começar a mexer???

Meus pacientes

Nome	Peso(kg)	Altura(m)	Gordura Corporal(%)	IMC
Paulo	100	2.00	10	0
João	80	1.72	40	0
Erica	54	1.64	14	0
Douglas	85	1.73	24	0
Tatiana	46	1.55	19	0

Adicionar novo paciente

Nome:

Peso:

Altura:

% de Gordura:

Adicionar

O que usamos para buscar informações no nosso DOM?

Temos duas formas:



`.querySelector()`

`.getElementBy...()`

Obs¹.: Não esqueça que pra manipular o DOM precisamos acessá-lo através da palavra **document**!

Obs².: Ao utilizar o **querySelector**, temos que especificar se vamos chamar uma **classe** ou **id** através dos símbolos **.** e **#**, respectivamente.

Trocando o título da página

- 1- Selecionar o elemento da página que queremos alterar, neste caso o título <h1>
- 2- Utilizar a **função seletora** que é responsável de trazer um elemento do mundo do HTML para o mundo do Javascript.
- 3- Por fim, para trocar o texto do <h1> podemos usar duas propriedades:

.textContent

.innerHTML

```
document.querySelector("#titulo").textContent = "Aparecida Nutricionista";
```

```
document.querySelector(".titulo").innerHTML = "Aparecida Nutricionista";
```

Buscando dados do paciente

Para entendermos vamos usar apenas o primeiro paciente, pra isso precisamos atribuir um id para que possamos manipular.

```
<tr class="paciente" id="primeiro-paciente">
```

Precisamos resgatar seu peso e sua altura.

```
let paciente = document.querySelector("#primeiro-paciente");  
  
let tdPeso = paciente.querySelector(".info-peso");  
let peso = tdPeso.textContent;  
  
let tdAltura = paciente.querySelector(".info-altura");  
let altura = tdAltura.textContent;
```

Calculando IMC

Vamos utilizar a fórmula para calcular uma vez que já temos os dados do paciente.

```
let imc = peso / (altura * altura);
```

Agora precisamos colocar essa informação na tabela.

```
let tdImc = paciente.querySelector(".info-imc");  
tdImc.textContent = imc;
```


Validação de dados

O que acontece se alguém, no momento de digitar o código HTML, colocar um valor que não é válido para o peso ou altura do paciente?

Temos que validar as informações e exibir uma mensagem de aviso informando que os dados do paciente são inválidos e devem ser verificados.

```
let alturaEhValida = true;
let pesoEhValido = true;

if (peso <= 0 || peso > 1000) {
    pesoEhValido = false;
}

if (altura <= 0 || altura >= 3) {
    alturaEhValida = false;
}

if (alturaEhValida && pesoEhValido) {
    tdImc.textContent = imc;
} else {
    tdImc.textContent = "Altura e/ou peso inválidos!"
}
```

Calculando IMC de todos os pacientes ^[des] programe

Para evitar **Repetição de Código**, podemos usar um *Loop*.
Mas para isso precisamos selecionar todos os pacientes usando a função:

```
const pacientes = document.querySelectorAll(".paciente");
```

Essa função retorna um *Array*, com isso podemos usar o *for* para iteração.

```
for (let i = 0; i < pacientes.length; i++) {  
  }  
}
```

Assim podemos usar os índices do *Array* para percorrê-lo.

```
let paciente = pacientes[i];
```

Limitando as casas decimais

Alguns resultados possuem diversas casas decimais, para isso usamos a função:

```
tdImc.textContent = imc.toFixed(2);
```

Adicionando estilo pelo JS

Para destacar os erros podemos alterar o estilo da linha.

```
paciente.style.backgroundColor = "lightcoral";
```

```
paciente.classList.add("paciente-invalido");
```

Eventos

A ação de "perceber o que o usuário está fazendo na página" é o que chamamos de **evento** do browser, que pode ser escutado com o JavaScript.

```
var botaoAdicionar = document.querySelector("#adicionar-paciente");  
  
botaoAdicionar.addEventListener("click", function(){  
  
}
```

Como estamos trabalhando com um formulário temos que evitar seu comportamento padrão.

```
event.preventDefault();
```


Adicionando pacientes na tabela

Primeiro temos que selecionar os dados do formulário:

```
let form = document.querySelector("#form-adiciona");  
  
let nome = form.nome.value;  
let peso = form.peso.value;  
let altura = form.altura.value;  
let gordura = form.gordura.value;
```

Criando elementos HTML com JS

```
var pacienteTr = document.createElement("tr");  
  
var nomeTd = document.createElement("td");  
var pesoTd = document.createElement("td");  
var alturaTd = document.createElement("td");  
var gorduraTd = document.createElement("td");  
var imcTd = document.createElement("td");
```

Adicionando pacientes na tabela

Preenchendo os dados dos elementos criados.

```
nomeTd.textContent = paciente.nome;  
pesoTd.textContent = paciente.peso;  
alturaTd.textContent = paciente.altura;  
gorduraTd.textContent = paciente.gordura;  
imcTd.textContent = paciente.imc;
```

Adicionando um elemento dentro de outro

```
pacienteTr.appendChild(nomeTd);  
pacienteTr.appendChild(pesoTd);  
pacienteTr.appendChild(alturaTd);  
pacienteTr.appendChild(gorduraTd);  
pacienteTr.appendChild(imcTd);
```

Adicionando a linha na tabela:

```
var tabela = document.querySelector("#tabela-pacientes");  
tabela.appendChild(pacienteTr);
```

Criando Objeto Paciente

Para facilitar colocaremos os dados do paciente criado dentro de um objeto.

```
obtemPaciente = (form) => {  
  let paciente = {  
    nome: form.nome.value,  
    peso: form.peso.value,  
    altura: form.altura.value,  
    gordura: form.gordura.value,  
    imc: form.peso.value  
  }  
  
  return paciente;  
}
```

Reaproveitando código do IMC

Criando uma função para calcular o IMC.

```
calculaImc = (peso, altura) => {  
  let imc = peso / (altura * altura);  
  return imc.toFixed(2);  
}
```

Alterando o código para que utilize a função criada.

```
if (alturaEhValida && pesoEhValido) {  
  tdImc.textContent = calculaImc(peso, altura);  
} else {  
  tdImc.textContent = "Altura e/ou peso inválidos!"  
  paciente.classList.add("paciente-invalido");  
}
```

```
obtemPaciente = (form) => {  
  let paciente = {  
    nome: form.nome.value,  
    peso: form.peso.value,  
    altura: form.altura.value,  
    gordura: form.gordura.value,  
    imc: calculaImc(form.peso.value, form.altura.value)  
  }  
  
  return paciente;  
}
```




Aeeeeeeee!!!

Terminamos a revisão!
:)

“

Um navio no porto está seguro, mas não é para isso que os navios são feitos. Vá para o mar aberto para fazer novas coisas.

— GRACE HOPPER

“Acredito que é muito importante obter mais mulheres na computação. Meu slogan é: computing é demasiado importante para ser deixado para os homens.

— KAREN SPÄRCK JONES

Dever de Casa:

- Refatorar Código quebrando as funcionalidades em funções
- Organizar arquivos de JS por funcionalidade.



Obrigada Monica



Monica Craveiro de Menezes



@monikinhax



@monikinhax



Monica Craveiro de Menezes