

JavaScript

Introdução.

`<script>` - Usado para definir um JS dentro do HTML. Tem que ser chamado no final do HTML, pois precisa carregar o documento antes.

`alert("Olá mundo")` - Abre um pop up alerta com a mensagem.

`console.log("Mensagem")` - Mostra a mensagem no console do navegador.

`<script src="contador.js"></script>` - Indicar caminho do arquivo JS no HTML.

`document.querySelector("parametro")` - Procura o parâmetro passado dentro do HTML e retorna o valor. Pode salvar o valor encontrado dentro de uma variável. Imprime a tag inteira, até com o parametro.

`console.log(variavel.textContent);` - Imprime somente o conteúdo dentro do parametro, sem os parametros.

`variavel.textContent = "Outro conteúdo";` - Muda o conteúdo dentro da variável.

- Boas práticas: nunca buscar elementos por query pela tag (h1, por exemplo), e sim por classes ou id, por exemplo.
Criar arquivos externos para JS e um arquivo para cada feature nova.
- O que contém a variável document do JavaScript ?
Ela é a representação do DOM ou Document Object Model, que é a representação da nossa página HTML para o navegador, que é utilizada pelo Javascript para manipular a página.

Variáveis e operadores.

Dentro de uma variável, é possível dar outro `querySelector` para analisar mais a fundo dela. Por exemplo:

```
var paciente = document.querySelector("#primeiro-paciente");
```

```
var tdPeso = paciente.querySelector("#info.peso");
```

```
var peso = tdPeso.textContent; - Pega somente o txt (conteúdo) que está dentro da variável.
```

```
var tdAltura = paciente.querySelector("#info.altura");
```

```
var altura = tdAltura.textContent; - Pega somente o txt (conteúdo) que está dentro da variável.
```

`var imc = peso / (altura * altura);` - É possível fazer cálculos dentro do JS com os operadores numéricos. JS lê da esquerda para a direita, então é necessário deixar os cálculos iguais a de uma calculadora.

- É possível usar IF para controlar o que será passado onde.
Operadores: `||` = ou `&&` = e
- Também é possível utilizar valores booleanos.

`var pacientes = query.SelectAll(".paciente");` - Trás todos os registros de todas as pessoas que estão dentro da classe `paciente`.

`console.log(pacientes.length)` - pega o tamanho do Array criado pelo `SelectAll`.

- Também é possível criar FOR para percorrer listas e afins.

`variavel.toFixed(2)` - Determina quantidade de casas decimais serão exibidas.

`variavel.style.color = "red";` - Muda a cor do conteúdo para vermelho.

`variavel.style.backgroundColor = "red";` - Muda o background da variável para vermelho.

`variavel.classList.add("nome-classe")` - Percorre todas as classes existentes (`classList`) e adiciona mais essa classe para a variável (`add`). Usado para modificar estilos de CSS a partir do JS. Criar classe no arquivo CSS e estilo será aplicado quando chamar no JS.

`variavel.classList.remove("nome-classe")` - Remove classe da variável.

Eventos, formulários e criando elementos.

`variavel.addEventListener("click", nomeDaFuncaoChamada);` - Quando clicar, ele chamará uma função, **chamada de função nomeada**.

`variavel.addEventListener("click", function () {`

`//Comportamento da função.`

`});` - Quando clicar, ele fará a função que foi criada dentro do parâmetro, **chamada de função anônima**.

`event.preventDefault();` - Previne que o item faça o comportamento padrão dele, por exemplo, quando apertar um botão, impedir que ele recarregue a página para enviar os dados.

`console.log(variavel.nameCampo.value);` - Pega o valor contido dentro de um campo com determinado nome.

Exemplo: `(input id="peso" name="peso" type="text")` seria `console.log(variavel.peso.value)` para pegar o valor contido no campo `peso`.

`var variavel = document.createElement("tagDoElemento");` - Especifica o tipo de elemento que você quer criar. (`document.createElement("tr");`)

`pacienteTr.appendChild(nomeTd);` - Coloca a `nomeTd` como filha do `pacienteTr`, ou seja, coloca uma coluna dentro da linha.

Boas práticas com JavaScript.

- Quebrar funcionalidades em arquivos menores.
- Separar funções para re-aproveitar.

Objetos - Variável com várias variáveis que estão relacionadas dentro.

Exemplo:

```
function obterPacienteDoFormulario (form) {
```

```
    var paciente = {
```

```
        //não é necessário colocar var antes// nome: form.nome.value,
```

```
        peso: form.peso.value,
```

```
        altura: form.altura.value,
```

```
        gordura: form.gordura.value
```

```
        } return paciente;
    }
    form.reset(); - Limpa os campos de input após adicionar.
```

Validação de formulários.

```
var variavel = [ ]; - Cria um array.
variavel.push ("Mensagem"); - Empurra mensagem para proxima entrada do array variavel.
variavel.forEach (function(){
    //Para cada item do array variável, executar determinada função.
});
```

tag.innerHTML= ""; - Permite você controlar o HTML dentro de um elemento.

- Com a propriedade innerHTML, podemos obter o conteúdo HTML (HTML interno) de um elemento.
- Quando utilizada para obter o HTML interno de um elemento, seu retorno é uma String, que representa todo o conteúdo HTML do próprio elemento.
- Com a propriedade innerHTML, podemos editar o conteúdo HTML (HTML interno) de um elemento.

Remoção, delegação e animação.

variavel.addEventListener("dblclick", nomeDaFuncaoChamada); - Ouvir double click.

this.remove() - Remove elemento que chamou. this está atrelado ao elemento que chamou o evento, no caso acima, variavel chamou.

event.target.remove(); - Remove o elemento que sofreu o evento (que foi clicado).

event.target.parentNode.remove(); - Elimina o pai do elemento que sofreu o evento.

```
setTimeout(function (){
    event.target.parentNode.remove();
},500); - Determina uma pausa de 500 milissegundos a ser tomada antes de seguir com o script.
```

Filtrando uma tabela.

```
variavel.addEventListener("input", function(){
    // Ouve quando um input é digitado.
});
```

var expressao = new RegExp (this.value, "i"); - RegExp é um construtor JS usado para filtrar se o alguma parte do valor que ele está recebendo corresponde com outro elemento que está sendo comparado. Usado, por exemplo, para realizar pesquisa por nomes sem a necessidade de informá-lo completo. "i" - Para case insensitive.

expressao.test(nome) - Testa se o valor que foi pego pelo RegEx consta em alguma parte dos nomes.

Ajax - Buscando pacientes com AJAX

var xhr = new XMLHttpRequest(); - Responsavel por fazer a requisição http.

xhr.open("GET", "enderecoHttp"); - Abre a conexão. Tipo GET, para pegar o endereço/requisição.

xhr.send(); - Envia a requisição.

xhr.addEventListener("load", function(){ - Quando carregar a requisição, fazer...

console.log(xhr.responseText); }); - Imprime no console o texto resposta que obteve da requisição.

console.log(typeof variavel); - Fala qual o tipo da resposta. Ex: String.

var pacientes = JSON.parse(resposta); - Lê um texto em JSON e te responde com um objeto (array) em JS com o conteúdo do JSON.