

Aula 2

HTML, CSS e JavaScript

Dedicado à minha família e amigos,

Aos meus familiares, que sempre acreditaram em mim e me ensinaram o valor do esforço e da persistência.

Aos meus amigos, que, com suas palavras de apoio e momentos compartilhados, tornaram essa jornada mais leve e mais alegre. Cada um de vocês contribuiu para este projeto de alguma forma, e sou imensamente grato por tê-los ao meu lado.

Dheison A. da Silva

Registro de agendamento com HTML, CSS e JavaScript

```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Formulário com Tabela</title>
    <style>
      table {
        border-collapse: collapse;
      }
      table,
      th,
      td {
        border: 1px solid black;
      }
    </style>
  </head>
  <body>
    <h2>Adicionar Informações à Tabela</h2>
    <label for="nome">Nome:</label>
    <input type="text" id="nome" placeholder="Digite o nome" />
    <br /><br />
    <label for="email">Email:</label>
    <input type="email" id="email" placeholder="Digite o email" />
    <br /><br />
    <label for="telefone">Telefone:</label>
    <input type="tel" id="telefone" placeholder="Digite o telefone" />
    <br /><br />
    <label for="data">Telefone:</label>
    <input type="date" id="data" placeholder="Escolha uma data" />
    <br /><br />
    <button onclick="adicionarLinha()">Adicionar</button>
    <h2>Informações Adicionadas</h2>
    <table>
      <thead>
```

```

<tr>
  <th>Nome</th>
  <th>Email</th>
  <th>Telefone</th>
  <th>Data</th>
  <th>Ação</th>
</tr>
</thead>
<tbody id="tabela-corpo"></tbody>
</table>
<script>
function adicionarLinha() {
  let nome = document.getElementById("nome").value;
  let email = document.getElementById("email").value;
  let telefone = document.getElementById("telefone").value;
  let data = document.getElementById("data").value;
  if (nome && email && telefone) {
    let novaLinha = document.createElement("tr");
    let celulaNome = document.createElement("td");
    celulaNome.textContent = nome;
    let celulaEmail = document.createElement("td");
    celulaEmail.textContent = email;
    let celulaTelefone = document.createElement("td");
    celulaTelefone.textContent = telefone;
    let celulaData = document.createElement("td");
    celulaData.textContent = data.split("-").reverse().join("/");
    let celulaAcao = document.createElement("td");
    let botaoExcluirLinha = document.createElement("button");
    botaoExcluirLinha.addEventListener("click", function () {
      excluirLinha(novaLinha);
    });
    botaoExcluirLinha.textContent = "Excluir";
    botaoExcluirLinha.style.backgroundColor = "red";
    botaoExcluirLinha.style.color = "white";
    celulaAcao.appendChild(botaoExcluirLinha);
    novaLinha.appendChild(celulaNome);
    novaLinha.appendChild(celulaEmail);
  }
}

```

```

        novaLinha.appendChild(celulaTelefone);
        novaLinha.appendChild(celulaData);
        novaLinha.appendChild(celulaAcao);
        document.getElementById("tabela-corpo").appendChild(novaLinha);
        document.getElementById("nome").value = "";
        document.getElementById("email").value = "";
        document.getElementById("telefone").value = "";
        document.getElementById("data").value = "";
    } else {
        alert("Por favor, preencha todos os campos.");
    }
}
function excluirLinha(linha) {
    let confirmacao = confirm("Deseja realmente excluir a linha?");
    if (confirmacao) {
        linha.remove();
    }
}
</script>
</body>
</html>

```

Este código HTML cria uma página simples que permite ao usuário adicionar informações (nome, email, telefone e data) em uma tabela dinamicamente. A funcionalidade principal está no JavaScript, que manipula o DOM para adicionar e remover linhas na tabela. Vou explicar cada parte do código, focando principalmente no JavaScript.

Estrutura HTML:

1. Formulário para entrada de dados:

- O usuário insere o nome, email, telefone e data através de campos de entrada (`<input>`).
- Cada campo de entrada tem um `id`, que será usado para obter os valores no JavaScript.
- Um botão "Adicionar" chama a função `adicionarLinha()` quando clicado.

2. Tabela:

- A tabela contém um cabeçalho (`<thead>`) com as colunas "Nome", "Email", "Telefone", "Data" e "Ação".

- O corpo da tabela (`<tbody>`) está vazio inicialmente, e novas linhas serão adicionadas dinamicamente usando JavaScript. O `id="tabela-corpo"` identifica onde as novas linhas serão inseridas.

Estilo CSS:

- A tabela utiliza um estilo simples para colapsar bordas, e cada célula tem uma borda de 1 pixel para tornar a tabela visualmente organizada.

JavaScript:

1. **Função `adicionarLinha()`:** Esta função é responsável por capturar os dados dos campos de entrada, criar uma nova linha na tabela com esses dados, e adicionar um botão de exclusão à nova linha.

- **Captura dos dados:**

```
let nome = document.getElementById("nome").value;
let email = document.getElementById("email").value;
let telefone = document.getElementById("telefone").value;
let data = document.getElementById("data").value;
```

Aqui, a função captura os valores dos campos de entrada usando `document.getElementById().value`.

- **Verificação dos dados:**

```
if (nome && email && telefone) {
```

A função verifica se os campos obrigatórios (`nome`, `email`, `telefone`) estão preenchidos. Caso contrário, exibe um alerta pedindo que o usuário preencha todos os campos.

- **Criação da nova linha na tabela:**

```
let novaLinha = document.createElement("tr");
```

Se todos os dados forem válidos, a função cria um novo elemento `<tr>` (linha da

tabela).

- **Criação das células:** Para cada campo de entrada (nome, email, telefone, data), é criada uma célula (`<td>`), e o valor capturado é atribuído a essa célula.

```
let celularNome = document.createElement("td");
celularNome.textContent = nome;
```

A célula para a data utiliza uma transformação para reformatar a data do padrão `AAAA-MM-DD` para `DD/MM/AAAA`:

```
celulaData.textContent = data.split("-").reverse().join("/");
```

- **Criação do botão de exclusão:** Para permitir que o usuário exclua a linha, a função cria um botão de exclusão:

```
let botaoExcluirLinha = document.createElement("button");
botaoExcluirLinha.textContent = "Excluir";
botaoExcluirLinha.style.backgroundColor = "red";
botaoExcluirLinha.style.color = "white";
```

O botão é estilizado e recebe um evento `click` com a função `excluirLinha()`:

```
botaoExcluirLinha.addEventListener("click", function () {
    excluirLinha(novaLinha);
});
```

Esse código garante que, ao clicar no botão, a linha correspondente será excluída.

- **Inserção da linha na tabela:** Após criar todas as células e o botão, a nova linha é adicionada ao corpo da tabela:

```
document.getElementById("tabela-corpo").appendChild(novaLinha);
```

- **Limpeza dos campos de entrada:** Para facilitar novas entradas, a função limpa os campos de entrada após a adição de uma linha:

```
document.getElementById("nome").value = "";  
document.getElementById("email").value = "";  
document.getElementById("telefone").value = "";  
document.getElementById("data").value = "";
```

2. **Função `excluirLinha(linha)`**: Essa função é responsável por remover uma linha da tabela quando o botão de exclusão for clicado.

- A função usa um `confirm()` para pedir a confirmação do usuário antes de excluir a linha:

```
let confirmacao = confirm("Deseja realmente excluir a linha?");
```

- Se o usuário confirmar, a linha é removida do DOM:

```
linha.remove();
```

Conclusão:

Este código cria uma página que permite adicionar e remover linhas em uma tabela. Ele exemplifica manipulação de DOM, eventos e controle de formulários com JavaScript. O ponto central do JavaScript é a função `adicionarLinha`, que captura os dados dos inputs, cria dinamicamente novos elementos de tabela e insere esses elementos no DOM. A função `excluirLinha` complementa ao permitir a exclusão de linhas individualmente.

Exercícios

Dia 1: Formulário de Cadastro de Livros

Exercício:

Crie uma página HTML com um formulário que permita cadastrar livros. O formulário deve ter campos para o título do livro, o nome do autor e o ano de publicação. Ao clicar no botão "Adicionar", o JavaScript deve criar uma linha em uma tabela contendo as informações inseridas no formulário.

Objetivo: Praticar a criação de formulários e manipulação de tabelas com JavaScript.

Dia 2: Formulário de Cadastro de Produtos

Exercício:

Crie um formulário para cadastrar produtos de uma loja virtual. O formulário deve conter os seguintes campos: nome do produto, preço e quantidade em estoque. Quando o usuário clicar em "Adicionar", uma linha com as informações do produto deve ser criada em uma tabela.

Objetivo: Exercitar o uso de JavaScript para preencher tabelas com informações numéricas e de texto.

Dia 3: Formulário de Registro de Participantes de Evento

Exercício:

Desenvolva uma página HTML com um formulário para registrar participantes de um evento. O formulário deve conter campos para o nome do participante, e-mail e telefone. Ao clicar no botão "Registrar", os dados inseridos devem ser adicionados a uma tabela exibida abaixo do formulário.

Objetivo: Praticar a inserção de dados de contato em tabelas.

Dia 4: Formulário de Cadastro de Veículos

Exercício:

Crie um formulário para cadastrar veículos. Os campos devem incluir: marca, modelo e ano de fabricação. Ao clicar no botão "Adicionar Veículo", o JavaScript deve criar uma nova linha na tabela com as informações fornecidas.

Objetivo: Trabalhar com diferentes tipos de dados (texto e números) em formulários e tabelas.

Dia 5: Formulário de Cadastro de Tarefas

Exercício:

Crie uma página HTML com um formulário para cadastrar tarefas de um projeto. O formulário deve conter campos para o nome da tarefa, responsável e data de entrega. Ao clicar no botão "Adicionar Tarefa", as informações devem ser exibidas em uma tabela.

Objetivo: Praticar a inserção de datas e manipulação de tabelas.

Dia 6: Formulário de Cadastro de Alunos

Exercício:

Desenvolva um formulário que permita cadastrar alunos de uma escola. Os campos devem incluir: nome, idade e série escolar. Quando o aluno clicar no botão "Cadastrar", os dados devem ser inseridos em uma tabela abaixo do formulário.

Objetivo: Trabalhar com diferentes tipos de informações (idade e série escolar) em tabelas, reforçando o uso de JavaScript para criar interfaces dinâmicas.