



HTML, CSS e JavaScript: A Jornada do Codificador.



Diego Francisco da Silva

INTRODUÇÃO AO HTML, CSS E JAVASCRIPT

Conceitos e Exemplos dos Elementos

Bem-vindo ao guia básico sobre *HTML, CSS e JavaScript!* Este e-book vai te ajudar a entender e utilizar as principais tags e funcionalidades dessas três tecnologias essenciais para o desenvolvimento web. Vamos direto ao ponto, com exemplos claros e objetivos para facilitar seu aprendizado.



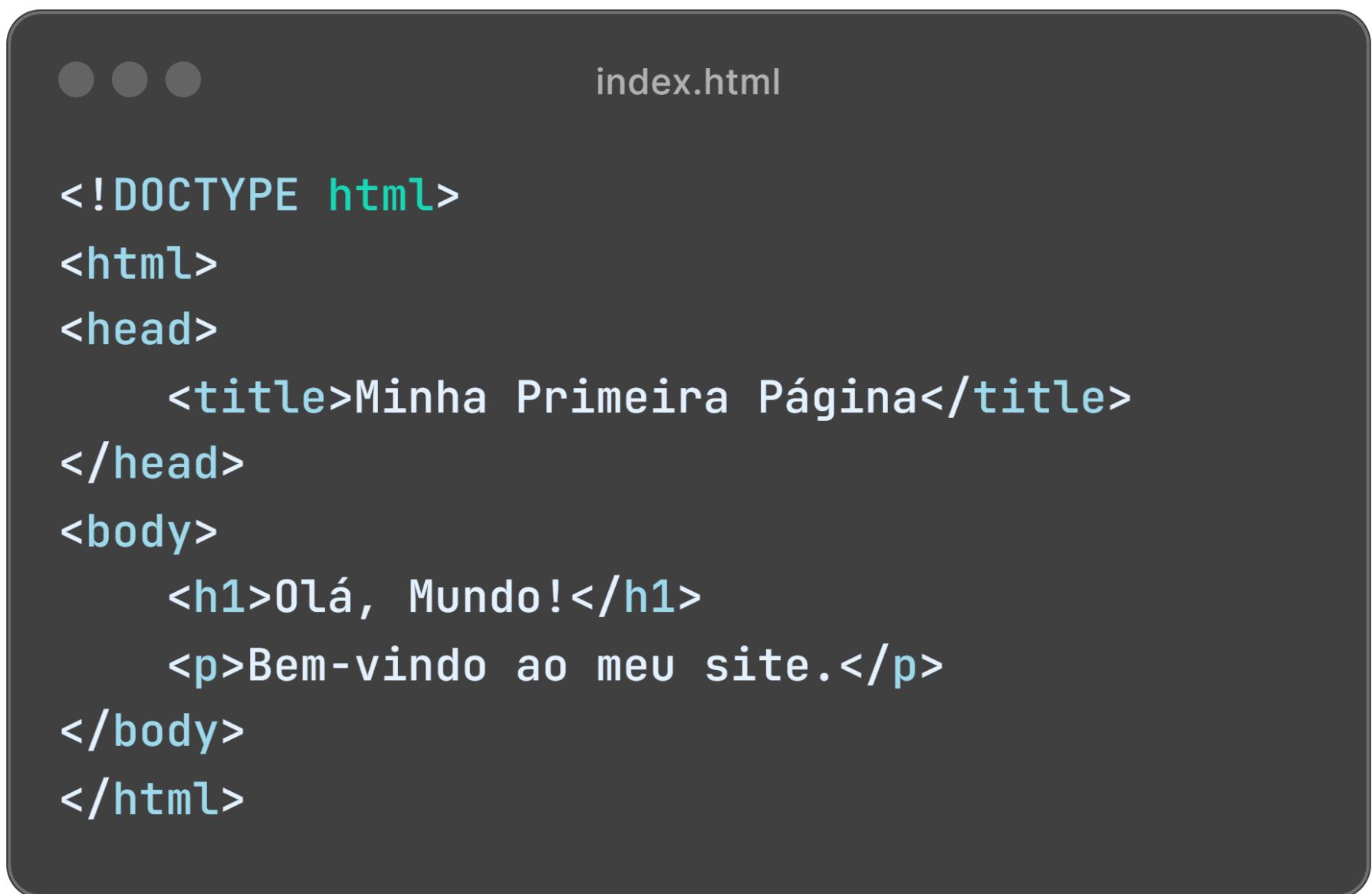
HTML - Estrutura da Página

HTML (HyperText Markup Language) é a linguagem usada para criar a estrutura das páginas web. Vamos conhecer algumas das principais tags.

1.1 <html>, <head>, <body>.

Conceitos e Exemplos dos Elementos

Essas são as tags básicas que formam a estrutura de qualquer documento HTML.



The screenshot shows a dark-themed code editor window. In the top right corner, the file name "index.html" is displayed. On the far left, there are three small circular icons. The main area contains the following HTML code:

```
<!DOCTYPE html>
<html>
<head>
    <title>Minha Primeira Página</title>
</head>
<body>
    <h1>Olá, Mundo!</h1>
    <p>Bem-vindo ao meu site.</p>
</body>
</html>
```

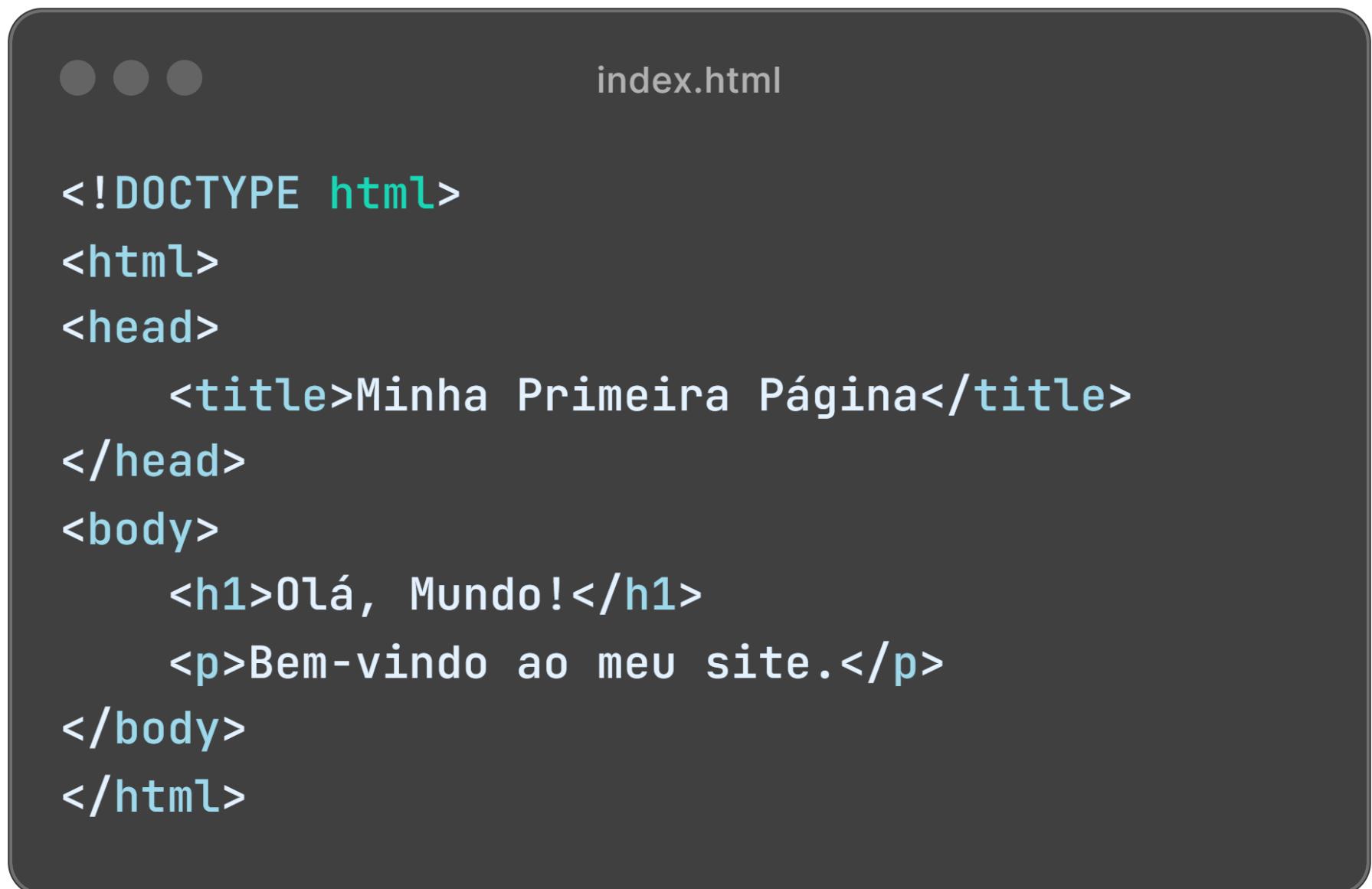
1.1 <html>, <head>, <body>.

Conceitos e Exemplos dos Elementos

<html>: Envolve todo o conteúdo da página.

<head>: Contém meta-informações, como o título e links para CSS.

<body>: Contém o conteúdo visível da página.



```
index.html

<!DOCTYPE html>
<html>
<head>
    <title>Minha Primeira Página</title>
</head>
<body>
    <h1>Olá, Mundo!</h1>
    <p>Bem-vindo ao meu site.</p>
</body>
</html>
```

1.2 <h1> a <h6>: Títulos.

Conceitos e Exemplos dos Elementos

Os títulos ajudam a organizar o conteúdo em seções.



Títulos <h1> a <h6>

```
<h1>Título Principal</h1>
<h2>Subtítulo</h2>
<h3>Seção</h3>
<h4>Seção</h4>
<h5>Seção</h5>
<h6>Seção</h6>
```

<h1>: Título mais importante.

<h6>: Título menos importante.

1.3 <p> : Parágrafo.

Conceitos e Exemplos dos Elementos

Para textos corridos.



Parágrafo <p>

```
<p>Este é um parágrafo de exemplo.</p>
```

1.4 <a> : Links.

Cria links para outras páginas ou seções da mesma página.



Link <a>

```
<a href="https://www.example.com">Visite o Example</a>
```

href: Define o destino do link.

1.5 : Imagens.

Conceitos e Exemplos dos Elementos

Insere imagens na página.



imagens

```

```

src: Caminho da imagem.

alt: Texto alternativo para acessibilidade.



CSS - Estilizando a Página

CSS (Cascading Style Sheets) é usado para estilizar elementos HTML. Vamos ver como aplicá-lo.

2.1 Seletores Básicos.

Conceitos e Exemplos dos Elementos

Você pode selecionar elementos pelo nome da tag, classe ou ID.



Seletores CSS

```
<style>
    body {
        background-color: #f0f0f0;
        font-family: Arial, sans-serif;
    }
    .destacado {
        color: red;
        font-weight: bold;
    }
    #titulo-principal {
        font-size: 2em;
    }
</style>
```

body: Seleciona a tag <body>.

.destacado: Seleciona elementos com a classe "destacado".

#titulo-principal: Seleciona o elemento com o ID "titulo-principal".

2.2 Layout com Flexbox.

Conceitos e Exemplos dos Elementos
Flexbox facilita a criação de layouts flexíveis.



Flexbox

```
<style>
  .container {
    display: flex;
    justify-content: space-around;
  }
  .item {
    padding: 10px;
    background-color: #ccc;
  }
</style>
<div class="container">
  <div class="item">Item 1</div>
  <div class="item">Item 2</div>
  <div class="item">Item 3</div>
</div>
```

display: flex;: Ativa o flexbox no container.
justify-content: space-around;: Distribui os itens com espaços ao redor.



Exemplos Avançados de CSS

Neste capítulo, vamos explorar exemplos avançados de CSS para estilizar suas páginas web de maneira mais sofisticada. Veremos como criar layouts complexos, aplicar animações e utilizar recursos modernos do CSS.

3.1 Layout Responsivo com Grid.

Conceitos e Exemplos dos Elementos

O CSS Grid Layout é uma poderosa ferramenta para criar layouts responsivos.

```
● ● ● Grid layout

<head>
  <style>
    .container {
      display: grid;
      grid-template-columns: repeat(3, 1fr);
      gap: 10px;
    }
    .item {
      background-color: lightblue;
      padding: 20px;
      text-align: center;
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="item">Item 1</div>
    <div class="item">Item 2</div>
    <div class="item">Item 3</div>
    <div class="item">Item 4</div>
    <div class="item">Item 5</div>
    <div class="item">Item 6</div>
  </div>
</body>
```

grid-template-columns: Define a estrutura das colunas do grid.

repeat(3, 1fr): Cria três colunas de igual tamanho.

gap: Define o espaço entre os itens do grid.

3.2 Media Queries.

Conceitos e Exemplos dos Elementos

Media queries permitem criar layouts que se adaptam a diferentes tamanhos de tela.

```
••• Media queries

<head>
  <style>
    .container {
      background-color: lightgreen;
      padding: 20px;
      text-align: center;
    }
    @media (max-width: 600px) {
      .container {
        background-color: lightcoral;
      }
    }
  </style>
</head>
<body>
  <div class="container">
    Este container muda de cor em telas menores que 600px.
  </div>
</body>
```

@media (max-width: 600px): Aplica estilos quando a largura da tela é de 600 pixels ou menor.

3.3 Pseudo-classes e Pseudo-elementos.

Conceitos e Exemplos dos Elementos

Pseudo-classes e pseudo-elementos permitem aplicar estilos com base em estados específicos dos elementos.



Pseudo classes

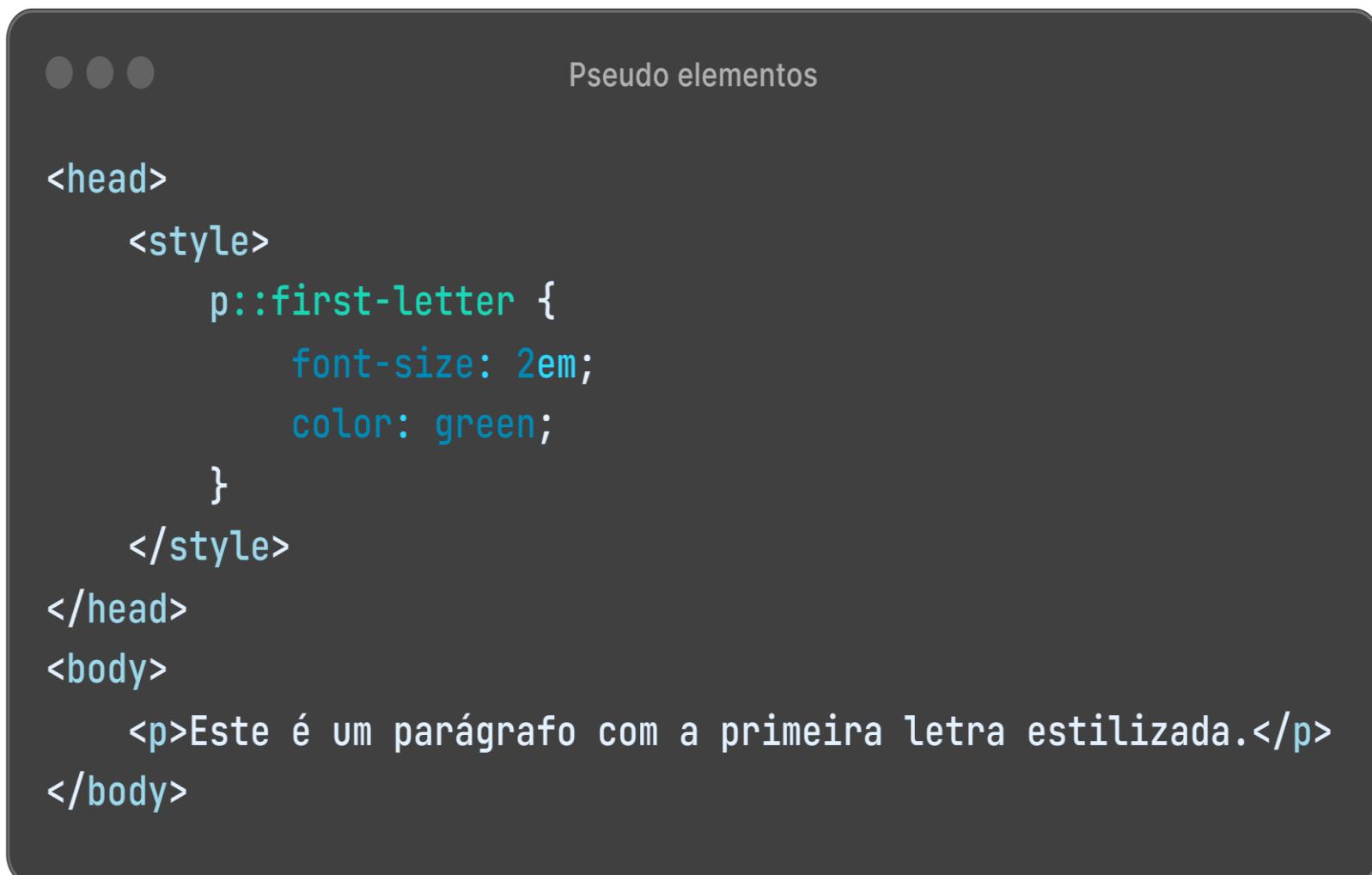
```
<head>
  <style>
    a {
      text-decoration: none;
      color: blue;
    }
    a:hover {
      color: red;
    }
  </style>
</head>
<body>
  <a href="#">Passe o mouse sobre este link</a>
</body>
```

:hover: Aplica estilo quando o mouse está sobre o elemento.

3.3 Pseudo-classes e Pseudo-elementos.

Conceitos e Exemplos dos Elementos

Pseudo-classes e pseudo-elementos permitem aplicar estilos com base em estados específicos dos elementos.



The screenshot shows a dark-themed code editor window. In the top right corner, there are three small circular icons. The title bar reads "Pseudo elementos". The code itself is as follows:

```
<head>
  <style>
    p::first-letter {
      font-size: 2em;
      color: green;
    }
  </style>
</head>
<body>
  <p>Este é um parágrafo com a primeira letra estilizada.</p>
</body>
```

The code uses the `::first-letter` pseudo-class to target the first letter of each paragraph and applies a large green font size.

::first-letter: Estiliza a primeira letra do parágrafo.

3.4 Animações com CSS.

Conceitos e Exemplos dos Elementos

CSS permite criar animações sem o uso de JavaScript.



Transições simples

```
<head>
  <style>
    .caixa {
      width: 100px;
      height: 100px;
      background-color: blue;
      transition: background-color 1s;
    }
    .caixa:hover {
      background-color: green;
    }
  </style>
</head>
<body>
  <div class="caixa"></div>
</body>
```

transition: Define a transição entre os estados dos elementos.

3.4 Animações com CSS.

Conceitos e Exemplos dos Elementos
Animações Complexas.

```
● ● ● Animações complexas

<head>
  <style>
    @keyframes mover {
      from {left: 0;}
      to {left: 100px;}
    }
    .animacao {
      width: 50px;
      height: 50px;
      background-color: red;
      position: relative;
      animation: mover 2s infinite alternate;
    }
  </style>
</head>
<body>
  <div class="animacao"></div>
</body>
```

@keyframes: Define a animação.
animation: Aplica a animação ao elemento.

3.5 Flexbox.

Conceitos e Exemplos dos Elementos

Flexbox é outra ferramenta poderosa para criar layouts flexíveis.



```
<head>
  <style>
    .container {
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
    }
    .item {
      background-color: lightblue;
      padding: 20px;
      margin: 10px;
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="item">Item 1</div>
    <div class="item">Item 2</div>
    <div class="item">Item 3</div>
  </div>
</body>
```

display: flex;: Ativa o Flexbox no container.

justify-content: center;: Centraliza os itens horizontalmente.

align-items: center;: Centraliza os itens verticalmente.



JavaScript - Interatividade

JavaScript adiciona interatividade às páginas web.
Vamos ver alguns exemplos básicos.

4.1 Manipulando o DOM.

Conceitos e Exemplos dos Elementos

Modificando o conteúdo de elementos HTML.

```
Manipulando o DOM

<script>
    document.getElementById("meuBotao").addEventListener("click", function() {
        document.getElementById("demo").innerHTML = "Texto alterado!";
    });
</script>
<button id="meuBotao">Clique aqui</button>
<p id="demo">Texto original.</p>
```

getElementById: Seleciona um elemento pelo ID.

addEventListener: Adiciona um evento ao elemento.

4.2 Funções e Eventos.

Conceitos e Exemplos dos Elementos

Criando uma função que é chamada quando um evento ocorre.



The screenshot shows a presentation slide with a dark gray background. In the top left corner, there are three small circular icons. In the top right corner, the text "Funções e Eventos" is displayed. The main content is a block of code:

```
<script>
    function mostrarAlerta() {
        alert("Olá, Mundo!");
    }
</script>
<button onclick="mostrarAlerta()">Clique aqui</button>
```

function: Define uma função.

onclick: Evento que chama a função quando o botão é clicado.



JavaScript

Exemplos Avançados

Vamos explorar mais alguns exemplos de JavaScript para aumentar a interatividade das suas páginas web. Esses exemplos irão te ajudar a entender melhor como manipular o DOM, trabalhar com eventos e utilizar algumas funcionalidades mais avançadas.

5.1 Alterando Estilos com JavaScript.

Conceitos e Exemplos dos Elementos

Você pode usar JavaScript para alterar o estilo dos elementos HTML dinamicamente.



The screenshot shows a dark-themed browser window. At the top, there are three circular icons on the left and the title "Alterando estilos com JavaScript" on the right. Below the title, the browser displays the following HTML and JavaScript code:

```
<!DOCTYPE html>
<html>
<head>
<style>
    .cor-fundo {
        background-color: yellow;
    }
</style>
</head>
<body>
    <p id="meuParagrafo">Clique no botão para mudar minha cor de fundo.</p>
    <button onclick="mudarCor()">Mudar Cor</button>

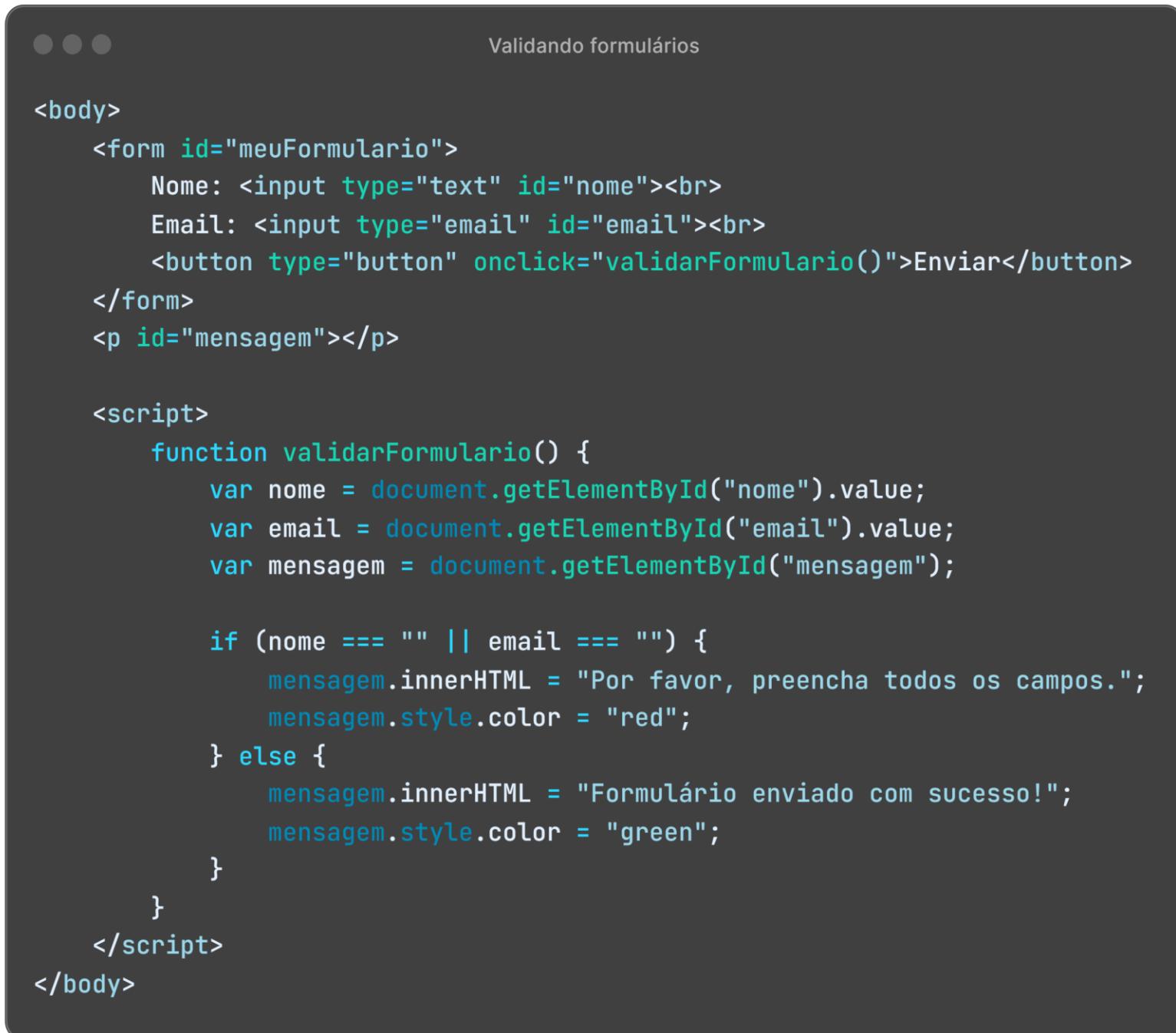
    <script>
        function mudarCor() {
            document.getElementById("meuParagrafo").classList.toggle("cor-fundo");
        }
    </script>
</body>
</html>
```

classList.toggle: Adiciona ou remove a classe especificada do elemento.

5.2 Validando Formulários.

Conceitos e Exemplos dos Elementos

Vamos criar um exemplo de validação de formulário usando JavaScript.



The screenshot shows a code editor window with a dark theme. The title bar says "Validando formulários". The code itself is as follows:

```
<body>
    <form id="meuFormulario">
        Nome: <input type="text" id="nome"><br>
        Email: <input type="email" id="email"><br>
        <button type="button" onclick="validarFormulario()">Enviar</button>
    </form>
    <p id="mensagem"></p>

    <script>
        function validarFormulario() {
            var nome = document.getElementById("nome").value;
            var email = document.getElementById("email").value;
            var mensagem = document.getElementById("mensagem");

            if (nome === "" || email === "") {
                mensagem.innerHTML = "Por favor, preencha todos os campos.";
                mensagem.style.color = "red";
            } else {
                mensagem.innerHTML = "Formulário enviado com sucesso!";
                mensagem.style.color = "green";
            }
        }
    </script>
</body>
```

value: Obtém o valor do campo de entrada.

innerHTML: Altera o conteúdo HTML interno de um elemento.

5.3 Manipulando Listas.

Conceitos e Exemplos dos Elementos

Adicionando e removendo itens de uma lista.

```
Manipulando listas

<body>
    <h2>Minha Lista de Tarefas</h2>
    <ul id="minhaLista">
        <li>Estudar JavaScript</li>
        <li>Praticar exercícios</li>
    </ul>
    <input type="text" id="novaTarefa" placeholder="Nova tarefa">
    <button onclick="adicionarTarefa()">Adicionar</button>
    <button onclick="removerTarefa()">Remover Última</button>

    <script>
        function adicionarTarefa() {
            var novaTarefa = document.getElementById("novaTarefa").value;
            var li = document.createElement("li");
            li.textContent = novaTarefa;
            document.getElementById("minhaLista").appendChild(li);
        }

        function removerTarefa() {
            var lista = document.getElementById("minhaLista");
            lista.removeChild(lista.lastChild);
        }
    </script>
</body>
```

createElement: Cria um novo elemento HTML.

appendChild: Adiciona um novo nó como o último filho de um elemento.

removeChild: Remove um nó filho de um elemento.

5.4 Temporizadores e Animações.

Conceitos e Exemplos dos Elementos

Usando setTimeout e setInterval para temporizadores e animações.

```
••• Temporizadores e Animações

<body>
    <button onclick="mostrarMensagem()">Mostrar Mensagem em 3 Segundos</button>
    <p id="temporizadorMensagem"></p>

    <script>
        function mostrarMensagem() {
            setTimeout(function() {
                document.getElementById("temporizadorMensagem").innerHTML = "3 segundos se
passaram!";
            }, 3000);
        }
    </script>
</body>
```

setTimeout: Executa uma função após um período de tempo especificado.

5.4 Temporizadores e Animações.

Conceitos e Exemplos dos Elementos

Animação Simples.

```
● ● ● Temporizadores e Animações

<body>
    <div id="quadrado"></div>
    <button onclick="iniciarAnimacao()">Iniciar Animação</button>

    <script>
        function iniciarAnimacao() {
            var elem = document.getElementById("quadrado");
            var pos = 0;
            var id = setInterval(frame, 10);

            function frame() {
                if (pos == 350) {
                    clearInterval(id);
                } else {
                    pos++;
                    elem.style.top = pos + "px";
                    elem.style.left = pos + "px";
                }
            }
        }
    </script>
</body>
```

setInterval: Executa uma função repetidamente com um intervalo de tempo especificado.

clearInterval: Para a execução repetida da função.

Conclusão

Muito Obrigado por Ler Até Aqui !

HTML, CSS e JavaScript :
A Jornada do Codificador.

Este guia básico apresentou as principais tags e funcionalidades do HTML, CSS e JavaScript. Com esses exemplos, você já pode começar a criar e estilizar suas próprias páginas web, além de adicionar interatividade. Pratique bastante e explore mais para se tornar um desenvolvedor web completo!

Agradecimento

Muito Obrigado por Ler Até Aqui !

HTML, CSS e JavaScript :
A Jornada do Codificador.

E-Book desenvolvido com ChatGpt (Roteiro),
Midjourney (Artista),
Power Point (Diagramação) e
Site ray.so (Imagens de Código).

Conteúdo gerado para fins didáticos de estudo e elaborado por mim, Diego Francisco da Silva como parte do projeto BootCamp IA para Devs Santander, ministrado pelo instrutor do curso Felipe Aguiar.



https://github.com/diegofranciscodasilva/projeto_e-book-gerado-por-IAs-DIO.git