

Curso lógica de programação

CODE - X



Lógica de programação - HTML



PALAVRA DO ORIENTADOR

Olá! Sou formado nas áreas técnicas de Química, Meio Ambiente e Tecnologia da Informação, além de possuir graduação em Licenciatura em Matemática. Minha paixão pelo conhecimento e pelo ensino sempre me motivou a buscar a excelência nas minhas áreas de atuação. Ao longo da minha jornada, tive a vantagem de acumular experiências valiosas, tanto no campo profissional quanto acadêmico, e acredito que o aprendizado é um processo contínuo e enriquecedor.

É com grande satisfação que ofereço apoio e orientação a você, que está começando ou se aprofundando em sua carreira e nos estudos. Compreendo os desafios e as conquistas desta fase e fico muito feliz por poder compartilhar um pouco da minha trajetória e conhecimento.

Sinta-se à vontade para contar comigo durante essa jornada intensa de estudos e trabalho. Estou aqui para ajudar no que for necessário, oferecendo orientação, apoio e incentivo. Juntos, podemos transformar desafios em oportunidades de crescimento e alcançar grandes objetivos. Conte sempre comigo!

SUMÁRIO

Introdução.....	05
1. Estrutura Básica de um Documento HTML.....	07
2. Tags e Elementos HTML	08
3. Semântica no HTML (Continuação)	09
4. Atributos HTML	09
5. Listas em HTML	10
6. Formulários em HTML	12
7. Links e Navegação	13
8. Tabelas no HTML	14
9. Estrutura e Hierarquia do HTML	15
10. Multimídia no HTML	15
11. Elementos de Estilo Inline	16
12. Elementos de script (JavaScript)	16
13. Meta Tags e SEO (Search Engine Optimization)	17
14. Iframe e Conteúdos Incorporados	17
15. Elementos Interativos Avançados	18
16. Acessibilidade e ARIA (Aplicativos de Internet Ricos Acessíveis)	19

17. Boas Práticas de HTML e Otimização.....	20
18. Estrutura de Código HTML Limpo e Comentários	20
19. Validação e Compatibilidade entre Navegadores	21
Referências	22

CODE-X

FUNDAMENTOS DE LÓGICA DE PROGRAMAÇÃO

Olá, estudante! Seja muito bem-vindo ao fascinante universo da programação! Se você está aqui, é porque já ouviu falar sobre códigos, softwares, linguagens de programação e, principalmente, sobre a lógica por trás deles. Talvez você já tenha se perguntado: “Por onde começar? Como transformar ideias em algo funcional e visual na tela?”. Essas são dúvidas comuns para quem está iniciando. A boa notícia é que estamos juntos nessa jornada, e o primeiro passo será desbravar os fundamentos da lógica de programação aplicada ao HTML.

Nesta unidade, você irá explorar os conceitos fundamentais que fazem do HTML a base estrutural de qualquer página web. Aprenderemos como transformar ideias em códigos, construindo a espinha dorsal de um site ou aplicação web. Você descobrirá como a lógica de programação se aplica à organização, estruturação e apresentação de conteúdo na web, começando pelas tags e elementos básicos até os conceitos mais avançados que permitem a criação de páginas interativas e dinâmicas.

Dominar a lógica de programação no HTML é mais do que apenas aprender uma linguagem; é entender como o pensamento lógico se traduz em estruturas claras, eficientes e funcionais. Esse conjunto de conhecimentos não irá apenas prepará-lo para enfrentar os desafios do mercado de trabalho, mas também irá capacitá-lo a contribuir de forma significativa para o avanço tecnológico, criando soluções que impactam o cotidiano de milhões de pessoas. A construção de uma web acessível, rápida e confiável começa aqui, com os primeiros passos que você dará nesta unidade.

FUNDAMENTOS DE LÓGICA DE PROGRAMAÇÃO

Vamos juntos transformar conceitos abstratos em realizações concretas e construir uma base sólida para um futuro na programação!

CODE-X

Introdução

Embora o HTML (HyperText Markup Language) não seja uma linguagem de programação no sentido tradicional, ele é uma peça fundamental na construção de sites e aplicações web. A lógica de programação no contexto do HTML está ligada à estruturação correta do conteúdo da página, à semântica adequada dos elementos e à forma como esses elementos interagem com linguagens como CSS e JavaScript.

Entender a lógica por trás do HTML é essencial para criar páginas organizadas, acessíveis e eficientes. Isso significa não apenas saber quais tags utilizar, mas compreender o propósito de cada uma e como elas afetam a situação e o comportamento do conteúdo. Uma página bem estruturada em HTML facilita a aplicação de estilos com CSS, melhora o desempenho da página e, principalmente, proporciona uma experiência de usuário mais consistente.

Além disso, uma lógica aplicada ao HTML é crucial para o desenvolvimento de sites responsivos e acessíveis. Seguir uma lógica coerente ao utilizar cabeçalhos (<h1>, <h2>, <h3>...), listas, tabelas e outros elementos garantem que uma página seja interpretada corretamente por navegadores e tecnologias assistivas, como leitores de tela. A organização correta do HTML impacta diretamente o SEO (Search Engine Optimization), tornando o conteúdo mais visível para mecanismos de busca.

Para desenvolvedores que desejam criar sites dinâmicos e interativos, uma compreensão sólida da lógica no HTML é ainda mais importante. Ela serve como base para integrar JavaScript, que adiciona funcionalidades avançadas e interatividade, e CSS, que melhora a aparência visual. Uma boa lógica no HTML facilita a manipulação desses elementos no JavaScript e o uso de frameworks modernos, como React, Angular e Vue.

Portanto, dominar a lógica de programação no HTML é um passo essencial tanto para iniciantes quanto para desenvolvedores avançados, pois permite criar estruturas sólidas, flexíveis e escaláveis, que servem como alicerce para a construção de aplicações web de alta qualidade.



FUNDAMENTOS DE LÓGICA DE PROGRAMAÇÃO - HTML

1. Estrutura Básica de um Documento HTML

O que é: A estrutura do HTML é a base de qualquer página web. Ela define a organização dos elementos e garante que o navegador compreenda como o conteúdo deve ser listado.

Exemplo da Estrutura Básica:

```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Título da Página</title>
  </head>
  <body>
    <h1>Olá, Mundo!</h1>
    <p>Bem-vindo ao mundo do HTML.</p>
  </body>
</html>
```

- **<!DOCTYPE html>**: Informa ao navegador que o documento usa HTML5.
- **<html>**: O elemento raiz que envolve todo o conteúdo.
- **<head>**: Contém informações sobre a página, como título, meta tags e links para CSS ou scripts.
- **<body>**: Onde o conteúdo visível da página é colocado.

FUNDAMENTOS DE LÓGICA DE PROGRAMAÇÃO - HTML

2. Tags e Elementos HTML

O que são: Tags são os blocos de construção do HTML. Cada tag define a função de um conteúdo específico, criando uma "lógica semântica" do HTML.

Exemplos de Tags Comuns:

- Cabeçalhos (<h1>a <h6>): Indicam a importância e hierarquia do conteúdo.

HTML->

```
<h1>Título Principal</h1>  
<h2>Subtítulo</h2>
```

- Parágrafos (<p>): Usados para texto corrido.

HTML->

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

- Links (<a>): Cria links para outras páginas ou partes da página.

HTML->

```
<a href="https://www.exemplo.com">Visite o nosso site</a>
```

- Imagens (): Inserem imagens na página.

HTML->

```
`: Define o conteúdo principal do documento. Deve conter o que é único da página e não se repetir em outras páginas, como o conteúdo central.

HTML->

```
<main>
 <article>
 <h2>Artigo Importante</h2>
 <p>Este é o conteúdo principal do artigo.</p>
 </article>
</main>
```

- `<article>`: Usado para conteúdos independentes e auto-contidos, como artigos, blogs ou notícias.
- `<section>`: Representa uma seção genérica do conteúdo, que pode agrupar elementos com um tema semelhante.

HTML->

```
<section>
 <h3>Seção de Notícias</h3>
 <p>Atualizações recentes e novidades.</p>
</section>
```

- `<aside>`: Contém informações adicionais que são relacionadas ao conteúdo principal, mas que não fazem parte do fluxo principal (ex. barra lateral).
- `<footer>`: Defina o rodapé do documento ou de uma seção, geralmente incluindo informações de contato, direitos autorais ou links importantes.



# FUNDAMENTOS DE LÓGICA DE PROGRAMAÇÃO - HTML

## 4. Atributos HTML

O que são: Atributos que fornecem informações adicionais sobre elementos HTML, controlando características e comportamentos específicos.

Exemplos de Atributos Comuns:

- id: Identifica um elemento de forma única.

HTML->

```
<div id="cabecalho">Cabeçalho da Página</div>
```

- class: Atribui uma classe a um elemento, facilitando a aplicação de estilos CSS ou manipulação com JavaScript.

HTML->

```
<p class="destaque">Este texto é importante.</p>
```

- href: Especifica o destino de um link.

HTML->

```
Visite o site
```

- src: Define a URL da imagem a ser exibida.

HTML->

```

```

- alt: fornecer texto alternativo para imagens, importante para acessibilidade.

# FUNDAMENTOS DE LÓGICA DE PROGRAMAÇÃO - HTML

## 5. Listas em HTML

O que são: Listas ajudam a organizar informações de forma estruturada. Existem três tipos principais:

- Listas Não Ordenadas ( `<ul>`): Crie listas de itens com marcadores.

HTML->

```

 Item 1
 Item 2
 Item 3

```

- Listas Ordenadas ( `<ol>`): Cria listas numeradas.

HTML->

```

 Primeiro passo
 Segundo passo
 Terceiro passo

```

- Listas de Definição ( `<dl>`): Usadas para listas de termos e suas definições.

HTML->

```
<dl>
 <dt>HTML</dt>
 <dd>Linguagem de Marcação de Hipertexto.</dd>
 <dt>CSS</dt>
 <dd>Folhas de Estilo em Cascata.</dd>
</dl>
```

# FUNDAMENTOS DE LÓGICA DE PROGRAMAÇÃO - HTML

## 6. Formulários em HTML

O que são: Formulários permitem a entrada de dados pelo usuário. São compostos por elementos como caixas de texto, botões e menus de seleção.

Exemplos de Elementos de Formulário:

- Campo de texto ( `<input type="text">`):

HTML->

```
<form>
 <label for="nome">Nome: </label>
 <input type="text" id="nome" name="nome">
</form>
```

- Listas Ordenadas ( `<ol>`): Cria listas numeradas.

HTML->

```
<button type="submit">Enviar</button>
```

- Caixa de seleção :

HTML->

```
<input type="checkbox" id="aceito" name="termos">
<label for="aceito">Aceito os termos e condições</label>
```

- Menu de seleção ( `<select>`):

HTML->

```
<select name="cor">
 <option value="vermelho">Vermelho</option>
 <option value="azul">Azul</option>
 <option value="verde">Verde</option>
</select>
```



# FUNDAMENTOS DE LÓGICA DE PROGRAMAÇÃO - HTML

## 7. Links e Navegação

O que são: Links são essenciais para a navegação entre páginas ou seções de um site. Eles são criados usando a tag <a>.

Exemplos de Navegação:

- Link para outra página :

HTML->

```
Ir para a Página 2
```

- Menu de seleção ( <select>):

HTML->

```
Sobre Nós
...
<section id="sobre">
 <h2>Sobre Nós</h2>
 <p>Informações sobre a empresa.</p>
</section>
```

## 7. Links e Navegação

O que são: Links são essenciais para a navegação entre páginas ou seções de um site. Eles são criados usando a tag <a>.

Exemplos de Navegação:

- Link para outra página :

HTML->

```
Sobre Nós
...
<section id="sobre">
 <h2>Sobre Nós</h2>
 <p>Informações sobre a empresa.</p>
</section>
```

# FUNDAMENTOS DE LÓGICA DE PROGRAMAÇÃO - HTML

## 8. Tabelas no HTML

O que são: Tabelas são usadas para organizar dados em linhas e colunas. Embora não sejam recomendados para layout, são úteis para apresentar informações tabulares.

Exemplo de Tabela Simples:

HTML->

```
<table>
 <tr>
 <th>Nome</th>
 <th>Idade</th>
 <th>Cidade</th>
 </tr>
 <tr>
 <td>Ana</td>
 <td>28</td>
 <td>São Paulo</td>
 </tr>
 <tr>
 <td>Carlos</td>
 <td>35</td>
 <td>Rio de Janeiro</td>
 </tr>
</table>
```

# FUNDAMENTOS DE LÓGICA DE PROGRAMAÇÃO - HTML

## 9. Estrutura e Hierarquia do HTML

O que é: A hierarquia de conteúdo refere-se à organização visual e estrutural dos elementos HTML, enquanto a herança define como estilos e comportamentos são passados de elementos pai para filho.

Práticas Importantes:

- Usar tags de cabeçalho ( <h1>a <h6>) de formato adequado para criar uma posição clara.
- Agrupar conteúdo relacionado a elementos semânticos como <section>, <article>, <aside>.
- A aplicação correta de estilos CSS e comportamentos JavaScript é facilitada por uma classificação HTML bem definida.

## 10. Multimídia no HTML

O que é: HTML permite a incorporação de conteúdo multimídia, como áudio, vídeo e gráficos, usando elementos específicos.

Exemplos:

- Imagens ( <img>): Inserção de imagens, já discutida anteriormente.
- Áudio ( <audio>): Usado para incorporar áudio.

HTML->

```
<audio controls>
 <source src="audio.mp3" type="audio/mpeg">
 Seu navegador não suporta o elemento de áudio.
</audio>
```



# FUNDAMENTOS DE LÓGICA DE PROGRAMAÇÃO - HTML

- Vídeo ( <video>): Usado para incorporar vídeos.

HTML->

```
<video width="320" height="240" controls>
 <source src="video.mp4" type="video/mp4">
 Seu navegador não suporta o elemento de vídeo.
</video>
```

## 11. Elementos de Estilo Inline

O que é: Em HTML, é possível aplicar estilos diretamente aos elementos usando o atributo style. Embora o CSS seja uma abordagem recomendada para estilização, style é útil para ajustes rápidos.

Exemplo:

HTML->

```
<p style="color: blue; font-size: 20px;">Este é um texto estilizado em azul
com tamanho 20px.</p>
```

## 12. Elementos de script (JavaScript)

O que é: HTML permite a integração de JavaScript para criar comportamentos dinâmicos e interatividade na página. O script pode ser incluído diretamente no HTML ou vinculado a um arquivo externo.

Exemplo:

HTML->

```
<button onclick="alert('Você clicou no botão!')">Clique Aqui</button>
```

# FUNDAMENTOS DE LÓGICA DE PROGRAMAÇÃO - HTML

- Uso de JavaScript Externo:

HTML->

```
<script src="script.js"></script>
```

## 13. Meta Tags e SEO (Search Engine Optimization)

O que é: Meta tags fornecem informações adicionais sobre a página para navegadores e motores de busca, influenciando o SEO.

Exemplos Comuns:

- Meta Descrição: Defina uma breve descrição da página.

HTML->

 Copiar código

```
<meta name="description" content="Esta é uma página sobre a importância do HTML.">
```

- Meta Keywords: Palavras-chave que descrevem o conteúdo.

HTML->

```
<meta name="keywords" content="HTML, programação, web design">
```

- Meta Viewport: controla o comportamento da visualização em dispositivos móveis.

HTML->

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

## 14. Iframe e Conteúdos Incorporados

O que é: O elemento <iframe> permite incorporar conteúdo de outras fontes ou páginas dentro da página atual.

Exemplos:

Exibir um vídeo do YouTube:

# FUNDAMENTOS DE LÓGICA DE PROGRAMAÇÃO - HTML

HTML->

```
<iframe width="560" height="315" src="https://www.youtube.com/embed/video_id"
frameborder="0" allowfullscreen></iframe>
```

- Incorporar outra página web:

HTML->

```
<iframe src="https://www.exemplo.com" width="600" height="400"></iframe>
```

## 15. Elementos Interativos Avançados

O que é: HTML oferece elementos para criar componentes interativos como menus suspensos, abas e painéis expansíveis.

Exemplos:

- Detalhes e Resumo ( <details>e <summary>): Usados para criar conteúdo expansível.

HTML->

 Copiar código

```
<details>
 <summary>Mais Informações</summary>
 <p>Este texto será exibido quando o usuário clicar em "Mais Informações".</p>
</details>
```

- Campos de Entrada Avançados ( <input>): Tipos de entrada como date, colore range permitem entradas mais específicas.

HTML->

```
<input type="date">
<input type="color">
<input type="range" min="0" max="100">
```



## FUNDAMENTOS DE LÓGICA DE PROGRAMAÇÃO - HTML

### 16. Acessibilidade e ARIA (Aplicativos de Internet Ricos Acessíveis)

O que é: HTML combinado com ARIA permite criar páginas acessíveis para todos, incluindo pessoas com deficiências. As atribuições ARIA ajudam a melhorar a acessibilidade.

Exemplos de Atributos ARIA:

- `aria-label`: forneça um rótulo para um elemento.

HTML->

```
<button aria-label="Fechar">X</button>
```

- `role`: Definir o papel de um elemento, ajudando tecnologias assistivas a entender seu propósito.

HTML->

```
<div role="navigation">Menu de Navegação</div>
```

- `aria-hidden`: Indica que o conteúdo não deve ser acessível aos leitores de tela.

HTML->

```
<div aria-hidden="true">Conteúdo Invisível para Leitores de Tela</div>
```

# FUNDAMENTOS DE LÓGICA DE PROGRAMAÇÃO - HTML

## 17. Boas Práticas de HTML e Otimização

O que é: Seguir boas práticas de HTML garante que uma página seja eficiente, acessível e fácil de manter. Isso inclui:

- Estrutura clara : Use tags semânticas e evite anular elementos desnecessariamente.
- Evitar o uso de tabelas para layout : Utilize tabelas apenas para dados tabulares.
- Reduzir a dependência de elementos inline : Prefira CSS externo para estilização.
- Otimização de Imagens : Utilize formatos otimizados e atributos `width` e `height` para imagens.
- Uso de tags alternativas : Sempre forneça textos alternativos para imagens e multimídia.

## 18. Estrutura de Código HTML Limpo e Comentários

O que é: Manter o código HTML limpo e bem comentado facilita a manutenção e a compreensão por outros desenvolvedores.

Exemplo de comentário:

```
HTML->
```

```
<!-- Este é um comentário que não será exibido na página -->
```

# FUNDAMENTOS DE LÓGICA DE PROGRAMAÇÃO - HTML

## 19. Validação e Compatibilidade entre Navegadores

O que é: É importante validar o código HTML para garantir que ele siga os padrões web, tornando-o compatível com diferentes navegadores.

Validação de HTML:

- Use ferramentas como o W3C HTML Validator para verificar se o código está de acordo com os padrões.
- Teste a página em diferentes navegadores (Chrome, Firefox, Safari, Edge) para garantir que o design e a funcionalidade sejam consistentes.

Com esses fundamentos, você estará bem preparado para criar páginas web robustas, acessíveis e bem estruturadas, compreendendo desde os conceitos básicos até as práticas avançadas de HTML. Esses elementos formam a base necessária para avançar no desenvolvimento web, integrando CSS e JavaScript para criar sites mais interativos e dinâmicos.



## REFERÊNCIAS

DUCKETT, Jon. HTML e CSS: Projete e crie sites . 1. ed. Indianápolis: John Wiley & Sons, 2011.

KRUG, Steve. Não me faça pensar: uma abordagem de senso comum para usabilidade na Web . 3. ed. Berkeley: New Riders, 2014.

ROBBINS, Jennifer Niederst. Aprendendo Web Design: Um Guia para Iniciantes em HTML, CSS, JavaScript e Web Graphics . 5. ed. Sebastopol: O'Reilly Media, 2018.

FELKE-MORRIS, Terry. Fundamentos de desenvolvimento e design web com HTML5 . 9. ed. Boston: Pearson, 2018.

FRAIN, Ben. Web design responsivo com HTML5 e CSS3 . 3. ed. Birmingham: Packt Publishing, 2020.