

Guia Completo: HTML Básico

Este material cobre os conceitos essenciais de HTML (HyperText Markup Language) para começar a criar páginas web. Inclui estrutura, tags mais usadas, exemplos práticos, boas práticas e referência rápida.

1. O que é HTML?

HTML é uma linguagem de marcação usada para estruturar conteúdo na web. Um documento HTML contém elementos (tags) que descrevem diferentes partes do conteúdo, como títulos, parágrafos, imagens e links.

2. Estrutura básica de um documento HTML

Todo documento HTML segue uma estrutura padrão. Aqui está um exemplo mínimo funcional:

```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Exemplo Básico</title>
  </head>
  <body>
    <h1>Olá, mundo!</h1>
    <p>Este é um documento HTML básico.</p>
  </body>
</html>
```

3. Elementos e Tags Comuns

As tags HTML vêm em pares (abertura e fechamento) ou são 'self-closing'. Exemplo: **<p>...</p>** para parágrafos.

Lista das tags mais usadas:

Cabeçalhos: **<h1>** até **<h6>** — definem importância e tamanho visual.

Parágrafo: **<p>** — bloco de texto.

Link: **texto** — cria hiperlink.

Imagem: **** — insere imagem (atributo alt é obrigatório para acessibilidade).

Lista não ordenada: ** item ... **

Lista ordenada: ** item ... **

Tabela: **<table>**, **<thead>**, **<tbody>**, **<tr>**, **<th>**, **<td>**

Formulário: **<form>**, **<input>**, **<textarea>**, **<select>**, **<button>**

Comentários: **<!-- comentário -->** — não exibido pelo navegador.

4. Atributos importantes

Atributos adicionam informações às tags. Exemplo: ****

Alguns atributos frequentes:

id — identificador único; class — classes para estilização; src — fonte de imagens/scripts; alt — descrição alternativa; href — URL para links; style — estilos inline; title — texto de dica (tooltip).

5. Imagens e mídia

Imagens: utilize sempre o atributo alt. Exemplo:

```

```

Vídeos e Áudio: use as tags <video> e <audio> com controles nativos:

```
<video src="video.mp4" controls width="640"></video>
<audio src="musica.mp3" controls></audio>
```

6. Links e navegação

Links absolutos apontam para outros domínios; relativos apontam para arquivos no mesmo site.

```
<!-- Link absoluto -->
<a href="https://www.example.com">Externo</a>

<!-- Link relativo -->
<a href="pasta/pagina.html">Página interna</a>

<!-- Abrir em nova aba -->
<a href="https://example.com" target="_blank" rel="noopener noreferrer">Abrir</a>
```

7. Listas e tabelas

Listas organizam itens; tabelas mostram dados tabulares.

```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
</ul>

<table>
  <thead>
    <tr><th>Nome</th><th>Idade</th></tr>
  </thead>
  <tbody>
    <tr><td>Ana</td><td>30</td></tr>
    <tr><td>Bruno</td><td>25</td></tr>
  </tbody>
</table>
```

8. Formulários (Forms)

Formulários coletam dados do usuário. Importante: sempre validar no servidor mesmo se houver validação no cliente.

```
<form action="/enviar" method="post">
  <label for="nome">Nome:</label>
  <input type="text" id="nome" name="nome" required>

  <label for="email">Email:</label>
  <input type="email" id="email" name="email">
  <label for="mensagem">Mensagem:</label>
```

```
<textarea id="mensagem" name="mensagem"></textarea>

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

Tipos de input comuns: text, email, password, number, checkbox, radio, file, hidden, date, url, tel, search.

9. Semântica HTML

Tags semânticas ajudam na organização e acessibilidade: <header>, <nav>, <main>, <section>, <article>, <aside>, <footer>.

Usar semântica facilita SEO e torna o conteúdo mais compreensível para leitores de tela.

10. Meta tags e

A seção <head> contém metadados: título, codificação, viewport, links para CSS, scripts e meta descriptions.

```
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta name="description" content="Descrição da página para motores de busca.">
  <title>Título da página</title>
  <link rel="stylesheet" href="styles.css">
</head>
```

11. Acessibilidade (a11y)

Boas práticas de acessibilidade:

- Sempre usar atributos alt em imagens.
- Usar labels associados a inputs (for e id).
- Estruturar conteúdo com headings em ordem (h1, h2...).
- Evitar apenas cores para transmitir informação; fornecer texto alternativo.
- Testar com leitor de tela e navegação por teclado.

12. Introdução ao CSS (integração básica)

HTML estrutura; CSS estiliza. Três formas de aplicar CSS:

- Inline: style="..." (não recomendado).
- Interno: tag <style> dentro do head.
- Externo: arquivo .css referenciado com <link> (recomendado).

```
<!-- Externo -->
<link rel="stylesheet" href="styles.css">

/* Exemplo CSS */
body { font-family: Arial, sans-serif; line-height: 1.5; }
h1 { color: #1a73e8; }
```

13. Introdução ao JavaScript (integração básica)

JavaScript adiciona interatividade. Pode ser incluído no final do body ou em arquivo externo.

```
<!-- Externo -->
<script src="script.js"></script>

<!-- Inline -->
<script>
    document.getElementById('botao').addEventListener('click', function() {
        alert('Clicou!');
    });
</script>
```

14. Boas práticas e organização

- Separe estrutura (HTML), apresentação (CSS) e comportamento (JS).
- Use nomes de classes claros e consistentes.
- Comentários úteis no código; não escrever comentários óbvios demais.
- Estruture pastas: /css, /js, /images, /pages.
- Otimize imagens para web e use formatos modernos (WebP quando possível).

15. SEO básico

- Use títulos descritivos (<title>) e meta description.
- Utilize headings corretamente e conteúdo legível.
- Inclua atributos alt em imagens e links com texto claro.
- Estruture URLs amigáveis e utilize sitemap quando o site crescer.

16. Exemplo prático: Página de portfólio simples

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Portfólio - Seu Nome</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <header>
        <h1>Seu Nome</h1>
        <nav>
            <a href="#sobre">Sobre</a>
            <a href="#projetos">Projetos</a>
            <a href="#contato">Contato</a>
        </nav>
    </header>

    <main>
        <section id="sobre">
            <h2>Sobre</h2>
            <p>Descrição curta sobre você.</p>
        </section>
        <section id="projetos">
```

```
<h2>Projetos</h2>
<article>
  <h3>Projeto 1</h3>
  <p>Descrição do projeto.</p>
</article>
</section>

<section id="contato">
  <h2>Contato</h2>
  <form action="#" method="post">
    <label for="nome">Nome</label>
    <input id="nome" name="nome" type="text">
    <button type="submit">Enviar</button>
  </form>
</section>
</main>

<footer>
  <p>© 2025 Seu Nome</p>
</footer>

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

17. Referência rápida: Atalhos e lembretes

- Comentário: <!-- ... -->
- Espaço de codificação: prefira UTF-8 (<meta charset="utf-8">)
- Abrir link em nova aba: target="_blank" rel="noopener noreferrer"
- Input obrigatório: required
- Desativar autocomplete: autocomplete="off"

18. Próximos passos para estudo

Depois de dominar HTML básico, avance para:

- CSS avançado (Flexbox, Grid, animações)
- JavaScript (DOM, Fetch, eventos, ES6+)
- Ferramentas de build (npm, bundlers) e controle de versão (Git)
- Acessibilidade aprofundada e performance web

19. Recursos úteis

- Documentação MDN Web Docs
- Cursos e tutoriais interativos (ex.: freeCodeCamp, Codecademy)
- Praticar criando pequenos projetos e clonando layouts simples

20. Exercício prático sugerido

Crie uma página com:

- 1) Cabeçalho com nome e menu de navegação (links âncora).
- 2) Seção 'Sobre' com foto e texto.
- 3) Seção 'Projetos' com pelo menos 3 cards (cada um com título, imagem e descrição).
- 4) Seção 'Contato' com formulário que valida o email no cliente.
- 5) Estilize com CSS básico (cores, fonte, espaçamento) e torne responsivo com media queries.

