

Apostila 1

HTML – A Linguagem de Marcação de Hipertexto

Seção 1: Introdução ao HTML

1.1. Definição: O que é HTML?

HTML é a sigla para *HyperText Markup Language*, ou Linguagem de Marcação de Hipertexto. Esta linguagem é o bloco de construção mais fundamental da World Wide Web. Ela não é uma linguagem de programação, mas sim uma linguagem de *marcação*, o que significa que seu propósito é definir o *significado* e a *estrutura* do conteúdo de uma página web.

O HTML utiliza um sistema de "marcação" (markup) para anotar texto, imagens e outros tipos de conteúdo, permitindo que um navegador (como Google Chrome, Firefox ou Safari) interprete e exiba esse conteúdo de forma estruturada.

O termo "Hipertexto" refere-se à capacidade do HTML de conectar páginas web entre si através de links (hiperlinks). Esses links são um aspecto fundamental da Web, permitindo a navegação entre diferentes documentos e sites.

1.2. O Papel do HTML vs. CSS e JavaScript

Para desenvolver um website "bem estruturado", é imperativo compreender o princípio da "Separação de Preocupações" (Separation of Concerns). O desenvolvimento web moderno baseia-se em uma tríade de tecnologias, cada uma com uma responsabilidade distinta:

1. **HTML (Estrutura):** O HTML é o "esqueleto" do site. Sua única responsabilidade é organizar o conteúdo e dar-lhe significado semântico. Perguntas que o HTML responde: "Isto é um título principal?", "Isto é um parágrafo?", "Isto é uma lista de itens?".
2. **CSS (Apresentação):** O CSS (Cascading Style Sheets) é o "estilo" do site. Sua responsabilidade é a aparência, a apresentação e o layout. Perguntas que o CSS responde: "Qual a cor desse título?", "Qual o tamanho da fonte desse parágrafo?",

"Como essa lista deve ser alinhada na página?".

3. **JavaScript (Comportamento):** O JavaScript é a "interatividade" do site. Sua responsabilidade é a funcionalidade dinâmica. Perguntas que o JavaScript responde: "O que acontece quando o usuário clica neste botão?", "Como validar este formulário?".

Um site "bem estruturado" mantém essas três camadas o mais independentes possível. Esta apostila foca exclusivamente na primeira camada: a estrutura HTML.

Exercícios para testar o aprendizado

1. O que significa a sigla HTML e qual é o seu principal propósito na web?
2. Qual é a diferença fundamental entre HTML, CSS e JavaScript? Descreva a "responsabilidade" de cada um.
3. Por que o HTML é considerado uma linguagem de "marcação" e não de "programação"?

Seção 2: A Estrutura Fundamental de um Documento HTML

2.1. A Anatomia do Documento

Todo documento HTML, independentemente de sua complexidade, segue uma estrutura básica. Considere este o "esqueleto" inicial para qualquer página web:

```
HTML
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>
  </body>
</html>
```

Vamos analisar cada parte:

- `<!DOCTYPE html>`: Esta é a declaração de tipo de documento (doctype). Ela informa ao navegador que o documento deve ser interpretado usando o padrão HTML5 (a versão moderna do HTML).
- `<html></html>`: Este é o elemento raiz. Ele envolve todo o conteúdo da página, com exceção da declaração `<!DOCTYPE>`.
- `<head></head>`: Este elemento atua como um contêiner para metadados, ou seja, informações sobre o documento que não são exibidas diretamente na tela para o usuário (como o título da aba, a codificação de caracteres e links para arquivos CSS).
- `<body></body>`: Este elemento contém *todo* o conteúdo que será exibido ao usuário na janela do navegador, como textos, imagens, vídeos e links.

2.2. Detalhando o `<head>`: O Cérebro do Documento

O elemento `<head>` é crucial para um site funcional e bem estruturado. Embora seu conteúdo não seja visível, ele instrui o navegador sobre como lidar com a página. Os elementos essenciais dentro do `<head>` incluem:

- `<meta charset="utf-8">`: Este elemento meta define o conjunto de caracteres do documento como UTF-8. Isso é vital para garantir que o navegador exiba corretamente todos os caracteres, incluindo acentuação (como em "Apostila") e símbolos de diferentes idiomas.
- `<title></title>`: O elemento `<title>` define o título da página. Este texto é exibido na aba do navegador, nos favoritos (bookmarks) e é de extrema importância para os motores de busca (SEO).
- `<link rel="stylesheet" href="style.css">`: Esta é a forma correta e preferida de conectar sua folha de estilo CSS externa ao seu documento HTML. Este tópico será o foco da Apostila 2.
- `<meta name="viewport" content="width=device-width, initial-scale=1">`: Esta é uma das tags meta mais importantes para o desenvolvimento moderno.

O primeiro passo para a **Responsividade** (um requisito da sua solicitação) começa aqui, no HTML. Sem esta linha, os navegadores de dispositivos móveis assumirão que seu site é uma página de desktop e tentarão exibi-la em uma tela virtual ampla, "diminuindo o zoom" e tornando o site ilegível.

Esta tag instrui o navegador:

1. `width=device-width`: Define a largura da "viewport" (a área de visualização) como a

largura física real do dispositivo.

2. `initial-scale=1`: Defina o nível de zoom inicial como 100% (sem zoom).

Sem esta tag no `<head>`, todas as técnicas de responsividade do CSS (que veremos na Apostila 2) podem não funcionar como o esperado em dispositivos móveis.

Exercícios para testar o aprendizado

1. Quais são as 4 tags essenciais que formam o "esqueleto" básico de todo documento HTML?
2. Qual é a diferença entre o que vai dentro da tag `<head>` e o que vai dentro da tag `<body>`?
3. Qual é a finalidade da tag `<meta name="viewport"...">`? O que acontece em um celular se você *não* incluir esta tag?
4. **Prática:** Crie um novo arquivo chamado `index.html`. Escreva a estrutura básica do HTML (com `<!DOCTYPE>`, `<html>`, `<head>`, `<body>`, `<meta charset="utf-8">`, `<meta name="viewport">` e um `<title>`). Salve e abra-o no navegador. O que você vê na tela? E o que você vê na aba do navegador?

Seção 3: Tags Essenciais de Conteúdo e Texto

Agora que temos a estrutura, vamos preencher o `<body>` com conteúdo usando as tags HTML mais comuns.

3.1. Hierarquia de Títulos (`<h1>` a `<h6>`)

Títulos são usados para definir a hierarquia e a estrutura de tópicos do seu conteúdo.

- `<h1></h1>`: O título principal da página. Deve haver apenas um `<h1>` por página, descrevendo seu propósito principal.
- `<h2></h2>`: Subtítulos de seções principais.
- `<h3></h3>` a `<h6></h6>`: Subtítulos de níveis inferiores.

É crucial entender que os títulos são usados por seu *valor semântico* (significado estrutural), não por sua aparência (tamanho da fonte). Um site "bem estruturado" usa os títulos em ordem lógica (ex: um <h2> deve ser seguido por um <h3>, não por um <h5>). Esta prática é vital para a acessibilidade (leitores de tela usam títulos para navegação) e para o SEO.

3.2. Parágrafos (<p>)

A tag <p></p> é usada para envolver blocos de texto, como este parágrafo. O navegador automaticamente adiciona espaçamento (margem) antes e depois de cada parágrafo.

3.3. Links (<a>)

A tag de âncora <a> é o que torna a web "hipertexto". Ela é usada para criar links para outras páginas ou recursos. Seu atributo essencial é o href (*Hypertext Reference*), que especifica o URL de destino.

Exemplo:

HTML

```
<p>Visite o <a href="https://www.google.com">Google</a> para pesquisar.</p>
```

3.4. Imagens ()

A tag é usada para incorporar imagens em uma página. Note que esta é uma "tag vazia" ou "void element", o que significa que ela não possui uma tag de fechamento (como).

Ela requer dois atributos essenciais:

1. src (*Source*): O caminho ou URL para o arquivo de imagem.
2. alt (*Alternative Text*): Um texto descritivo para a imagem.

Exemplo:

HTML

```

```

Para um site "bem estruturado", o atributo alt *não é opcional*. Ele é fundamental por duas razões:

1. **Acessibilidade:** Leitores de tela leem o texto alt para usuários com deficiência visual, descrevendo o que está na imagem.
2. **SEO:** Motores de busca usam o alt para entender o conteúdo da imagem.

Exercícios para testar o aprendizado

1. Explique por que os títulos (<h1>-<h6>) devem ser usados por seu valor semântico (hierarquia) e não por sua aparência (tamanho).
2. Quais são os dois atributos *essenciais* da tag e por que o atributo alt é tão importante?
3. Qual atributo da tag <a> define o destino do link?
4. **Prática:** Adicione ao <body> do seu arquivo index.html (criado na Seção 2) um título principal (<h1>), dois parágrafos (<p>), um link (<a>) para seu site favorito e uma imagem (- pode ser de um link externo, se você ainda não tiver imagens locais).

Seção 4: Organização e Estruturação de Conteúdo

4.1. Listas

O HTML fornece tags específicas para listas.

Listas Não Ordenadas ()

Usadas quando a ordem dos itens não importa (listas com marcadores, ou *bullets*). A tag (Unordered List) envolve a lista, e cada item é definido pela tag (List Item).

Exemplo:

HTML

```
<ul>
  <li>Maçã</li>
  <li>Laranja</li>
  <li>Banana</li>
</ul>
```

Listas Ordenadas ()

Usadas quando a ordem sequencial dos itens é importante (listas numeradas). A tag (Ordered List) envolve a lista, e os itens também usam .

Exemplo:

HTML

```
<ol>
  <li>Primeiro passo</li>
  <li>Segundo passo</li>
  <li>Terceiro passo</li>
</ol>
```

As listas ordenadas podem ter atributos como type (para mudar o tipo de numeração, ex: type="A" para letras maiúsculas) ou start (para começar a numeração de um número diferente).

4.2. Tabelas de Dados

Tabelas são usadas *exclusivamente* para exibir dados tabulares (como uma planilha). Uma tabela básica é composta por:

- <table></table>: O elemento que envolve toda a tabela.
- <tr></tr>: (Table Row) Define uma linha da tabela.

- `<th></th>`: (Table Header) Define uma célula de cabeçalho (normalmente em negrito e centralizada). É usada na primeira linha.
- `<td></td>`: (Table Data) Define uma célula de dados padrão.

Exemplo:

HTML

```
<table>
<tr>
  <th>Nome</th>
  <th>Idade</th>
</tr>
<tr>
  <td>Maria</td>
  <td>30</td>
</tr>
<tr>
  <td>João</td>
  <td>25</td>
</tr>
</table>
```

Nota Importante: No passado, desenvolvedores usavam tabelas HTML para criar o *layout* de uma página (ex: criar colunas e barras laterais). Esta é uma prática obsoleta e um exemplo canônico de um site *mal estruturado*. O layout de uma página deve ser controlado exclusivamente pelo CSS, usando técnicas como Flexbox e Grid (abordadas na Apostila de CSS). Use tabelas HTML *apenas* para dados tabulares.

Exercícios para testar o aprendizado

1. Qual é a diferença entre uma lista ordenada (``) e uma lista não ordenada (``)?
 2. Qual tag é usada para definir um item de lista, independentemente do tipo de lista?
 3. Para que servem as tags `<table>`, `<tr>`, `<th>` e `<td>`?
 4. Qual é a prática *obsoleta* para a qual as tabelas eram usadas e por que ela não deve mais ser feita?
 5. **Prática:** Adicione ao seu `index.html` uma lista não ordenada de suas comidas favoritas e uma tabela simples com duas colunas (Ex: Produto, Preço) e três linhas de dados.
-

Seção 5: HTML Semântico – O Caminho para um Site "Bem Estruturado"

Este é o conceito mais importante para atender ao requisito de um site "bem estruturado".

5.1. O que é HTML Semântico?

HTML Semântico é o uso de tags HTML de acordo com seu *significado* (semântica), e não por sua *aparência*.

Por exemplo, um iniciante pode usar `<div>` (uma tag não semântica, genérica) para tudo e estilizá-la com CSS para *parecer* um título. Um desenvolvedor profissional usa a tag `<h1>`, que informa explicitamente ao navegador, aos motores de busca e aos leitores de tela: "Este é o título mais importante da página".

5.2. Por que usar Semântica?

Usar tags como `<div>` e `` para tudo resulta em um código ilegível e "burro". Um site semântico é superior por três razões principais:

1. **SEO (Otimização para Motores de Busca):** Motores de busca (como o Google) entendem a estrutura de um site semântico. O conteúdo dentro de uma tag `<main>` ou `<article>` tem mais relevância do que um `<div>` genérico.
2. **Acessibilidade:** Usuários com deficiência visual dependem de leitores de tela. Esses leitores usam tags semânticas como "placas de sinalização" para navegar. Um leitor de tela pode dizer: "Há um cabeçalho, um menu de navegação e um conteúdo principal. Deseja pular para o conteúdo principal?". Isso é impossível se a página for feita apenas de `<div>s`.
3. **Manutenibilidade:** O código é imensamente mais fácil de ler e atualizar, tanto para você quanto para outros desenvolvedores.

5.3. Tags de Layout Estrutural

O HTML5 introduziu tags semânticas projetadas especificamente para o layout da página.

Elas substituem o uso excessivo de <div>s.

- <header>: Representa o conteúdo introdutório ou um grupo de ajuda à navegação. Geralmente contém o logotipo do site, o título principal e, às vezes, o menu de navegação.
- <nav>: Representa um bloco de links de navegação principal (ex: o menu principal do site).
- <main>: Representa o conteúdo *dominante e principal* da página. Deve ser único no documento. Não deve conter conteúdo que se repete em todas as páginas, como barras laterais ou rodapés.
- <section>: Representa uma seção temática genérica do documento. Idealmente, cada <section> deve ser identificada com um título (ex: <h2>).
- <article>: Representa um bloco de conteúdo completo e autossuficiente que poderia ser distribuído independentemente (ex: um post de blog, um artigo de notícia, um comentário de fórum).
- <aside>: Representa conteúdo tangencialmente relacionado ao conteúdo principal (ex: uma barra lateral, uma biografia do autor, links relacionados).
- <footer>: Representa o rodapé da página ou da seção. Geralmente contém informações de copyright, autoria ou links relacionados.

5.4. Exemplo de Estrutura Semântica

Juntando tudo, um esqueleto de site "bem estruturado" se parece com isto:

HTML

```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Meu Site Semântico</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>

    <header>
      <h1>Meu Website</h1>
    </header>
    <nav>
      <ul>
        <li><a href="#">Home</a></li>
        <li><a href="#">Sobre</a></li>
```

```

        <li><a href="#">Contato</a></li>
    </ul>
</nav>
</header>

<main>
  <article>
    <h2>Meu Primeiro Artigo</h2>
    <p>Este é o conteúdo principal do meu artigo...</p>
  </article>

  <aside>
    <h3>Sobre o Autor</h3>
    <p>Informações sobre o autor aqui.</p>
  </aside>
</main>

<footer>
  <p>© 2024 Meu Website. Todos os direitos reservados.</p>
</footer>
</body>
</html>

```

Esta estrutura é a base sólida sobre a qual aplicaremos os estilos na próxima apostila.

Exercícios para testar o aprendizado

1. Defina "HTML Semântico". Por que usar <header> é melhor do que usar <div class="header">?
2. Liste três benefícios de usar HTML semântico (abordados na seção).
3. Qual é a diferença de propósito entre as tags <section> e <article>?
4. Qual tag deve conter o conteúdo *principal* e *único* da página?
5. **Prática:** Pegue o conteúdo que você criou até agora no seu index.html (títulos, parágrafos, listas, tabelas) e organize-o usando a estrutura de layout semântica (tags <header>, <nav>, <main>, <aside>, <footer>).