



5954024 - Introdução ao Desenvolvimento Web

HTML - Visão Geral e Prática

Prof. Dr. Denis M. L. Martins

DCM | FFCLRP | USP

Objetivos de Aprendizagem

- **Compreender** o papel do HTML na Web e sua evolução histórica.
- **Identificar** os elementos básicos de um documento HTML.
- **Criar** uma página web simples com estrutura semântica.
- **Diferenciar** HTML de outras tecnologias (CSS, JavaScript).

O que é HTML?

- HTML = *HyperText Markup Language*
- **Linguagem de marcação**, não de programação.
 - **Não executa lógica** (diferente de JavaScript).
 - **Descreve a estrutura e o conteúdo** da página.
- **Funcionamento:**
 - Navegadores (Chrome, Firefox) interpretam as *tags* HTML para renderizar páginas.
 - Exemplo: `<h1>Título</h1>` → "Título" é exibido como um título de nível 1.

Breve História do HTML

Período	Versão	Principais Mudanças
1993	HTML 2.0	Criação por Tim Berners-Lee (CERN).
1997	HTML 4.0	Introdução de <i>tables</i> e estilos inline.
2000	XHTML	Versão XML estrita (mais rigorosa).
2014	HTML5	Multimídia, APIs modernas, semântica melhorada.

Sintaxe Básica: Elemento

- O que é um elemento?

- É a unidade atômica do HTML.
- Formado por: `<etiqueta>conteúdo</etiqueta>`
 - Etiquetas ou tags possuem atributos (opcionais)

- Exemplo:

```
<b>Texto em negrito</b>  
<!-- 0 "texto" é o conteúdo, "b" define a ação. -->
```

- Conceito Chave:

- Tudo começa com `<`. Tudo termina com `>`.
- Se você esquece o `>`, o navegador tenta corrigir (o que é um erro!).

Sintaxe Básica: Tags de Fechamento

- **div** precisa fechar: **<div>...</div>** (Tag de Bloco):

```
<p>Esse parágrafo  
    <!-- 0 navegador vai "achar" onde acaba aqui. -->  
</p>
```

- **img** não fecha:

```

```

Estrutura Básica de uma Página

- **<HTML>**: Raiz do documento. Deve ser o primeiro elemento do documento.
 - Use **<html lang="pt-br">** para definir o idioma.
- **<head>**: área de configuração. Contém os metadados do documento.
 - Metadados descrevem características do documento. Exemplo: título, descrição, palavras-chave.
- **<body>**: área visível (no navegador) do documento. Aqui, o HTML define a semântica que será "estilizada" no futuro
- **Observação**: Comece sempre com `<!DOCTYPE html>` (Avisa ao navegador que é [HTML5](#), não [XHTML](#)).



www.pensandonaweb.com.br

Fluxo de Renderização

- O navegador lê da Cima para Baixo, Esquerda para Direita.
- **Parsing do HTML**: O navegador lê e analisa o código HTML.
- **Construção da Árvore DOM (Document Object Model)**: Cria uma representação hierárquica do documento.
- **Renderização na tela**: O navegador pinta os pixels na tela com base no DOM.

Prática Rápida: Primeiro de Documento HTML

```
<!DOCTYPE html> <!-- Declaração Obrigatória -->
<html lang="pt-br">
  <head>
    <meta charset="UTF-8"> <!-- Codificação de caracteres (Especialmente para acentos) -->
    <title>Meu Site</title> <!-- Título na aba do navegador -->
  </head>

  <body>
    <!-- O que o usuário vê começa aqui -->
    <div class="conteudo">
      <h1>Olá</h1>
      <p>Este é um parágrafo.</p>
    </div>
  </body>
</html>
```

O `<meta charset="UTF-8">` é **obrigatório** na primeira linha.
Sem ele, caracteres como `ã` ou `é` podem aparecer distorcidos (ex: ã).

Metadados

- Metadados são dados que descreve, dados.
- Exemplos:

```
<!-- Configuração para Responsividade (Mobile) -->  
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<!-- Título que aparece na janela do navegador -->  
<title>Navegador de Exemplo</title>
```

- **Prática rápida:** crie uma página para exibir o código abaixo com e sem o metadado

```
<meta charset="UTF-8" />
```

```
<p>Exemplo Japonês: ご飯が熱い</p>
```

Sintaxe Básica: Fluxo de Texto

- Navegação de Texto vs Tags de Blocos:

- Inline (`<span>`, `<a>`): Não quebra linha, flui na linha atual.
- Bloco (`<div>`, `<h1>`): Ocupa largura total e quebra a linha antes/depois.

- Por que importa?

- Elementos de bloco criam "containers" visuais.
- Elementos inline servem para detalhar dentro do texto.

- Prática Rápida:

- Crie uma tag `<p>` que digite "Olá Mundo".
- Envolve o texto com tags `<b>` (negrito).

Textos e Links

- Hierarquia de Títulos (`<h1>` a `<h6>`):
 - O `h1` deve conter o título principal do documento (apenas 1 por página)
- Texto (`<p>`): Use para parágrafos.
 - **Dica:** não quebre um texto manualmente com `<br>`
- Links (`<a href="...">`):

```
<a href="https://google.com" target="_blank">Clique Aqui</a>
```

- `href`: Para onde ir? (URL).
- `target="_blank"`: Abre em nova aba.

Imagens

- Tag `<img>`: Não é um container, é autônomo.
- Importância do atributo `alt`:
 - A imagem não funciona? O leitor de tela vai ler o texto da `alt`.
- Exemplo Correto:

```

```
- **Dica:** Não use emojis como substitutos do atributo `alt` se a imagem tiver texto.

Prática Rápida: Perfil Pessoal

- **Instrução:** Crie uma estrutura básica de "Perfil Pessoal".
- **Passo-a-passo:**
 - i. Crie um arquivo `index.html`.
 - ii. Use `<header>` com seu nome em `<h1>`.
 - iii. Use um `<img>` com uma foto e `alt` descritivo. Busque uma foto em <https://unsplash.com/s/photos/profile-pic> ↗
 - iv. Use um `<footer>` com o ano de criação.
- **Objetivo:** Ter uma página completa rodando em menos de 5 minutos.

Prática Rápida: Validando HTML

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <title>Meu Site</title>
  <meta charset="UTF-8">
</head>
<body>
  <h1>Bem-vindo ao Meu Site</h1>
  <p>Este é um parágrafo com <strong>texto em negrito</strong>.
  
  <a href="https://exemplo.com">Visite Example.com</a>
  <!-- Comentário -->
</body>
</html>
```

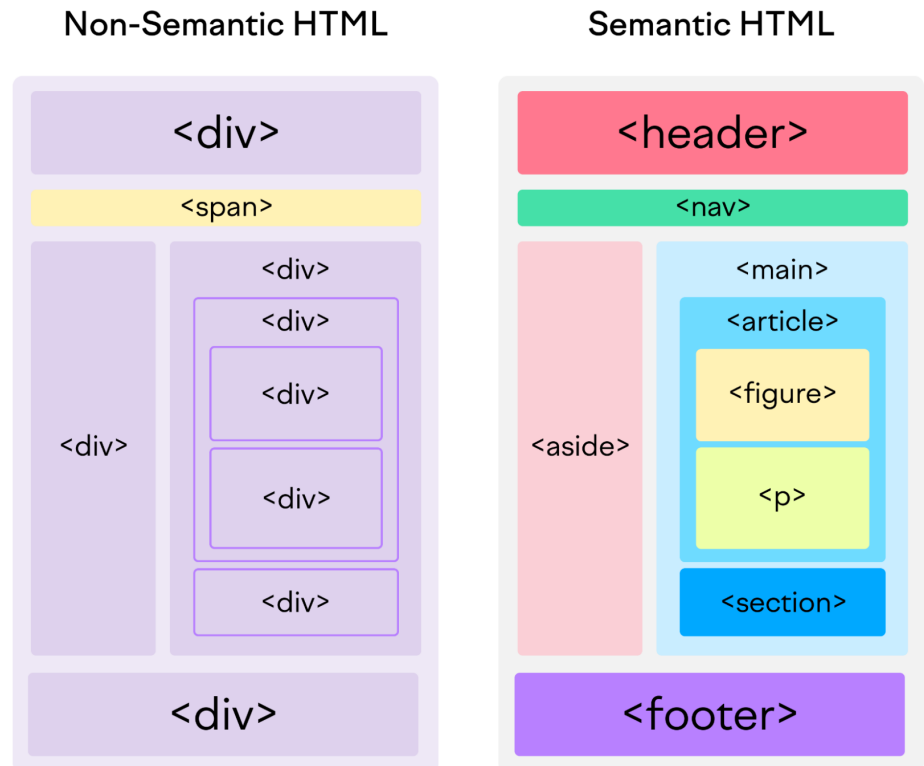
1. Copie o código e cole no [W3C Validator](#) .
2. Identifique os erros (e.g., tags mal fechadas, atributos inválidos)
3. Corrija o código localmente.
4. Valide novamente até que não haja erros.

Layout Semântico

- Não use apenas `<div>`: Isso polui o código.
- Use a estrutura lógica do corpo humano:
 - **Header:** Cabeçalho (Menu principal, Logo).
 - **Nav:** Navegação (Links de menu).
 - **Main:** Conteúdo principal da página.
 - **Footer:** Rodapé (Contatos, Copyright).
- **Benefício:** Leitores de tela e Robôs SEO entendem o site melhor.



What Is Semantic HTML?



Formulários em HTML: Exemplo

```
<form action="/submit-form" method="post">
  <label for="email">E-mail:</label>
  <input type="email" id="email" name="email" required>

  <label for="password">Senha:</label>
  <input type="password" id="password" name="password" required>

  <label for="comments">Comentários:</label>
  <textarea id="comments" name="comments" rows="4" cols="50"></textarea>

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

Formulários em HTML: Elementos

- **<form>**: Contêiner para elementos de formulário.
 - **action**: URL para onde os dados serão enviados.
 - **method**: Método HTTP (**get** ou **post**).
- **<input>**:
 - Tipos comuns: **text**, **password**, **email**, **number**, **checkbox**, **radio**.
 - Atributos importantes:
 - **type**: Define o tipo de input.
 - **name**: Nome do campo para identificação no servidor.
 - **required**: Indica que o campo é obrigatório.
- **<textarea>**: Para texto multilinha.
- **<button>**: Botões para ações (enviar, resetar).

Exercício

- Use as ferramentas de desenvolvedor do seu navegador para modificar localmente o código HTML de uma página Web de sua preferência.
- Atalho de teclado no Firefox: **Ctrl + Shift + I** ou **F12**
- Por exemplo, Use o site do DCM: <https://dcm.ffclrp.usp.br/> 

Conclusão

Conceitos-Chave:

- HTML é a linguagem de marcação que estrutura o conteúdo da web.
- Tags semânticas (`<header>`, `<footer>`, etc.) melhoram a acessibilidade e SEO.
- Atributos são fundamentais para funcionalidade e acessibilidade.
- O fluxo de processamento envolve *parsing*, construção do DOM e renderização.
- Formulários permitem coleta de dados do usuário.

Próxima aula: Introdução ao CSS (estilização).

Material adicional: [História da HTML por Gustavo Guanabara](#)   YouTube

Dúvidas e Discussão