



MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA
E INOVAÇÃO



Módulo 4: HTML + CSS

AULA 1/8 | Fundamentos e Estruturação Web

Estruturação de Documentos HTML - Aula 1 de 8

Desenvolvimento Front End em React: preparando você para o mercado de tecnologia com fundamentos sólidos e práticas profissionais.

Introdução

Olá! Eu sou o prof. Igor Vasconcelos e meu objetivo é guiar você nesta jornada fundamental. Nossa foco é a empregabilidade em Front End. Este módulo constrói a base sólida para o Módulo 5 (React), preparando você para os desafios do mercado.

O que já foi visto

Módulo 1: Fundamentos de **Lógica de Programação e JavaScript** (Algoritmos, Variáveis, Funções, Arrays/Objetos, Estruturas de Repetição).

- **Módulo 2: Noções de UX/UI** (Design Centrado no Usuário, Heurísticas de Usabilidade, **Acessibilidade na Web**, Wireframes e Design Responsivo).
- **Módulo 3: Programação Orientada a Objetos** (POO, Classes, Encapsulamento, Herança e Tratamento de Exceções).

O que veremos

Aulas 1-3: Fundação: **Estruturação (HTML)**, Formulários e Introdução ao **CSS (Seletores e Especificidade)**.

- **Aulas 4-6:** Layout Moderno: Dominaremos **Flexbox** e **CSS Grid** (ferramentas essenciais em qualquer estágio).
- **Aulas 7-8:** Evolução do CSS: **Transições/Animações** e a organização de código para o React (Pré-processadores como **SASS** e Estilização de Componentes).

A INTERNET EM AÇÃO: SERVIDOR, DNS E O PARADIGMA WEB

Entendendo a Comunicação Cliente-Servidor



A INTERNET EM AÇÃO: SERVIDOR, DNS E O PARADIGMA WEB

Entendendo a Comunicação Cliente-Servidor

Localizando e Entregando a Informação



O Cliente (Seu Navegador)

Você digita um URL (Uniform Resource Locator).



DNS (Domain Name System)

O DNS resolve o nome de domínio (ex: www.site.com) para o endereço IP do servidor correspondente. (Sem DNS, o navegador não consegue resolver o hostname e a conexão falha)



O Servidor Web

Uma vez localizado, o navegador usa um protocolo (geralmente HTTP) para solicitar a página ao servidor.



Publicação

Os arquivos HTML, CSS e JavaScript devem ser transferidos e residir no servidor Web (em uma pasta raiz, como www ou htdocs) para serem acessíveis a todos.

Programação Web (Client-Side) vs. Desktop

Aspecto	Programação Web	Programação Desktop
Execução	Roda em navegadores (cliente) e/ou em servidores (back-end).	Roda diretamente no sistema operacional do usuário
Instalação/Atualização	Não requer instalação local (acesso via URL). Atualização Centralizada	Requer instalação do software no computador. Atualização depende do usuário
Ambiente de Desenvolvimento	HTML, CSS, JavaScript, frameworks (React, Angular, Node.js, etc.).	Linguagens como C#, Java, Python, C++, com frameworks como .NET
Interface Gráfica	Renderizada pelo navegador usando HTML e CSS.	Interface nativa do sistema operacional (Windows Forms, GTK, WPF etc.)
Segurança	Controlada por protocolos (HTTPS, autenticação, CORS etc.).	Controlada localmente (firewall, permissões do sistema).

As 3 Camadas do Desenvolvimento Web

O desenvolvimento client-side baseia-se em três pilares fundamentais que trabalham juntos para criar experiências web completas. Cada camada tem seu papel específico e essencial.

1ª Camada: Informação

HTML – A camada mais importante. Marca o conteúdo dando-lhe significado semântico. Deve funcionar sempre, mesmo que as outras camadas falhem.

2ª Camada: Formatação

CSS – Controla o visual da informação e o layout. Define cores, tipografia, espaçamentos e a apresentação geral do conteúdo.

3ª Camada: Comportamento

JavaScript – Adiciona interatividade aos elementos, manipulando as características definidas pelo CSS e criando experiências dinâmicas.

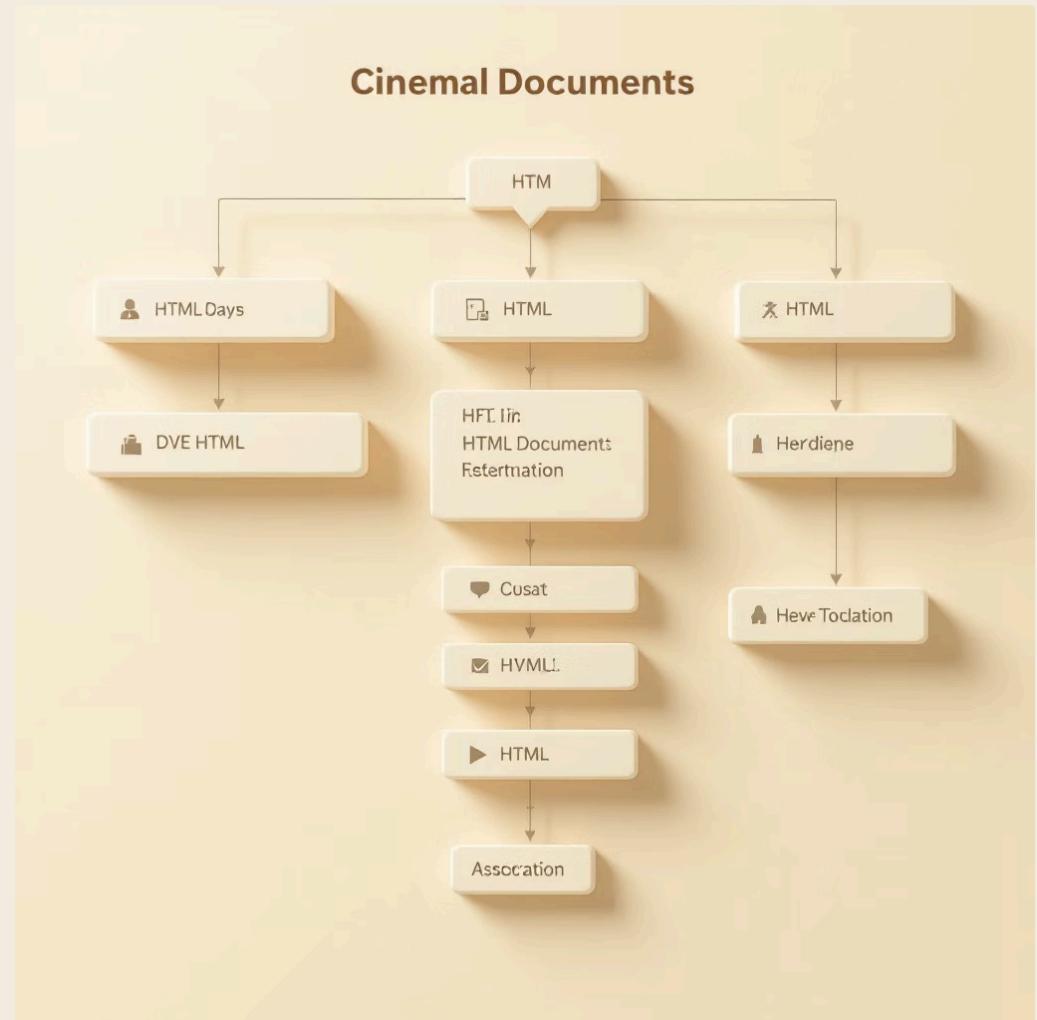
Lembre-se: **o conteúdo é o Rei**. A estrutura HTML é a base de tudo.

- ❑ O HTML define a estrutura e o significado (semântica) do documento. É a fundação que deve funcionar, independente das outras camadas ou do dispositivo.

O Esqueleto do Documento HTML5

Uma página HTML é um arquivo de texto que hospeda o conteúdo e descreve seu significado. Todos os documentos HTML devem conter os componentes base para funcionar corretamente nos navegadores.

A declaração `<!DOCTYPE html>` simplificada indica o uso do HTML5, a versão moderna e atual da linguagem de marcação.



<head> – Metadados

Contém informações sobre a página destinadas a navegadores e mecanismos de busca. É invisível para o usuário, exceto pelo título da aba.

<body> – Conteúdo Visível

Contém todo o conteúdo visível da página: textos, imagens, vídeos, formulários e todos os elementos que o usuário verá e interagirá.

Código Mínimo Requerido

Todo documento HTML5 profissional deve incluir esta estrutura básica. Copie este modelo para iniciar seus projetos com as configurações essenciais corretas.

```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <meta charset="UTF-8" />
    <title>Título descritivo da página</title>
  </head>
  <body>
    <!-- Seu conteúdo visível vai aqui -->
  </body>
</html>
```

01

DOCTYPE

Indica ao navegador que estamos usando HTML5

02

lang="pt-br"

Define o idioma principal do documento para acessibilidade

03

charset="UTF-8"

Codificação universal que suporta todos os caracteres especiais

04

title

Aparece na aba do navegador e nos resultados de busca



HTML: Hierarquia e Significado

O HTML define o **significado** do seu conteúdo através da semântica. A boa semântica melhora a acessibilidade para leitores de tela e o SEO para mecanismos de busca encontrarem seu conteúdo.

Use elementos HTML para estruturar logicamente o conteúdo, não para definir o tamanho da fonte. O visual é responsabilidade do CSS.

Cabeçalhos `<h1>` a `<h6>`

Definem a hierarquia e o esboço do documento. O `<h1>` é o título principal, seguido por subtítulos em ordem decrescente de importância.

Parágrafos `<p>`

Marca blocos de texto como parágrafos. Use `` para ênfase de tom de voz e `` para importância e relevância.

Trabalhando com Listas

Lista Não Ordenada

Use quando a ordem dos itens **NÃO é significativa**. Ideal para listas de recursos, ingredientes ou itens sem sequência específica.

```
<ul>
  <li>HTML</li>
  <li>CSS</li>
  <li>JavaScript</li>
</ul>
```

Resultado: marcadores (bullets) aparecem automaticamente.

Lista Ordenada

Use quando a ordem dos itens **É significativa**. Ideal para passos de um tutorial, rankings ou sequências que devem ser seguidas.

```
<ol>
  <li>Estruture com HTML</li>
  <li>Estilize com CSS</li>
  <li>Adicione JavaScript</li>
</ol>
```

Resultado: numeração automática na ordem correta.

- **Dica profissional:** Nunca adicione símbolos de marcadores ou números manualmente dentro dos itens – eles são gerados automaticamente pelo navegador.

Links: O Poder do Hipertexto

O elemento `<a>` (âncora) é o mais poderoso do HTML. É o que transforma a web em Hipertexto, permitindo a navegação entre páginas e criando a estrutura interconectada da internet.

1

Elemento `<a>`

O elemento âncora que cria o link clicável

2

Atributo `href`

Define o destino (URL) para onde o link aponta

3

Texto do Link

Conteúdo visível e clicável entre as tags

```
<a href="https://www.exemplo.com">Clique Aqui</a>
```

Links podem apontar para páginas externas, páginas internas do seu site, seções específicas da mesma página ou até arquivos para download.

Imagens e Atributos Essenciais

A imagem é um elemento **vazio** (self-closing) que apenas referencia um arquivo externo.

Não possui conteúdo interno, apenas atributos que definem seu comportamento.

O atributo **alt** (texto alternativo) é crucial para:

- Acessibilidade para deficientes visuais
- SEO e indexação por buscadores
- Fallback quando a imagem não carrega



```

```



src

Caminho do arquivo de imagem (local ou URL)



alt

Descrição textual para acessibilidade e SEO

ELEMENTO <p> (PARÁGRAFO): CONTEÚDO E FLUXO

O Bloco Básico de Texto do <body>

Definição e Função Semântica do <p>

Definição do Parágrafo

O elemento <p> representa um parágrafo. Parágrafos são blocos de texto contendo uma ou mais frases relacionadas que abordam um único ponto ou ideia.

Modelo de Conteúdo

É classificado como **Flow Content**, o que significa que pode conter texto e elementos de nível de frase.

Comportamento Padrão

O <p> é um **elemento de bloco** (block element), o que significa que os navegadores tipicamente inserem um espaço vertical em branco (margem) antes e depois dele.

Sintaxe e Requisito

O elemento <p> exige **tags de abertura e fechamento** (<p> e </p>). Exemplo:

<p>

Os navegadores adicionam automaticamente uma linha em branco antes e depois de cada <p> elemento.

</p>

Diferenciação Crucial com
 (Quebra de Linha)

Elemento

O elemento
 é um **elemento vazio** (empty element ou void element) que força uma quebra de linha. Ele não exige tag de fechamento.

Uso Correto do

O
 deve ser usado apenas quando a quebra de linha é parte **intrínseca do conteúdo** (ex: em endereços ou poemas).

Alerta: Evite Uso Incorreto

Não se deve usar o
 para criar separação ou simular múltiplos parágrafos; isso é uma **prática incorreta** que afeta a semântica e acessibilidade.

Exemplo HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <title>Exemplo de Parágrafos e Quebras de Linha</title>
</head>
<body>
```

<p>Este é um parágrafo de texto normal. Ele contém várias frases que formam uma única ideia coesa. O navegador adicionará margens automaticamente antes e depois deste bloco.</p>

```
<p>
    Endereço da Empresa:<br>
    Rua da Inovação, 123<br>
    Centro, Cidade Digital - CD 00000-000<br>
    Planeta Terra
</p>
```

```
<!-- ERRO: Não use <br><br> para criar espaçamento entre parágrafos -->
<p>Este é o primeiro parágrafo.</p>
<br><br> <!-- USO INCORRETO PARA ESPAÇAMENTO -->
<p>Este é o segundo parágrafo. Use <p> e </p> para criar novos parágrafos, não múltiplos <br>s.</p>
```

```
</body>
</html>
```

ALERTA: HEAD vs. HEADINGS vs. HEADER

Três Elementos Essenciais com Funções Distintas

O Elemento `<head>` (Cabeçalho do Documento)

- Contém metadados invisíveis para navegador e mecanismos de busca.
- Localização: primeiro filho do `<html>`, antes do `<body>`.
- Visibilidade: **INVISÍVEL** na área de conteúdo da página.
- Conteúdo essencial: `<title>`, `<meta charset="UTF-8">`, `<link>/<style>`.

Headings (`<h1>` a `<h6>`) (Títulos de Conteúdo)

- Elementos de bloco que definem hierarquia e esboço do documento.
- Localização: dentro do `<body>`.
- Função: definir semântica e importância do conteúdo.
- **Alerta:** **NÃO** escolha heading baseado no tamanho da fonte (isso é CSS).

O Elemento `<header>` (Região Introdutória)

- Elemento semântico HTML5 para seção introdução ou navegação.
- Localização: dentro do `<body>`.
- Conteúdo típico: títulos (`h1-h6`), navegação (`<nav>`), logotipo.
- **Alerta:** Pode haver vários `<header>` por página.

ELEMENTO <head>: A SEÇÃO INVISÍVEL

Metadados e Informações para o Navegador

1. Definição

O elemento `<head>` denota a seção de cabeçalho do documento HTML. Ele contém **metadados** (informações sobre a página, que não são o conteúdo visível).

2. Função

O conteúdo do `<head>` é destinado principalmente aos navegadores, mecanismos de busca (SEO) e sistemas, e é **INVISÍVEL** na área de conteúdo do `<body>`.

3. Localização

Deve ser o **primeiro filho** do elemento `<html>`, vindo **antes** do `<body>`.

4. Conteúdo Essencial

O `<head>` deve conter um elemento `<title>`, que é a única parte visível para o usuário (exibida na aba do navegador).

Elementos Típicos dentro do `<head>`

<code><title></code>	Define o título do documento (visível na aba do navegador).
<code><meta></code>	Fornece informações sobre o documento (ex: codificação de caracteres <code>charset="utf-8"</code> , descrição da página, viewport para responsividade).
<code><link></code>	Define a relação com recursos externos, como folhas de estilo CSS, ícones de favoritos (favicon) ou fontes.
<code><script></code>	Usado para incorporar ou referenciar código JavaScript (geralmente com o atributo <code>defer</code> para otimização de carregamento).

Exemplo de Estrutura `<head>` Completa

```
<!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 Minha Página - SEO Amigável</title>
  <meta name="description" content="Uma breve descrição do conteúdo da página para mecanismos de busca.">
  <link rel="icon" href="favicon.ico" type="image/x-icon">
  <link rel="stylesheet" href="estilo.css">
  <script src="script.js" defer></script>
</head>
<body>
  <!-- Conteúdo visível da página -->
</body>
</html>
```

HEADINGS (<h1> a <h6>): HIERARQUIA E ESTRUTURA

Os Títulos do Conteúdo Visível (Heading Content)

1. Definição

Elementos de bloco lógico que implementam seis níveis de títulos do documento. O <h1> é o mais proeminente.

2. Localização

Estão sempre localizados dentro do <body>, onde o conteúdo visível da página reside.

3. Função

Usados para estabelecer a hierarquia de informações e criar o esboço do documento (Document Outline). Essencial para SEO e acessibilidade, auxiliando na navegação e compreensão do conteúdo.

4. Regra Semântica

A escolha do nível (ex: <h2> ou <h4>) deve ser baseada na **importância semântica** do conteúdo, e **NÃO** no tamanho da fonte. O CSS é que controla a aparência visual.

5. Alerta de Diferenciação

O <h1> (e outros headings) são conteúdo visível e hierárquico. O <head>, por outro lado, contém metadados invisíveis.

Exemplo de Estrutura de Títulos Hierárquica

Para uma boa organização e semântica, siga esta hierarquia:

<**h1**>: Título principal da página ou do conteúdo mais importante (deve haver apenas um por página).

<**h2**>: Seções principais do conteúdo.

<**h3**>: Subseções dentro das seções principais.

<**h4**>: Sub-subseções ou tópicos específicos dentro das subseções.

<**h5**> e <**h6**>: Níveis ainda mais específicos e detalhados, usados com moderação.

Exemplo Prático de Hierarquia em HTML

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>Blog de Tecnologia - Introdução ao HTML</title>
</head>
<body>
  <h1>Blog de Tecnologia</h1>

  <h2>Post: Introdução ao HTML</h2>
  <p>Uma breve introdução sobre o que é HTML e sua importância na web.</p>

  <h3>Seção: Estrutura básica do documento HTML</h3>
  <p>Explorando os elementos essenciais de um documento HTML.</p>

  <h4>Subseção: A função do <h1> na estrutura</h4>
  <p>Detalhes sobre a singularidade e importância do <h1>.</p>

  <h5>Subseção: Outros elementos de cabeçalho</h5>
  <p>Como usar <h2> a <h6> de forma eficaz.</p>
</body>
</html>
```

ELEMENTO <header>: REGIÃO INTRODUTÓRIA

Cabeçalho Semântico de Seção ou Documento (HTML5)

1. Definição

O elemento `<header>` é um elemento novo do HTML5. Representa o cabeçalho de uma seção ou de todo o documento.

2. Função

É usado para marcar conteúdo introdutório ou de navegação. O `<header>` normalmente inclui o logotipo do site, navegação principal ou títulos.

3. Localização

É um elemento de fluxo (Flow Content) e é **VISÍVEL**, localizado sempre dentro do `<body>`.

4. Conteúdo Típico

Um `<header>` tipicamente contém um ou mais Headings (h1 a h6) e pode conter o elemento `<nav>`.

5. Alerta de Diferenciação

Você pode ter vários elementos `<header>` em uma página (um para o `<body>`, um para um `<article>`, um para uma `<section>`), mas apenas um elemento `<head>`.

Exemplo Prático de Uso do <header> em HTML5

Este exemplo demonstra a aplicação de múltiplos elementos `<header>` para estruturar semanticamente uma página.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>Exemplo de Múltiplos Headers</title>
</head>
<body>
  <!-- Header principal do documento -->
  <header>
    <h1>Site de Notícias Tech</h1>
    <h2>Fique por dentro das últimas novidades!</h2>
    <nav>
      <ul>
        <li><a href="#">Home</a></li>
        <li><a href="#">Artigos</a></li>
        <li><a href="#">Sobre</a></li>
        <li><a href="#">Contato</a></li>
      </ul>
    </nav>
  </header>

  <main>
    <section>
      <!-- Header de uma seção específica -->
      <header>
        <h2>Artigos Mais Recentes</h2>
      </header>
      <article>
        <!-- Header de um artigo individual -->
        <header>
          <h3>O Impacto da Inteligência Artificial</h3>
          <p>Publicado em: 15 de Outubro de 2023 por <a href="#">Ana Silva</a></p>
        </header>
        <p>A inteligência artificial está revolucionando diversas indústrias...</p>
      </article>
      <article>
        <!-- Outro header de artigo -->
        <header>
          <h3>Novidades em Desenvolvimento Web</h3>
          <p>Publicado em: 10 de Outubro de 2023 por <a href="#">Bruno Costa</a></p>
        </header>
        <p>Frameworks modernos e as melhores práticas para desenvolvimento front-end...</p>
      </article>
    </section>
  </main>

  <footer>
    <p>&copy; 2023 Notícias Tech. Todos os direitos reservados.</p>
  </footer>

</body>
</html>
```

Atributos Globais: class, id e title

Atributos globais são aplicáveis a quase todos os elementos HTML. Eles contêm informações sobre o elemento e são vitais para CSS e JavaScript manipularem o conteúdo.



class

Usado para **agrupar** múltiplos elementos que compartilham o mesmo estilo. Um elemento pode ter várias classes. Essencial para aplicar CSS de forma eficiente.

```
<p class="destaque alerta">
```



id

Identificador **exclusivo** por elemento na página. Use para JavaScript ou links internos que apontam para seções específicas. Deve ser único no documento.

```
<section id="contato">
```



title

Adiciona uma dica de ferramenta (tooltip) quando o usuário passa o mouse. É lido por leitores de tela, melhorando a acessibilidade e fornecendo contexto adicional.

```
<abbr title="HyperText">
```

EXEMPLO PRÁTICO: ATRIBUTOS GLOBAIS EM AÇÃO

Demonstração de class, id e title no HTML

Neste exemplo prático, vemos como os atributos globais `class`, `id` e `title` trabalham juntos para estruturar o conteúdo, aplicar estilos e melhorar a acessibilidade de uma página HTML.

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Atributos Globais: class, id e title</title>
<style>
body {
    background-color: #121212;
    color: #e0e0e0;
    font-family: "Segoe UI", Arial, sans-serif;
    padding: 40px;
    line-height: 1.6;
}
h1 {
    color: #00ff88; /* Verde Neon */
    text-transform: uppercase;
}
p {
    margin-bottom: 1rem;
}
.destaque {
    color: #00ff88;
    font-weight: bold;
}
.importante {
    background-color: #1e1e1e;
    border-left: 4px solid #00ff88;
    padding: 10px;
    margin-top: 10px;
}
#introducao {
    border-bottom: 2px solid #8affc1;
    padding-bottom: 8px;
}
</style>
</head>
<body>

<h1>Atributos Globais: <code>class</code>, <code>id</code> e <code>title</code></h1>

<section id="introducao" title="Seção de introdução ao tema">
    <p>Os <strong>atributos globais</strong> são aqueles que podem ser usados em <em>qualquer elemento HTML</em>.
        Eles ajudam na identificação, organização e acessibilidade do conteúdo.</p>
</section>

<p class="destaque" title="Parágrafo de destaque">Este parágrafo usa o atributo <code>class="destaque"</code> para aplicar um estilo especial e o atributo <code>title</code> para mostrar uma dica ao passar o mouse.</p>

<div class="importante" title="Bloco informativo">
    <p>O atributo <code>class</code> é ideal para aplicar o mesmo estilo a vários elementos.</p>
    <p>Já o atributo <code>id</code> deve ser único e é útil para identificar um elemento específico no documento, como o cabeçalho, rodapé ou seções de uma página.</p>
</div>

</body>
</html>
```

PRÁTICA ORIENTADA:

Este exercício prático desafia você a aplicar os conhecimentos de HTML semântico na criação de um artigo ou perfil sobre um tópico de sua paixão. O objetivo é construir uma estrutura lógica e hierárquica robusta, utilizando os elementos aprendidos para um documento bem organizado e acessível.



01

Atividade Prática: Estrutura Semântica de um Artigo de Blog

Tema: Criação da estrutura básica de um artigo de blog ou perfil sobre um tópico de sua escolha (banda, jogo, série, hobby, personalidade etc.).

Objetivo:

Aplicar todos os elementos **semânticos e de bloco** aprendidos em aula, garantindo uma estrutura lógica e hierárquica adequada para o documento – preparando o código para uso em ambientes profissionais.

Tempo Estimado: 25 a 40 minutos (atividade de sala).

🔧 Instruções Passo a Passo

1. Criação da Base do Arquivo

- No editor de texto ou IDE, inicie o documento com a declaração `<!DOCTYPE html>` e a estrutura básica: `<html>`, `<head>` e `<body>`.
- Dentro do `<head>`, inclua `<meta charset="UTF-8">` e um `<title>` descritivo (exemplo: *O Horizonte Digital: Tecnologias que Moldam o Amanhã*).

2. Definição da Região Principal

- No `<body>`, use o elemento `<article>` para envolver todo o conteúdo principal, representando uma composição independente (como um post de blog).
- Dê ao `<article>` um **id exclusivo**, por exemplo `id="artigo-principal"` (o id deve ser único em toda a página).

3. Estabelecimento da Hierarquia

- Dentro do `<article>`, utilize `<h1>` para o título principal do artigo (exemplo: *Os 5 Melhores Recursos de [Nome do Tema]*).
- Crie uma região introdutória com o elemento `<header>`, contendo um `<h2>` para o subtítulo (exemplo: *Por que escolhi este tópico?*).
- **Importante:** use os headings (`<h1>` a `<h6>`) conforme a hierarquia lógica – o tamanho da fonte é controlado pelo CSS, não pela tag.

4. Criação de Blocos de Texto e Listas

- Use `<p>` para escrever os parágrafos introdutórios ou descritivos.
- Crie uma seção temática com o elemento `<section>`.
- Dentro da seção, insira uma **lista não ordenada** (``) contendo os "Top 5" ou principais características, com cada item em ``.
- Atribua uma **classe genérica**, como `class="lista-destaque"`, à lista para permitir estilização futura.

5. Adição de Links e Imagens

- Dentro de um `<p>` ou ``, insira um link (`<a>`) para uma fonte externa, utilizando os atributos `href` e `title`.
- Adicione uma imagem (``) relacionada ao tema, com os atributos `src` e `alt`.
→ O atributo `alt` deve conter uma descrição textual que substitui a imagem em caso de erro de carregamento.

📦 Requisitos de Entrega

1. **Nome do Arquivo:** use apenas letras minúsculas e hifens (exemplo: *meu-primeiro-artigo.html*).
2. **Estrutura Completa:** `<head>` com metadados e `<body>` com conteúdo visível.
3. **Atributos Globais:** uso de `id`, `class` e `title` obrigatórios.
4. **Semântica:** presença de `<h1>`, `<h2>`, `<p>`, `` e ``.

Próximos Passos: Práticas Profissionais

Agora você domina os fundamentos do HTML. Estas práticas profissionais começam a te preparar para o mercado de trabalho.

Separação de Preocupações

HTML para a estrutura e significado, CSS para a apresentação visual, JavaScript para comportamento. Nunca misture responsabilidades.

Consistência de Nomenclatura

Use nomes de arquivos e pastas em **letras minúsculas**, separando palavras com hifens: `sobre-nos.html`, `estilo-principal.css`

Sintaxe Limpa

Mantenha código legível com indentação correta. Use minúsculas para elementos e atributos. Comente seções complexas para facilitar manutenção.

Projeto Prático (FER-PO21)

Você construirá a estrutura de uma página de conteúdo simples (artigo ou perfil) utilizando apenas HTML semântico, aplicando todos os elementos aprendidos hoje.

Próxima Aula (Aula 2)

Formulários e Controles de Entrada: Inputs, Selects, Textareas e Validação – elementos essenciais para interação do usuário.