Tags HTML

**-** Novos elementos do HTML 5.0

**1.<section>**

**Conceito**: Representa uma seção genérica de conteúdo dentro de um documento. É usado para agrupar conteúdos relacionados em uma página. Geralmente, cada seção pode ter seu próprio título.

**Funcionalidade**: Ajuda a estruturar uma página de forma lógica, dividindo o conteúdo em blocos distintos.

**Exemplo**:

<section>

<h2>Seção de Notícias</h2>

<p>Aqui você encontra as últimas notícias.</p>

</section>

**2.<article>**

**Conceito**: Representa uma unidade independente e autocontida de conteúdo, como uma postagem de blog, um artigo de revista ou uma notícia.

**Funcionalidade**: Facilita o uso de conteúdos que podem ser distribuídos ou reutilizados independentemente.

**Exemplo**:

<article>

<h2>Título do Artigo</h2>

<p>Este é o conteúdo do artigo.</p>

</article>

**3.<nav>**

**Conceito**: Representa um bloco de links de navegação, como um menu principal ou uma lista de links para diferentes partes do site.

**Funcionalidade**: Indica ao navegador que o conteúdo deste elemento está relacionado à navegação da página ou do site.

**Exemplo**:

<aside>

<h3>Anúncios</h3>

<p>Confira as ofertas do dia!</p>

</aside>

**5.<footer>**

**Conceito**: Representa o rodapé de uma seção ou de uma página inteira. Geralmente contém informações de direitos autorais, links para políticas de privacidade, ou autores.

**Funcionalidade**: Indica o final de uma seção ou documento.

**Exemplo**:

<footer>

<p>&copy; 2024 Minha Empresa. Todos os direitos reservados.</p>

</footer>

**7.<time>**

**Conceito**: Representa um dado ou hora específica em um formato legível para máquinas e seres humanos.

**Funcionalidade**: Indica ao navegador que o conteúdo se refere a uma data ou hora, útil para eventos, publicações e prazos.

**Exemplo**:

<p>Publicado em: <time datetime="2024-10-07">7 de outubro de 2024</time></p>

**8.<mark>**

**Conceito**: Destaca o texto, geralmente para indicar uma parte importante ou relevante do conteúdo, como o resultado de uma busca.

**Funcionalidade**: Permite dar ênfase visual a trechos de texto importantes.

**Exemplo**:

<p>Pesquisando por: <mark>HTML5</mark> você encontrará ótimos recursos.</p>

**-** Elementos iterativos, agrupadores e multimídia do HTML 5

**1. <summary>**

O elemento <summary> é usado para fornecer um resumo ou título para o conteúdo do elemento <details>. O <details> é uma caixa expansível que, quando clicada, exibe ou oculta o conteúdo. O <summary> aparece como um botão que o usuário pode clicar para expandir ou contrair o conteúdo.

**Exemplo:**

<details>

<summary>Mais informações</summary>

<p>Este é o conteúdo que pode ser expandido ou recolhido.</p>

</details>

**2. <meter>**

O <meter> representa uma medição escalar dentro de um intervalo conhecido, ou seja, uma barra que representa visualmente um valor numérico. Ele é geralmente usado para exibir valores em contextos como notas, porcentagens de conclusão ou níveis de recurso.

**Exemplo:**

<p>Nível de bateria: <meter min="0" max="100" value="75">75%</meter></p>

**3. <progress>**

O elemento <progress> exibe uma barra de progresso que mostra o andamento de uma tarefa em curso, como um upload ou download. Ele é muito usado para monitorar o progresso de processos que podem demorar algum tempo.

**Exemplo:**

<p>Progresso de upload: <progress value="40" max="100">40%</progress></p>

**4. <figure>**

O <figure> é um elemento de agrupamento que associa conteúdos como imagens, gráficos, tabelas ou diagramas, junto com sua legenda. Ele é independente do fluxo principal do documento e pode ser movido ou estilizado separadamente.

**Exemplo:**

<figure>

<img src="imagem-exemplo.jpg" alt="Exemplo de imagem">

<figcaption>Figura 1: Um exemplo de imagem com legenda.</figcaption>

</figure>

**5. <figcaption>**

O <figcaption> é utilizado para fornecer uma legenda ao conteúdo dentro do <figure>. Ele descreve ou complementa o conteúdo do <figure>, ajudando a contextualizá-lo.

**Exemplo:**

<figure>

<img src="grafico.jpg" alt="Gráfico de vendas">

<figcaption>Gráfico de Vendas Anual</figcaption>

</figure>

**6. <audio>**

O elemento <audio> é usado para incorporar conteúdo de áudio em uma página. Ele permite a inclusão de arquivos de áudio como músicas, podcasts ou outros sons. Pode incluir controles como play, pause e volume.

**Exemplo:**

<audio controls>

<source src="musica.mp3" type="audio/mpeg">

Seu navegador não suporta o elemento de áudio.

</audio>

**7. <video>**

O <video> é utilizado para incorporar vídeos diretamente na página da web. Ele pode incluir fontes em diferentes formatos e opções de controles, como play, pause, volume, e até mesmo legendas.

**Exemplo:**

<video controls>

<source src="video.mp4" type="video/mp4">

Seu navegador não suporta o elemento de vídeo.

</video>

**-**  Novos elementos de Formulários HTML 5

**1. <input type="tel">**

Este campo de entrada é usado para capturar números de telefone. Não há validação automática de formato, mas navegadores móveis podem apresentar um teclado numérico específico.

**Exemplo:**

<label for="telefone">Telefone:</label>

<input type="tel" id="telefone" name="telefone" placeholder="(xx) xxxx-xxxx">

**2. <input type="number">**

Este campo é utilizado para capturar números. Ele permite definir limites mínimo e máximo, bem como passos incrementais para aumentar ou diminuir o valor.

**Exemplo:**

<label for="quantidade">Quantidade:</label>

<input type="number" id="quantidade" name="quantidade" min="1" max="10" step="1">

**3. <input type="range"> (Slider)**

Este campo permite que o usuário selecione um valor em um intervalo usando um controle deslizante (slider). Você pode definir valores mínimo, máximo e de passo.

**Exemplo:**

<label for="volume">Volume:</label>

<input type="range" id="volume" name="volume" min="0" max="100" step="5">

**4. <input type="time">**

Este campo permite que o usuário selecione uma hora do dia. A entrada é controlada pelo navegador, e o formato da hora é consistente com o padrão internacional de tempo.

**Exemplo:**

<label for="hora">Selecione a hora:</label>

<input type="time" id="hora" name="hora">

**5. <input type="week">**

Este campo permite a seleção de uma semana e de um ano. O formato é uma combinação do número da semana e do ano correspondente.

**Exemplo:**

<label for="semana">Semana:</label>

<input type="week" id="semana" name="semana">

**6. <input type="datetime-local">**

Este campo permite a seleção de uma data e hora (sem fuso horário). Ele oferece um controle para selecionar tanto a data quanto o horário.

**Exemplo:**

<label for="agendamento">Agendar para:</label>

<input type="datetime-local" id="agendamento" name="agendamento">

**7. <input type="month">**

Este campo permite a seleção de um mês e um ano. Ele exibe um controle para escolher um mês em um determinado ano.

**Exemplo:**

<label for="mes">Selecione o mês:</label>

<input type="month" id="mes" name="mes">

**8. <input type="datetime"> (Descontinuado)**

Este campo era usado para seleção de data e hora com fuso horário, mas foi descontinuado e substituído por <input type="datetime-local">.

**9. <input type="date">**

Este campo permite que o usuário selecione uma data. O formato da data é tratado automaticamente pelo navegador.

**Exemplo:**

<label for="data">Data de nascimento:</label>

<input type="date" id="data" name="data">

**10. <input type="search">**

O campo de busca é similar ao campo de texto, mas otimizado para entradas de pesquisa. Ele pode apresentar uma opção de limpar o campo, dependendo do navegador.

**Exemplo:**

<label for="busca">Pesquisar:</label>

<input type="search" id="busca" name="busca" placeholder="Pesquisar...">

**11. <input type="email">**

Este campo captura endereços de e-mail e valida automaticamente o formato inserido. Se o e-mail estiver no formato incorreto, o navegador pode alertar o usuário.

**Exemplo:**

<label for="email">E-mail:</label>

<input type="email" id="email" name="email" placeholder="exemplo@dominio.com">

**12. <input type="url">**

Este campo captura URLs (endereços de websites). O navegador pode validar automaticamente o formato da URL fornecida.

**Exemplo:**

<label for="site">Website:</label>

<input type="url" id="site" name="site" placeholder="https://www.exemplo.com">

**13. <input type="color">**

Este campo permite que o usuário selecione uma cor de uma paleta de cores gerada pelo navegador.

**Exemplo:**

<label for="cor">Escolha uma cor:</label>

<input type="color" id="cor" name="cor" value="#ff0000">

**14. <datalist>**

Este elemento fornece uma lista de opções pré-definidas que pode ser associada a um campo de entrada. Ele permite que o usuário selecione uma opção da lista ou digite um valor próprio.

**Exemplo:**

<label for="navegador">Escolha seu navegador:</label>

<input list="navegadores" id="navegador" name="navegador">

<datalist id="navegadores">

<option value="Google Chrome">

<option value="Firefox">

<option value="Safari">

<option value="Edge">

</datalist>

**15. <output>**

O elemento <output> é usado para exibir o resultado de um cálculo ou operação. Ele pode ser atualizado dinamicamente com valores calculados através de JavaScript.

**Exemplo:**

<label for="numero1">Número 1:</label>

<input type="number" id="numero1" name="numero1">

<label for="numero2">Número 2:</label>

<input type="number" id="numero2" name="numero2">

<button onclick="calcular()">Calcular</button>

<output id="resultado"></output>

<script>

function calcular() {

var num1 = document.getElementById('numero1').value;

var num2 = document.getElementById('numero2').value;

var resultado = parseFloat(num1) + parseFloat(num2);

document.getElementById('resultado').value = resultado;

}

</script>

**-**  Novos atributos, validadores e expressões regulares em HTML 5

**1. autofocus** O atributo autofocus define que um campo de entrada será automaticamente focado assim que a página for carregada. É útil para melhorar a experiência do usuário em formulários ao tornar o campo mais relevante o primeiro a receber o foco.

**Exemplo:**

<label for="nome">Nome:</label>

<input type="text" id="nome" name="nome" autofocus>

**2. autocomplete** O atributo autocomplete sugere valores previamente inseridos pelo usuário em campos de formulários, como nomes, endereços ou e-mails. Ele pode ser configurado para "on" (ativado) ou "off" (desativado).

**Exemplo:**

<label for="email">E-mail:</label>

<input type="email" id="email" name="email" autocomplete="on">

**3. placeholder** O atributo placeholder fornece um texto de dica dentro de um campo de entrada antes que o usuário digite algo. Ele é substituído assim que o usuário começa a digitar.

**Exemplo:**

<label for="usuario">Usuário:</label>

<input type="text" id="usuario" name="usuario" placeholder="Digite seu nome de usuário">

**4.** **maxlength** O atributo maxlength define o número máximo de caracteres que podem ser inseridos em um campo de texto.

**Exemplo:**

<label for="senha">Senha:</label>

<input type="password" id="senha" name="senha" maxlength="12">

**5. pattern** O atributo pattern define uma **expressão regular** (regex) que valida a entrada do usuário. Se o valor inserido não corresponder ao padrão definido, o formulário será considerado inválido.

**Exemplo:** Aceitar somente números de telefone no formato (xx) xxxx-xxxx:

<label for="telefone">Telefone:</label>

<input type="tel" id="telefone" name="telefone" pattern="\(\d{2}\) \d{4}-\d{4}" placeholder="(xx) xxxx-xxxx">

**6. required** O atributo required indica que o campo é obrigatório. O formulário não pode ser enviado se o campo estiver vazio.

**Exemplo:**

<label for="email">E-mail:</label>

<input type="email" id="email" name="email" required>

**7. novalidate** O atributo novalidate desativa a validação automática do formulário HTML5, o que significa que o formulário pode ser enviado mesmo que os campos sejam inválidos. Isso é útil quando você deseja lidar com a validação manualmente, geralmente via JavaScript.

**Exemplo:**

<form action="/enviar" method="post" novalidate>

<label for="nome">Nome:</label>

<input type="text" id="nome" name="nome">

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

</form>

**8. formnovalidate** O atributo formnovalidate pode ser aplicado especificamente a botões de envio (<button>) para que, ao clicar nesse botão, o formulário seja enviado sem validação. Ele anula a validação automática para esse envio específico.

**Exemplo:**

<form action="/enviar" method="post">

<label for="nome">Nome:</label>

<input type="text" id="nome" name="nome" required>

<button type="submit" formnovalidate>Enviar sem validar</button>

</form>

**9.** **Expressões Regulares (Regex)**

**O que é uma Expressão Regular?**

Uma **expressão regular** é uma sequência de caracteres que define um padrão de pesquisa. Em HTML5, as expressões regulares são usadas com o atributo pattern para validar entradas de dados em campos de formulários. Elas permitem especificar padrões complexos de validação, como e-mails, números de telefone, formatos de data, etc.

**Para que Serve uma Expressão Regular?**

Expressões regulares são úteis para garantir que a entrada do usuário siga um formato específico, como:

* E-mails que devem ter um "@" seguido de um domínio.
* Senhas que devem conter letras, números e caracteres especiais.
* Números de telefone que devem seguir um formato predefinido.

**Como Definir uma Expressão Regular?**

* **Símbolos básicos de expressões regulares:**
  + ^ - Início da string.
  + $ - Fim da string.
  + \d - Qualquer dígito (0-9).
  + \w - Qualquer letra ou número.
  + {n} - Exatamente n repetições.
  + + - Um ou mais do símbolo anterior.
  + [a-z] - Qualquer letra de 'a' a 'z'.

**Exemplo de Expressão Regular**

<label for="email">E-mail:</label>

<input type="email" id="email" name="email" pattern="[a-z0-9.\_%+-]+@[a-z0-9.-]+\.[a-z]{2,4}$" required>

**Exemplo 1: Validação de e-mail**

Aqui está um exemplo de como validar um e-mail com uma expressão regular usando o atributo pattern.

<label for="email">E-mail:</label>

<input type="email" id="email" name="email" pattern="[a-z0-9.\_%+-]+@[a-z0-9.-]+\.[a-z]{2,4}$" required>

Explicação:

* [a-z0-9.\_%+-]+ - Aceita letras minúsculas, números, pontos, sublinhados e outros caracteres permitidos antes do "@".
* @[a-z0-9.-]+ - Exige um "@" seguido de letras e números para o domínio.
* \.[a-z]{2,4}$ - Um ponto seguido por de 2 a 4 letras para o sufixo do domínio (como .com ou .net).

**Exemplo 2: Validação de senha**

Vamos criar uma expressão regular para validar uma senha que contenha pelo menos 8 caracteres, incluindo pelo menos uma letra maiúscula, uma minúscula, um número e um caractere especial.

html

<label for="senha">Senha:</label>

<input type="password" id="senha" name="senha" pattern="(?=.\*\d)(?=.\*[a-z])(?=.\*[A-Z])(?=.\*[@#$%^&+=]).{8,}" required>

Explicação:

* (?=.\*\d) - Exige pelo menos um dígito.
* (?=.\*[a-z]) - Exige pelo menos uma letra minúscula.
* (?=.\*[A-Z]) - Exige pelo menos uma letra maiúscula.
* (?=.\*[@#$%^&+=]) - Exige pelo menos um caractere especial.
* .{8,} - Exige pelo menos 8 caracteres.

Com o uso de atributos e expressões regulares, o HTML5 oferece uma validação de formulários mais robusta e interativa, garantindo que os dados inseridos pelos usuários estejam no formato correto.

4o