

# Desenvolvimento Web 1

Heder Dorneles Soares

IFSP

16 de agosto de 2023

HTML é a sigla para *HyperText Markup Language*, que, em português significa linguagem para marcação de hipertexto.

# O que é HTML?

HTML (*HyperText Markup Language*) é a linguagem de marcação padrão para criar páginas da web. É a base fundamental para construir qualquer site e permite que você defina a estrutura e o conteúdo do documento, como texto, imagens, links, formulários, etc.

As páginas HTML são construídas usando tags, que são usadas para marcar elementos na página. Cada tag HTML começa com um sinal de menor que ("<") e termina com um sinal de maior que (">"). Um elemento é formado por uma tag de abertura e uma tag de fechamento, envolvendo o conteúdo que deseja-se marcar.

Exemplo de um parágrafo: `<p>Este é um parágrafo.</p>`

# Estrutura básica do HTML

Uma página HTML básica geralmente possui a seguinte estrutura:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Título da Página</title>
  </head>
  <body>
    <!-- Conteúdo da página aqui -->
  </body>
</html>
```

O elemento `<html>` é o contêiner principal, contendo `<head>` e `<body>`, onde o conteúdo visível é colocado.

# Cabeçalhos e Parágrafos

Cabeçalhos (<h1> a <h6>) são usados para títulos e subtítulos na página.

Exemplo:

```
<h1>Título Principal</h1>
```

```
<h2>Subtítulo</h2>
```

```
<h3>Outro Subtítulo</h3>
```

```
<!-- ... -->
```

```
<h6>Subtítulo Menor</h6>
```

Parágrafos são marcados com a tag <p>:

```
<p>Este é um parágrafo.</p>
```

# O que é um Elemento HTML?

Um elemento HTML é definido por uma tag de abertura, algum conteúdo e uma tag de fechamento:

```
<tagname> Conteúdo vai aqui... </tagname>
```

O elemento HTML é tudo desde a tag de abertura até a tag de fechamento:

```
<h1>Meu Primeiro Título</h1>
```

```
<p>Meu primeiro parágrafo.</p>
```

# A Declaração <!DOCTYPE>

A declaração <!DOCTYPE> representa o tipo de documento e ajuda os navegadores a exibirem corretamente as páginas da web.

Ela deve aparecer apenas uma vez, no topo da página (antes de qualquer tag HTML).

A declaração <!DOCTYPE> não diferencia letras maiúsculas de minúsculas.

A declaração <!DOCTYPE> para HTML5 é:

```
<!DOCTYPE html>
```



Links HTML são definidos com a tag <a>:

## Exemplo

```
<a href="https://www.google.com">Este é um link</a>
```

O destino do link é especificado no atributo href.

Atributos são usados para fornecer informações adicionais sobre elementos HTML.

# Imagens HTML

Imagens HTML são definidas com a tag `<img>`.

O arquivo de origem (`src`), texto alternativo (`alt`), largura (`width`) e altura (`height`) são fornecidos como atributos:

## HTML

```

```

Podemos usar valores em porcentagem nos atributos `width` e `height`. `Alt` é usado para auxiliar serviços de busca e usuário que utilizam leitores de tela.

# Atributos da Tag <body>

A tag <body> é usada para definir o corpo de um documento HTML e contém os seguintes atributos:

- background: Especifica a imagem de plano de fundo da página.
- bgcolor: Define a cor de fundo da página.
- text: Define a cor do texto.
- link: Define a cor dos links não visitados.

## Exemplo

```
<body bgcolor="#f0f0f0" text="#000000" link="#0000ff">
```

# Padrão de Cores Hexadecimal no HTML

Em HTML, as cores são representadas usando o padrão de cores hexadecimal. O formato hexadecimal consiste em seis dígitos alfanuméricos que variam de 0 a 9 e de A a F.

Cada par de dígitos representa a intensidade das cores vermelho (R), verde (G) e azul (B) respectivamente. Os valores vão de 00 (sem intensidade) até FF (máxima intensidade).

Por exemplo:

- #FF0000 representa vermelho puro.
- #00FF00 representa verde puro.
- #0000FF representa azul puro.
- #FFFFFF representa branco.
- #000000 representa preto.

# O Atributo style do HTML

Definir o estilo de um elemento HTML pode ser feito com o atributo `style`. O atributo `style` do HTML possui a seguinte sintaxe:

## Exemplo

```
<tagname style="propriedade:valor;">
```

A propriedade e o valor são atributos do CSS<sup>1</sup>.

---

<sup>1</sup>Nós veremos sobre CSS posteriormente nesta disciplina.

# Exemplo de background-color

Você pode definir a cor de fundo de um elemento HTML usando o atributo `style` e a propriedade `background-color`.

## Exemplo:

```
<p style="background-color: #ff0000; color: #ffffff;">  
    Este é um parágrafo com fundo vermelho e texto branco.  
</p>
```

Neste exemplo, a cor de fundo é definida como vermelho (`#ff0000`) e a cor do texto é definida como branco (`#ffffff`).

A propriedade CSS `color` define a cor do texto para um elemento HTML:

## Exemplo

```
<h1 style="color:blue;">Este é um título</h1>  
<p style="color:red;">Este é um parágrafo.</p>
```

Neste exemplo, a cor do texto para o título é definida como azul (`blue`) e a cor do texto para o parágrafo é definida como vermelho (`red`).

# Fonte, Tamanho e Alinhamento

No CSS, você pode controlar a fonte, o tamanho da fonte e o alinhamento do texto para um elemento HTML:

- `font-family`: Define o tipo de fonte a ser usada (por exemplo: Arial, Verdana, Courier).
- `font-size`: Define o tamanho da fonte (por exemplo: 12px, 16px).
- `text-align`: Define o alinhamento do texto (por exemplo: left, right, center).

## Exemplo

```
<p style="font-family: Arial; font-size: 16px;  
text-align: center;">
```

Parágrafo com fonte Arial, tamanho 16px e centralizado.

```
</p>
```

Neste exemplo, o parágrafo tem fonte Arial, tamanho de fonte de 16 pixels e está centralizado.



## Exercício

Fazer uma página mini-curriculo.

Contendo: foto, formatação de fonte, e uso de links.



---

## Nome Sobrenome

Idade: 25 anos

E-mail: email@example.com

Telefone: (00) 0000-0000

## Formação Acadêmica

Bacharel em Ciência da Computação - Universidade XYZ

## Experiência Profissional

Desenvolvedor Web - Empresa ABC (2018 - Presente)

Estagiário em TI - Empresa XYZ (2016 - 2018)

## Habilidades Técnicas

- HTML
- CSS
- JavaScript
- PHP
- MySQL

# Exemplo de Listagem em HTML

Suponha que queremos criar uma lista de itens usando HTML:

```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ul>
```

Isso será exibido no navegador como:

- Item 1
- Item 2
- Item 3

# Exemplo de Listagem Ordenada em HTML

Suponha que queremos criar uma lista ordenada de itens usando HTML:

```
<ol>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ol>
```

Isso será exibido no navegador como:

- 1 Item 1
- 2 Item 2
- 3 Item 3

# Listas de Descrição em HTML

HTML também suporta listas de descrição. Uma lista de descrição é uma lista de termos, com uma descrição para cada termo.

A tag `<dl>` define a lista de descrição, a tag `<dt>` define o termo (nome), e a tag `<dd>` descreve cada termo:

## Exemplo

```
<dl>
  <dt>Português</dt>
  <dd>- Aulas às sextas</dd>
  <dt>Matemática</dt>
  <dd>- Aulas às segundas</dd>
</dl>
```

# HTML Links - O Atributo target

Por padrão, a página vinculada será exibida na janela do navegador atual. Para alterar isso, você deve especificar outro destino para o link.

O atributo `target` especifica onde abrir o documento vinculado.

O atributo `target` pode ter um dos seguintes valores:

- `_self` - Padrão. Abre o documento na mesma janela ou guia em que foi clicado.

- `_blank` - Abre o documento em uma nova janela ou guia.

- `_parent` - Abre o documento no quadro pai.

- `_top` - Abre o documento em toda a janela do navegador.

# Âncoras em HTML

Em HTML, as âncoras são usadas para criar links internos em uma página. Elas permitem que os visitantes da página sejam redirecionados para uma parte específica do conteúdo sem sair da página atual.

Para criar uma âncora, você precisa seguir dois passos:

- 1 Adicionar a âncora ao local de destino usando o atributo `id`.
- 2 Criar o link que aponta para a âncora usando o atributo `href` com o formato `#nome-da-ancora`.

## Exemplo:

```
<h2 id="secao1">Seção 1</h2>
<p>Este é o conteúdo da Seção 1.</p>

<a href="#secao1">Ir para Seção 1</a>
```

No exemplo acima, o link "Ir para Seção 1" redirecionará o usuário para a seção com o `id` "secao1" na mesma página.

# Exercício: Criando uma Página com Âncoras

Vamos praticar o uso de âncoras em HTML criando uma página com seções e links de navegação.

- 1 Crie uma página HTML com pelo menos três seções usando os cabeçalhos `<h2>`, `<h3>` ou `<h4>`.
- 2 Adicione conteúdo em cada seção, como texto ou imagens.
- 3 Inclua links de navegação que redirecionem o usuário para cada seção da página usando âncoras.
- 4 Certifique-se de que os links de navegação funcionam corretamente e redirecionam o usuário para a seção correta quando clicados.

Dica: Use o atributo `id` para criar as âncoras nos locais de destino e o atributo `href` com o formato `#nome-da-ancora` para criar os links de navegação.

Pratique suas habilidades em HTML e divirta-se criando uma página interativa com âncoras!



# Passo 1: Criar a Âncora

Passo 1: Adicione a âncora ao local de destino usando o atributo id.

```
<h2 id="secao1">Seção 1</h2>  
<p>Conteúdo da Seção 1.</p>
```

## Passo 2: Criar o Link

Passo 2: Crie o link que aponta para a âncora usando o atributo href com o formato #nome-da-ancora.

```
<a href="#secao1">Ir para Seção 1</a>
```

Isso redirecionará o usuário para a Seção 1 na mesma página quando o link for clicado.

# Âncoras de uma Página para Outra

Em HTML, você pode criar âncoras que levam de uma página para outra. Isso é útil quando você deseja criar links que direcionam os usuários para uma seção específica de outra página.

Para criar uma âncora que leva de uma página para outra, siga estes passos:

- 1 Crie a âncora no local de destino usando o atributo `id`.
- 2 Crie o link na página de origem usando o atributo `href` com o formato `#nome-da-ancora` no arquivo HTML da página de destino.

Vamos ver um exemplo de como fazer isso.

# Exemplo: Criando uma Âncora de uma Página para Outra

Considere que temos duas páginas HTML: `pagina_origem.html` e `pagina_destino.html`.

Na `pagina_destino.html`, crie a âncora usando o atributo `id`:

```
<h2 id="secao1">Seção 1</h2>  
<p>Conteúdo da Seção 1.</p>
```

## Exemplo (cont.): Criando o Link na Página de Origem

Na `pagina_origem.html`, crie o link para a âncora usando o atributo `href` com o formato `#nome-da-ancora`:

```
<a href="pagina_destino.html#secao1">Ir para Seção 1</a>
```

Quando os usuários clicarem no link "Ir para Seção 1", eles serão redirecionados para a Seção 1 na página `pagina_destino.html`.

# Introdução a Tabelas em HTML

As tabelas são usadas em HTML para organizar e exibir informações em linhas e colunas. Elas são úteis para apresentar dados de forma tabular e estruturada.

Para criar uma tabela em HTML, você precisa usar as tags `<table>`, `<tr>`, `<th>` e `<td>`.

- `<table>` define a tabela.
- `<tr>` define uma linha na tabela.
- `<th>` define uma célula de cabeçalho (cabeçalho de coluna).
- `<td>` define uma célula de dados (conteúdo da tabela).

Vamos ver um exemplo simples de como criar uma tabela em HTML.

# Exemplo: Criando uma Tabela em HTML

Aqui está um exemplo básico de como criar uma tabela em HTML:

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

Nome	Idade
------	-------

João	25
------	----

Maria	30
-------	----

Isso criará uma tabela com duas colunas ("Nome" e "Idade") e duas linhas de dados.

# Bordas em Tabelas HTML (Sem CSS)

Você pode adicionar bordas a tabelas em HTML sem a necessidade de CSS. Isso é feito usando os atributos de borda diretamente nas tags da tabela.

Para adicionar bordas a uma tabela, você pode usar os seguintes atributos:

- `border` - Define a largura da borda da tabela.
- `bordercolor` - Define a cor da borda da tabela.
- `cellspacing` - Define o espaçamento entre células.
- `cellpadding` - Define o preenchimento interno das células.

Vamos ver um exemplo de como adicionar bordas a uma tabela em HTML.



# Exemplo: Adicionando Bordas a uma Tabela

Aqui está um exemplo de como adicionar bordas a uma tabela em HTML:

```
<table border="1" bordercolor="black" cellspacing="5"
cellpadding="2">
<tr>
  <th>Nome</th>
  <th>Idade</th>
</tr>
<tr>
  <td>João</td>
  <td>25</td>
</tr>
<tr>
  <td>Maria</td>
  <td>30</td>
</tr>
</table>
```

# Atributos colspan e rowspan em Tabelas HTML

Os atributos `colspan` e `rowspan` permitem que você estenda células em tabelas HTML através de colunas e linhas, respectivamente.

`colspan` combina células horizontalmente, e `rowspan` combina células verticalmente.

Vamos ver um exemplo de como usar esses atributos para criar tabelas mais complexas.

## Exemplo: Usando colspan e rowspan

Aqui está um exemplo de como usar colspan e rowspan em uma tabela em HTML:

```
<table border="1" cellpadding="5">
<tr>
  <th>Produto</th>
  <th colspan="2">Preço</th>
</tr>
<tr>
  <td rowspan="2">Maçã</td>
  <td>R$ 2,00</td>
  <td>R$ 1,80</td>
</tr>
<tr>
  <td colspan="2">Oferta especial!</td>
</tr>
</table>
```

Produto	Preço	
Maçã	R\$ 2,00	R\$ 1,80
	Oferta especial!	

# Exercício

Reproduza a seguinte tabela:

Tabela usando Colspan e Rowspan			
Cantina no IFSP			
Item	Quantidade	Preço/Unit.	Total
Coxinha	4	2.50	R\$ 10.00
Misto	1	4.50	R\$ 4.50
Coca Lata	3	5.50	R\$ 16.50
Sub Total			R\$ 32.00
Taxa		5%	R\$ 1.60
Total da Conta			R\$ 33.60

# O Elemento <form> em HTML

O elemento HTML <form> é usado para criar um formulário HTML para entrada de usuário:

```
<form>  
  .  
  elementos do formulário  
  .  
</form>
```

O elemento <form> é um contêiner para diferentes tipos de elementos de entrada, como: campos de texto, caixas de seleção, botões de opção, botões de envio, etc.

# O Elemento `<input>` em HTML

O elemento HTML `<input>` é o elemento de formulário mais utilizado. Um elemento `<input>` pode ser exibido de várias maneiras, dependendo do atributo `type`.

Aqui estão alguns exemplos:

- `<input type="text">` - Campo de Texto
- `<input type="checkbox">` - Caixa de Seleção
- `<input type="radio">` - Botão de Opção
- `<input type="submit">` - Botão de Envio
- `<input type="button">` - Botão Clicável

# Botões de Opção (Radio Buttons) em HTML

A tag `<input type="radio">` define um botão de opção.

Botões de opção permitem que o usuário selecione UM de um número limitado de escolhas.

Exemplo:

```
<form>
  <input type="radio" id="male" name="gender"
  value="male">
  <label for="male">Masculino</label><br>
  <input type="radio" id="female" name="gender"
  value="female">
  <label for="female">Feminino</label><br>
  <input type="radio" id="other" name="gender"
  value="other">
  <label for="other">Outro</label>
</form>
```

# Caixas de Seleção (Checkboxes) em HTML

A tag `<input type="checkbox">` define uma caixa de seleção. Caixas de seleção permitem que o usuário selecione ZERO ou MAIS opções de um número limitado de escolhas.

Exemplo:

```
<form>
  <input type="checkbox" id="apple" name="fruits"
  value="apple">
  <label for="apple">Maçã</label><br>
  <input type="checkbox" id="banana" name="fruits"
  value="banana">
  <label for="banana">Banana</label><br>
  <input type="checkbox" id="orange" name="fruits"
  value="orange">
  <label for="orange">Laranja</label>
</form>
```



# Botão de Envio (Submit Button) em HTML

A tag `<input type="submit">` define um botão de envio para enviar os dados do formulário para um manipulador de formulário.

O manipulador de formulário é normalmente um arquivo no servidor com um script para processar os dados de entrada.

O manipulador de formulário é especificado no atributo `action` do formulário.

Exemplo:

```
<form action="/process_form" method="post">  
  <!-- campos de entrada do formulário -->  
  <input type="submit" value="Enviar">  
</form>
```

# O Elemento <select> em HTML

A tag <select> define uma lista suspensa (drop-down list).

Exemplo:

```
<form>
  <label for="fruits">Escolha uma fruta:</label>
  <select id="fruits" name="fruits">
    <option value="apple">Maçã</option>
    <option value="banana">Banana</option>
    <option value="orange">Laranja</option>
  </select>
  <input type="submit" value="Enviar">
</form>
```

Neste exemplo, criamos uma lista suspensa de frutas para seleção.

# Diferença entre GET e POST em Formulários HTML

Ao enviar dados de um formulário HTML para um servidor, você pode usar os métodos HTTP GET ou POST. Ambos os métodos têm propósitos e comportamentos diferentes.

- **GET:** Os dados do formulário são anexados à URL como parâmetros de consulta. Pode ser usado para solicitar dados do servidor. Os dados são visíveis na URL.
- **POST:** Os dados do formulário são enviados no corpo da solicitação HTTP. Pode ser usado para enviar dados confidenciais ou para modificar dados no servidor. Os dados não são visíveis na URL.

É importante escolher o método correto com base no objetivo da solicitação e na natureza dos dados enviados.

### Dados Pessoais:

Nome do Aluno:

Mascimento (dd/mm/aaaa):  /  /

Nome da Mãe:

Nome do Pai:

Telefone: DDD (  ) Tel:  Ramal:

E-Mail:

### Informações de Matrícula:

Série:

▼

Turno:

☐ Manhã

☐ Tarde

Atividades Extracurriculares:

☐ Informática

☐ Música

☐ Balet

☐ Pintura

☐ Judô

☐ Futebol

# Novos Elementos em Formulários HTML5

O HTML5 introduziu vários novos elementos para melhorar a semântica e funcionalidade dos formulários.

- `<input type="email">` - Campo de Email
- `<input type="url">` - Campo de URL
- `<input type="tel">` - Campo de Telefone
- `<input type="number">` - Campo Numérico
- `<input type="date">` - Campo de Data
- `<input type="color">` - Seletor de Cor
- `<input type="range">` - Controle de Intervalo
- `<datalist>` - Lista de Dados para Campos de Texto
- `<textarea>` - Área de Texto Multilinhas
- `<output>` - Resultado ou Cálculo de Formulário

# Campo de Email em HTML5 (<input type="email">)

O elemento <input type="email"> é usado para campos de entrada de endereço de email. Atributos importantes:

- name - Nome do campo.
- id - Identificador único do campo.
- required - Indica que o campo é obrigatório.
- placeholder - Texto de exemplo exibido no campo.
- pattern - Define uma expressão regular para validar o formato do email.

## Exemplo

```
<label for="email">Endereço de Email:</label>  
<input type="email" id="email" name="email" required  
      placeholder="seuemail@example.com"  
      pattern="[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,4}$">
```

# Campo de URL em HTML5 (<input type="url">)

O elemento <input type="url"> é usado para campos de entrada de URLs. Atributos importantes:

- name - Nome do campo.
- id - Identificador único do campo.
- required - Indica que o campo é obrigatório.
- placeholder - Texto de exemplo exibido no campo.
- multiple - Mais de uma entrada, seguida de (,).
- pattern - Define uma expressão regular para validar a URL.

## Exemplo

```
<label for="website">Website:</label>  
<input type="url" id="website" name="website" required  
  placeholder="https://www.example.com"  
  pattern="https?://.*">
```

# Campo de Telefone em HTML5 (<input type="tel">)

O elemento <input type="tel"> é usado para campos de entrada de números de telefone. Atributos importantes:

- name - Nome do campo.
- id - Identificador único do campo.
- required - Indica que o campo é obrigatório.
- placeholder - Texto de exemplo exibido no campo.
- pattern - Define uma expressão regular para validar o formato do telefone.

## Exemplo

```
<label for="phone">Telefone:</label>  
<input type="tel" id="phone" name="phone" required  
      placeholder="(XX) XXXX-XXXX"  
      pattern="\(\d{2}\) \d{4}-\d{4}">
```



# Campo Numérico em HTML5 (<input type="number">)

O elemento <input type="number"> é usado para campos de entrada numérica. Atributos importantes:

- name - Nome do campo.
- id - Identificador único do campo.
- min e max - Definem o intervalo mínimo e máximo de valores.
- step - Define o incremento para aumentar ou diminuir o valor.
- value - Define o valor inicial.
- required - Indica que o campo é obrigatório.

## Exemplo

```
<label for="quant">Quantidade:</label>  
<input type="number" id="quantity" name="quant"  
      min="1" max="10" step="1" value="1" required>
```

# Campo de Data em HTML5 (<input type="date"...>)

O elemento <input type="date"...> é usado para campos de entrada de datas. Atributos importantes:

- name - Nome do campo.
- id - Identificador único do campo.
- min e max - Definem o intervalo mínimo e máximo de datas.
- value - Define o valor inicial.
- required - Indica que o campo é obrigatório.

## Exemplo

```
<label for="dob">Data de Nascimento:</label>  
<input type="date" id="dob" name="dob"  
      min="1900-01-01" max="2023-12-31" required>
```

# Seletor de Cor em HTML5 (<input type="color"...>)

O elemento <input type="color"...> é usado para campos de seleção de cor. Atributos importantes:

- name - Nome do campo.
- id - Identificador único do campo.
- value - Define a cor inicial selecionada (formato hexadecimal).
- required - Indica que o campo é obrigatório.

## Exemplo

```
<label for="color">Escolha uma cor:</label>  
<input type="color" id="color" name="color"  
      value="#ff0000" required>
```

# Intervalo em HTML5 (<input type="range"...>)

O elemento <input type="range"...> é usado para criar controles de intervalo (faixas deslizantes). Atributos importantes:

- min e max - Definem o intervalo mínimo e máximo de valores.
- step - Define o incremento para ajustar o valor.
- value - Define o valor inicial.

## Exemplo

```
<label for="volume">Volume:</label>
<input type="range" id="volume" name="volume"
      min="0" max="100" step="1" value="50">
```

# Intervalo com Ticks (<input type="range"...>)

O elemento <input type="range"...> pode incluir marcas de seleção (ticks) para indicar valores específicos.

## Exemplo com Ticks

```
<label for="brightness">Brilho:</label>
<input type="range" id="brightness" name="brightness"
      min="0" max="100" step="10" value="50"
      list="tickmarks">
<datalist id="tickmarks">
  <option value="0">
  <option value="25">
  <option value="50">
  <option value="75">
  <option value="100">
</datalist>
```

# Elemento <datalist> em HTML5

O elemento <datalist> é usado para fornecer uma lista de opções predefinidas para um campo de entrada.

## Exemplo com <datalist>

```
<label for="browser">Navegador:</label>
<input type="text" id="browser" name="browser"
list="browsers">
<datalist id="browsers">
  <option value="Chrome">
  <option value="Firefox">
  <option value="Edge">
  <option value="Safari">
  <option value="Opera">
</datalist>
```

# Campo de Texto Multilinha em HTML5 (<textarea>)

O elemento <textarea> é usado para campos de texto multilinha.

Atributos importantes:

- rows e cols - Número de linhas e colunas do campo.
- placeholder - Texto de exemplo exibido no campo.
- required - Indica que o campo é obrigatório.
- maxlength - Número máximo de caracteres permitidos.

## Exemplo com <textarea>

```
<label for="comments">Comentários:</label>
<textarea id="comments" name="comments"
  rows="4" cols="40"
  placeholder="Digite seus comentários..."
  required maxlength="200"></textarea>
```

# Elemento <output> em HTML5

O elemento <output> é usado para exibir o resultado de um cálculo ou processamento.

## Exemplo com <output>

```
<form oninput="result.value=parseInt(a.value) +  
    parseInt(b.value)">  
  <label for="a">Número A:</label>  
  <input type="number" id="a" name="a" value="0">  
  +  
  <label for="b">Número B:</label>  
  <input type="number" id="b" name="b" value="0">  
  =  
  <output name="result">0</output>  
</form>
```



# Elementos <fieldset> e <legend>

Os elementos <fieldset> e <legend> são usados para agrupar e identificar elementos relacionados de um formulário.

## Exemplo com <fieldset> e <legend>

```
<form>
  <fieldset>
    <legend>Informações Pessoais</legend>
    <label for="name">Nome:</label>
    <input type="text" id="name" name="name">
    <br>
    <label for="email">E-mail:</label>
    <input type="email" id="email" name="email">
  </fieldset>
</form>
```

# Elementos <fieldset> e <legend>

