

# Programação Web

- ✓ HTML
- ✓ Conceitos iniciais e sintaxe básica
- ✓ Tags principais
- ✓ Formulários
- ✓ Exemplos
- ✓ Exercícios

**HTML (Hypertext Markup Language) = Linguagem de Marcação de Hipertexto**

A linguagem HTML desde a sua primeira especificação teve como **objetivo principal estruturar documentos**; sua especificação não visa a formatação ou função de apresentação visual como cor de fonte, aspectos de layout, etc.

Vale ressaltar que algumas tags possuem uma formatação prévia, porém são formatações que ainda estão dentro dos objetivos de estruturar o documento HTML.

## **CSS (Cascading Style Sheet) = Folha de Estilo em Cascata**

O CSS é uma linguagem que nos **permite adicionar estilos as nossas páginas**, em outras palavras, seria a **formatação visual** (aparência, layout) das páginas, como tipo de fonte, cores, espaçamentos, entre outros.

A W3C (World Wide Web Consortium) também especifica o CSS, existem diversos módulos que os navegadores aceitam em relação a última versão.

Com o uso do CSS externo, diminuímos o trabalho devido a possibilidade de reutilização dos estilos em diferentes tags e principalmente em diferentes páginas.

## Resumindo:

- O HTML é utilizado para estruturar as páginas, ou seja, definir parágrafos, listas numeradas, tabelas, cabeçalhos etc.
- O CSS é utilizado para formatar o visual da página (layout) estruturada com HTML, ou seja, adicionar a formatação de parágrafos, fontes, tabelas, entre outros.



A linguagem HTML precisa do CSS (folhas de estilo) para formatar o conteúdo de um documento e do Javascript para dar interatividade.

Um exemplo simples, as páginas abaixo possuem o mesmo código html, porém em uma delas aplicamos o CSS

## Título da Página

### Subtítulo

- [Home](#)
- [Artigos](#)
- [Sobre](#)
- [Contato](#)
- [Link 1](#)
- [Link 2](#)
- [Link 3](#)
- [Link 4](#)
- [Link 5](#)

### Título do artigo

Agora o Brasil é o segundo e tem oito pontos, dois a menos que os Estados Unidos e o mesmo que a Rússia, com vantagem no set average. Para confirmar a vice-liderança, a equipe de Bernardinho ainda precisa bater a Alemanha, quarta colocada, na rodada decisiva, para decidir quem é o segundo lugar nos critérios de desempate.

### Título do artigo

Agora o Brasil é o segundo e tem oito pontos, dois a menos que os Estados Unidos e o mesmo que a Rússia, com vantagem no set average. Para confirmar a vice-liderança, a equipe de Bernardinho ainda precisa bater a Alemanha, quarta colocada, na rodada decisiva, para decidir quem é o segundo lugar nos critérios de desempate.

SOMENTE HTML5

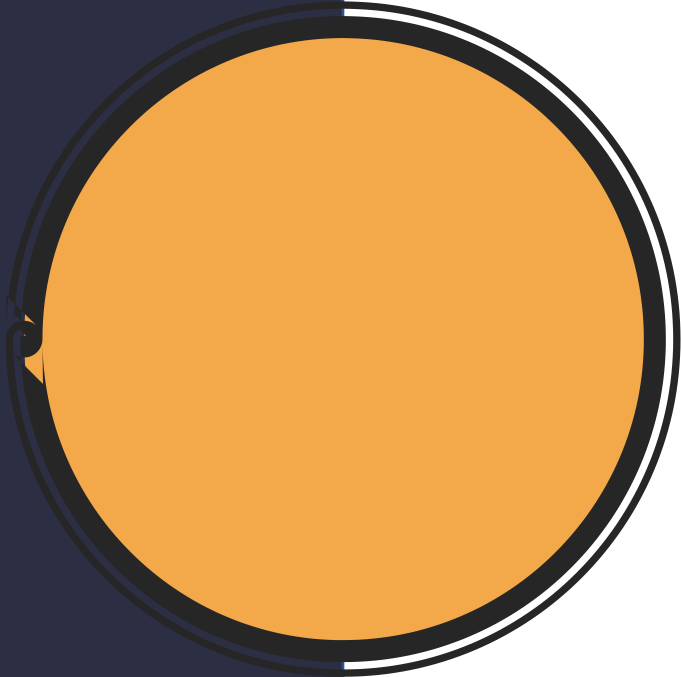
## Título da Página

### Subtítulo

| Home   | Artigos   | Sobre | Contato |
|--|---|-------|---------|
| <ul style="list-style-type: none"><li>• <a href="#">Link 1</a></li><li>• <a href="#">Link 2</a></li><li>• <a href="#">Link 3</a></li><li>• <a href="#">Link 4</a></li><li>• <a href="#">Link 5</a></li></ul> | <h3><u>Título do artigo</u></h3> <p>Agora o Brasil é o segundo e tem oito pontos, dois a menos que os Estados Unidos e o mesmo que a Rússia, com vantagem no set average. Para confirmar a vice-liderança, a equipe de Bernardinho ainda precisa bater a Alemanha, quarta colocada, na rodada decisiva, para decidir quem é o segundo lugar nos critérios de desempate.</p> <h3><u>Título do artigo</u></h3> <p>Agora o Brasil é o segundo e tem oito pontos, dois a menos que os Estados Unidos e o mesmo que a Rússia, com vantagem no set average. Para confirmar a vice-liderança, a equipe de Bernardinho ainda precisa bater a Alemanha, quarta colocada, na rodada decisiva, para decidir quem é o segundo lugar nos critérios de desempate.</p> |       |         |

HTML5 + CSS

# HTML



Geralmente uma tag é utilizada delimitando um texto, com sua abertura e fechamento, no HTML existem algumas tags que não possuem fechamento.

Exemplo de tag com fechamento

**<p> Está tag define um parágrafo </p>**

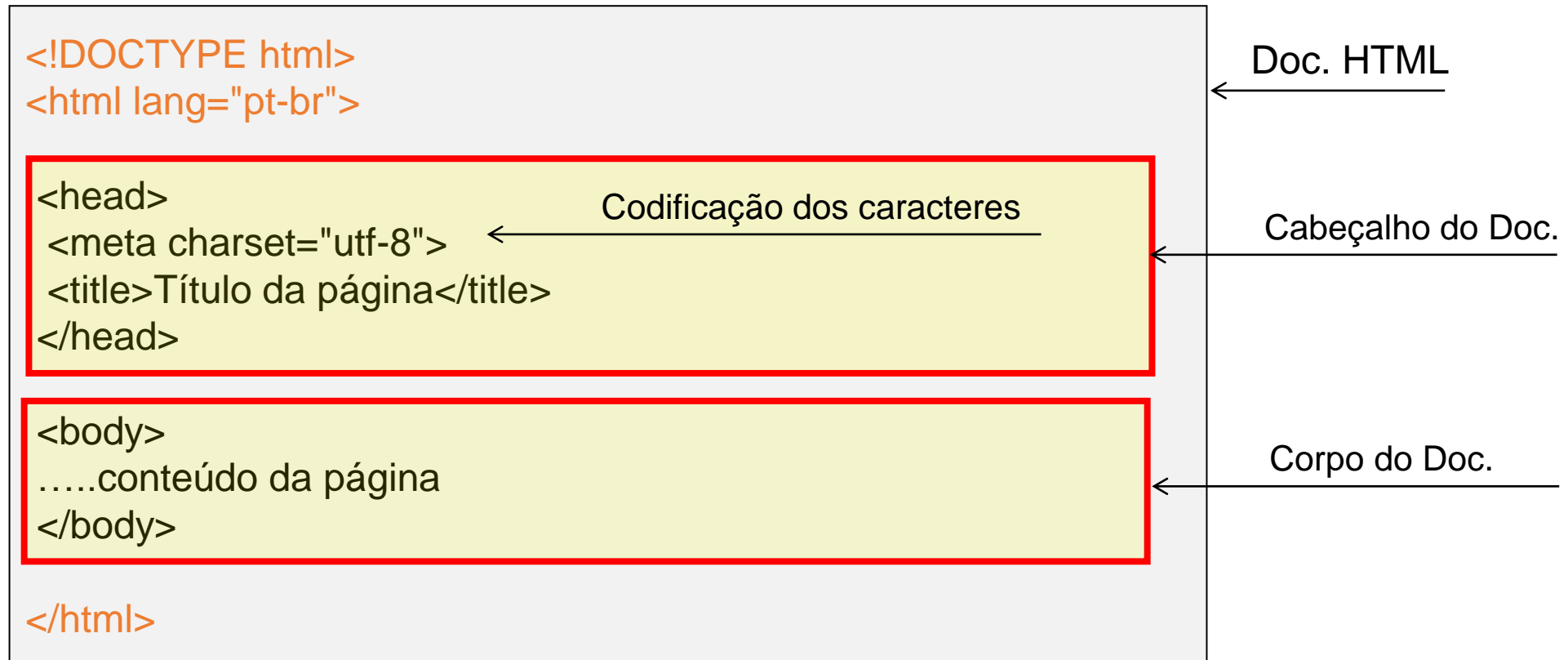
Exemplo de tag sem fechamento

Está tag quebra a linha. **<br>**

As tags são formadas pelos sinais "<" e ">", como no exemplo da estrutura básica. <html> .... </html>

Existem algumas tags que não são mais utilizadas devido ao uso do CSS.

Um documento HTML5 é formado basicamente pela estrutura:





- Os Elementos sempre ocupam um espaço quadrado na tela, que tem uma largura e uma altura. Quando estudarmos CSS veremos que eles podem ter seu tamanho alterado assim como ter sua posição natural alterada (podendo, inclusive, se sobreporem)
- Elementos podem conter outros Elementos (um bloco pode conter dois outros blocos menores dentro dele, por exemplo)
- O que define um elemento no HTML são as TAGs. Um elemento que pode ter conteúdo marcado deve ter uma TAG inicial e uma TAG de fechamento. Elementos que não devem ter conteúdo delimitado (como imagem ou linha horizontal) não possuem TAG de fechamento.

Para criarmos nossos documentos HTML, iremos seguir as observações abaixo:

- Devemos criar os documentos bem-formados.
- Todas as tags devem ser escritas com **letras minúsculas**.
- Uso de tags de fechamento é obrigatória.
- Atributos devem ser escritos também com **letras minúsculas**.
- Os valores dos atributos devem ser escritos dentro de aspas ("....").

Exemplo da tag `div` com um atributo `id`:

```
<div id="principal">Esta é o texto de uma div</div>
```

A HTML possui algumas codificações para caracteres, por exemplo, não se pode usar os caracteres "<" e ">" em um texto, pois o navegador pode confundi-los com as marcações do documento, devemos substituí-los por códigos.

Alguns exemplos de codificação e seus caracteres:

| Caracteres | Nome no HTML | Descrição                 |
|------------|--------------|---------------------------|
| "          | &quot;       | Aspas (quotation mark)    |
| '          | &apos;       | Apóstrofe                 |
| &          | &amp;        | "E" comercial (ampersand) |
| <          | &lt;         | Menor que (less-than)     |
| >          | &gt;         | Maior que (greater-than)  |
|            | &nbsp;       | Espaço (branco)           |

"Conteúdo é composto de uma sequência de caracteres. Caracteres representam letras do alfabeto, pontuação, etc. Conteúdos são armazenados em um computador como uma sequência de bytes, que são valores numéricos. Em alguns casos um simples caractere é representado por mais de um byte. Tal como os códigos usados em espionagem a maneira como uma sequência de bytes é convertida em caracteres depende do formato como o conteúdo foi codificado. Nesse contexto tal *formato* é denominado **codificação de caracteres**. Uma página HTML pode ter apenas uma codificação de caracteres. A codificação baseada em Unicode, tal como UTF-8, oferece suporte para vários idiomas e assim sendo admite páginas e formulários em qualquer combinação de idiomas." (W3C, 2014)

- No HTML5, a codificação padrão de caracteres é UTF-8.
- Para mais informações acesse:

<http://www.w3schools.com/charsets/>

<http://www.w3.org/International/O-charset.pt-br.php>

Tag usada para definir informações do documento como autor, palavras-chave etc. Vamos utilizar somente a tag meta para definir a codificação de caracteres das páginas e a visualização em aparelhos móveis.

Exemplos:

- especifica a codificação de caracteres da página

```
<meta charset="UTF-8">  
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Em linguagens de programação, espaços em branco (gerados pela tecla de espaço ou pelo tecla TAB do teclado) servem para separar os termos. Porém, mais de um espaço em sequência poderia ter o mesmo significado que um único espaço.

Assim é no HTML.

Por exemplo, os dois textos a seguir tem o mesmo resultado numa página HTML, porque espaços repetidos serão ignorados:

Este texto é separado por espaços únicos

Este    texto é separado            por            espaços    únicos

Em algumas linguagens de programação, a separação em linhas (inseridas pela tecla Enter do teclado) servem apenas para organizar o documento. Ou seja, se elas não forem colocadas, dá na mesma!

Assim é no HTML

Por exemplo, os dois textos a seguir tem o mesmo resultado numa página HTML (a utilização das tags *br* ou *p*, por exemplo, forçariam as quebras de linha) :

Tudo é apresentado em uma linha

Este texto está em uma linha

Este texto é  
está em  
uma linha

Criamos parágrafos com *p* e quebra de linha com *br*

`<p>Este texto é um parágrafo</p>`

Este texto <br>  
está em<br>  
várias linhas.

<p> - utilizada para delimitar um parágrafo

<p>Este conteúdo foi definido como um parágrafo</p>

<strong> - texto importante

<strong>em negrito</strong>

<em> - ênfase

<em>em itálico</em>

<br> - quebra de linha

Linha 1 <br>

Linha 2



<h1> até <h6> - cabeçalho, ou títulos

<h1>Título da página grande</h1>

<h2>Título da página menor</h2>

<h3>Título da página ainda menor</h3>

<hr> - quebra temática de linha

Linha 1

<hr>

Linha 2

Em HTML podem ser criadas listas ordenadas (numéricas, alfabéticas etc.) ou não ordenadas (utiliza marcadores como imagens).

<ol> - lista ordenada

```
<ol>  
  <li> item 1</li>  
  <li> item 2</li>  
</ol>
```

1. Item 1
2. Item 2
3. Item 3

<ul> - lista não ordenada

```
<ul>  
  <li> item 1</li>  
  <li> item 2</li>  
</ul>
```

- Item 1
- Item 2
- Item 3

Para definir uma tabela básica, utilize as tags:

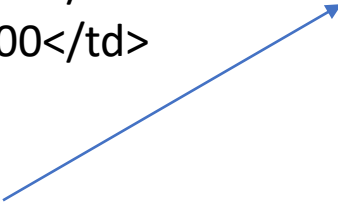
`<table>` define uma tabela

- `<tr>` define uma linha na tabela
- `<td>` define uma coluna na tabela

```
<table>
  <tr>
    <td>Linha 1 - Coluna 1</td>
    <td>Linha 1 - Coluna 2</td>
  </tr>
  <tr>
    <td>Linha 2 - Coluna 1</td>
    <td>Linha 2 - Coluna 2</td>
  </tr>
</table>
```

É possível construir uma tabela simples apenas com as tags **tr** e **td**:

```
<table border="1">
  <tr>
    <td>Mês</td>
    <td>Vendas</td>
  </tr>
  <tr>
    <td>JAN</td>
    <td>600</td>
  </tr>
  <tr>
    <td>FEV</td>
    <td>600</td>
  </tr>
  <tr>
    <td>Soma</td>
    <td>R$ 1200,00</td>
  </tr>
</table>
```



| Mês  | Vendas      |
|------|-------------|
| JAN  | 600         |
| FEV  | 600         |
| Soma | R\$ 1200,00 |

- Para definir uma tabela completa:

| Mês  | Vendas      |
|------|-------------|
| JAN  | 600         |
| FEV  | 600         |
| Soma | R\$ 1200,00 |

```
<table border="1">  
<thead>  
  <tr>  
    <th>Mês</th>  
    <th>Vendas</th>  
  </tr>  
</thead>  
<tfoot>  
  <tr>  
    <td>Soma</td>  
    <td>R$ 1200,00</td>  
  </tr>  
</tfoot>  
<tbody>  
  <tr>  
    <td>JAN</td>  
    <td>600</td>  
  </tr>  
  <tr>  
    <td>FEV</td>  
    <td>600</td>  
  </tr>  
</tbody>  
</table>
```

- 1 - <thead> = cabeçalho da tabela
  - 1.1 - <th> colunas do cabeçalho
- 2 - <tfoot> = última linha da tabela
- 3 - <tbody> = conteúdo da tabela

A função básica da tag DIV é de criar blocos de textos, porém também é utilizada na criação de layouts dos sites, para definir regiões genéricas da página, ou seja, regiões que não precisam de uma semântica (significado) no conteúdo. Também pode ser utilizada para marcar conteúdos que serão manipulados pelo JavaScript.

`<div>`

`<div>` Itens do cabeçalho do site`</div>`

`<div>` Itens do conteúdo do site`</div>`

`</div>`

## Olimpíadas 2012 - Londres

- Conteúdo
- Esportes
  - Quantidade de medalhas
  - Países

### Esportes

#### Atletismo

##### Descrição

O atletismo é o esporte mais tradicional do programa olímpico. Disputado desde a primeira edição, atualmente distribui 47 medalhas de ouro, 24 entre os homens e 23 entre as mulheres.

As provas são divididas em:

- *Campo* (saltos e arremessos)
- *Pista* (corridas com e sem obstáculos e revezamento)
- *Rua* (maratona e marcha)
- *Combinadas* (decatlo e heptatlo)

#### Basquete

##### Descrição

O objetivo é marcar o maior número de pontos, acertando uma cesta de 45 cm que fica a 3,05 m do chão. Cada cesta comum no basquete vale dois pontos. Lances livres (cobranças de infrações) valem um ponto. Arremessos a uma distância igual ou superior a 6,75 m da cesta valem três pontos. O jogo é disputado em quatro quartos de 10 minutos, e, a cada parada da bola, o cronômetro também para.

##### Quantidade de medalhas

| País         | Ouro | Prata | Bronze |
|--------------|------|-------|--------|
| China        | 29   | 16    | 14     |
| EUA          | 27   | 14    | 15     |
| Grã-Bretanha | 16   | 10    | 10     |

##### Países

1. China
2. EUA
3. Grã-Bretanha

Uma referência absoluta é aquela que inclui todo o caminho de um arquivo, incluindo o protocolo de comunicação (HTTP, por exemplo). Essa referência é válida sempre que o arquivo permaneça em um mesmo diretório.

Utilizam-se referências absolutas sempre que se deseja referenciar um arquivo que não faz parte da página construída.

```

```



Uma referência relativa é aquela que se expressa a partir de um diretório conhecido, dentro da mesma página.

no mesmo diretório:

```

```

ou na pasta filha `img`:

```

```

Caso a pasta "img" esteja um nível acima da pasta raiz, utiliza-se algo como:

```

```

Utilizado para ligação das páginas.

Exemplos:

- para acessar um arquivo que está dentro de uma pasta filha **paginas**, a partir do diretório principal do projeto

```
<a href="paginas/gato.html">Texto ou conteúdo do enlace</a>
```

- para acessar um arquivo que se encontra em um nível superior, a partir da pasta do arquivo atual

```
<a href="../cavalo.html"> Texto ou conteúdo do enlace</a>
```

- para acessar outros arquivos na web

```
<a href="http://www.microsoft.com.br/index.html ">Site da Microsoft Brasil</a>
```

**Nunca** crie seus links com o caminho físico do arquivo, ou seja, sua localização real na máquina, como **C:\site\index.html**, pois caso você disponibilize esse arquivo na Web e em outros sistemas como Linux, irá resultar em Página não encontrada.

## Imagens suportadas nas páginas Web


- GIF / GIF animado
  - ideal para desenhos, ícones, figuras simples e animações simples
  - Máximo de 256 cores
  - permite transparência
- JPG (a mesma coisa de JPEG)
  - ideal para banners de sites, imagens para compor o layout da página e fotos
  - mais de 16 milhões de cores
- PNG
  - formato livre para imagens
  - aceita mais de 16 milhões de cores
  - não perde resolução no processo de salvar a imagem
  - permite transparência

Testar exemploTAG\_IMG.html disponível no arquivo zipado exemplosHTML.

Quem nunca preencheu um formulário de cadastro na WEB? A ideia de formulário todos conhecem. Em resumo, sua principal utilidade é coletar os dados do usuário e fazer algum tipo de processamento no servidor, porém também podemos utilizar para processar algo em JavaScript, no computador cliente.

Podemos criar nossos formulários através do HTML utilizando tags para cada tipo de elemento de entrada de dados. No HTML5 também temos alguns atributos que irão facilitar a validação do formulário sem a necessidade de desenvolver scripts em outras linguagens, ou seja, para validar alguns dados não precisaremos utilizar o JavaScript diretamente.

|                                       |                                       |
|---------------------------------------|---------------------------------------|
| Nome                                  | <input type="text" value="Nome"/>     |
| Email                                 | <input type="text" value="Email"/>    |
| Telefone                              | <input type="text" value="Telefone"/> |
| Assunto                               | <input type="text" value="Assunto"/>  |
| Mensagem                              | <input type="text" value="Mensagem"/> |
| <input type="button" value="Enviar"/> |                                       |

|                                     |   |   |
|-------------------------------------|---|---|
| Email:                              | <input type="text" value="name@example.com"/> |  |
| Password:                           | <input type="password" value="....."/>        |   |
| <a href="#">Esqueceu sua senha?</a> |   |   |

Como a linguagem HTML5 ainda está em desenvolvimento, alguns atributos e comandos de formulários não são compatíveis com todos os navegadores, por isso iremos utilizar somente o que for compatível com pelo menos dois navegadores (Google Chrome e Firefox).

Para criarmos um formulário podemos utilizar a tag `<form>` ou em alguns casos somente o elemento como um botão para disparar uma ação na nossa página.

## Tag form

```
<form name="form1" method="post" action="">
```

Elementos do formulário

```
</form>
```

get ou post

Endereço, página  
do servidor

Tag para entrada de dados, que poderá ter diversas variações, conforme exemplos abaixo:

Exemplos:

Nome: `<input type="text" name="nome">`

Senha: `<input type="password" name="senha">`

E-mail: `<input type="email" name="usermail">`

URL: `<input type="url" name="homepage">`

Submeter o form: `<button type="submit">Enviar</button>`

Campo oculto

```
<input name="oculto" type="hidden" value="algumacoisa">
```

Checkbox (o usuário pode marcar mais de uma opção)

```
<input type="checkbox" name="carros" value="Camaro">Camaro <br>  
<input type="checkbox" name="carros" value="Ferrari">Ferrari
```

Button para chamar funções em Javascript

```
<button type="button">Clique aqui</button>
```

Omitimos a tag label para facilitar o exemplo.

Observe que o valor do argumento *name* é o mesmo para todos os checkboxes.

File para carregar arquivos no formulário

Escolha um arquivo `<input name="arquivo" type="file">`

Image para definir um botão com imagem para enviar o formulário

`<input name="imagem" type="image" src="images.jpg">`

Radio (neste tipo o usuário poderá marcar somente uma opção desde que o name seja o mesmo para os elementos do mesmo grupo)

`<input type="radio" name="genero" value="F">Feminino <br>`

`<input type="radio" name="genero" value="M">Masculino`

Omitimos a tag label para facilitar o exemplo.

Observe que o valor do argumento *name* é o mesmo para todos os radiobuttons.



Nos exemplos anteriores mostramos a utilização dos argumentos ou atributos *id* e *name* nos diferentes tipos de tags *input*.

```
<input type="radio" name="sexo" id="masculino" value="m" checked>
```

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

O atributo **id** é frequentemente utilizado em JavaScript para acessar os elementos do documento (da página) e efetuar algum processamento com os mesmos no computador do cliente.

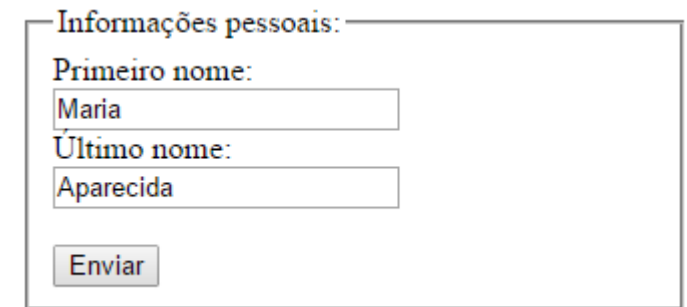
O atributo **name** é frequentemente utilizado em elementos de formulários (Form) que serão enviados para um servidor Web, permitindo desta forma o processamento dos dados em alguma "página destino".

O atributo **name** possui características de vetor, sendo interessante sua utilização em componentes como listas de itens e grupos de checkboxes, permitindo que a "página destino" consiga recuperar os valores selecionados pelo usuário na "página inicial".

FIELDSET é utilizado para agrupar elementos ou itens de formulários com características em comum, o Chrome por exemplo, delimita essa tag com uma borda, lembrando que isso pode ser formatado em CSS

LEGEND é usado para definir uma legenda para o fieldset

```
<form action="pagina.php">  
  <fieldset>  
    <legend>Informações pessoais: </legend>  
    Primeiro nome:  
    <br>  
    <input type="text" name="firstname" value="Maria">  
    <br> Último nome:  
    <br>  
    <input type="text" name="lastname" value="Aparecida">  
    <br>  
    <br><button type="submit">Enviar</button>  
  </fieldset>  
</form>
```



Informações pessoais:

Primeiro nome:  
Maria

Último nome:  
Aparecida

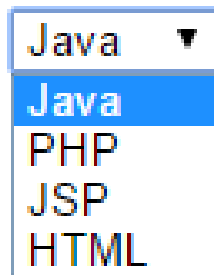
Enviar

Nesta tag você poderá colocar texto ou imagens como um botão *clicável*. Este comando pode ser utilizado fora de um formulário.

```
<button type="button">  
    
</button>
```

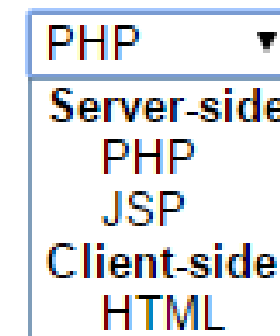
Cria uma lista de opções (conhecido como combobox)

```
<form>
  <select>
    <option value="Java">Java</option>
    <option value="PHP">PHP</option>
    <option value="JSP">JSP</option>
    <option value="HTML">HTML</option>
  </select>
</form>
```



|        |
|--------|
| Java ▼ |
| Java   |
| PHP    |
| JSP    |
| HTML   |

```
<form>
  <select>
    <optgroup label="Server-side">
      <option value="PHP">PHP</option>
      <option value="JSP">JSP</option>
    </optgroup>
    <optgroup label="Client-side">
      <option value="HTML">HTML</option>
    </optgroup>
  </select>
</form>
```



|             |
|-------------|
| PHP ▼       |
| Server-side |
| PHP         |
| JSP         |
| Client-side |
| HTML        |

## Textarea

Define um campo de texto de várias linhas

```
<form>
```

```
  <textarea rows="8" cols="70" name="msg"></textarea>
```

```
</form>
```

| Atributo  | Descrição  |
|-----------|--|
| disabled  | Especifica que o campo está desabilitado, o usuário não pode alterar e o mesmo não é enviado para o servidor                     |
| maxlength | Especifica o número máximo de caracteres que podemos digitar no elemento   |
| pattern   | Especifica uma expressão regular, um formato pré-determinado   |
| readonly  | Especifica que o campo está somente em modo leitura, semelhante ao disabled, porém neste caso, o valor é enviado para o servidor |
| required  | Especifica que o campo é obrigatório   |
| size      | Especifica o tamanho (largura) do campo em caracteres  |
| value     | Especifica um valor padrão inicial para o campo  |

Este atributo pode ser utilizado em campos para obrigar a digitação dos dados. A mensagem será ativada quando o usuário tentar enviar o formulário.

Pode ser utilizado em diversas tags de entrada de dados no formulário.

```
<input name="nome" required>
```

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

Este atributo identifica o elemento ao qual será dado o foco quando a página for carregada. Pode ser utilizado em diversos elementos de um formulário.

```
<form action="x.php">
```

```
  Nome:<input type="text" name="nome" autofocus><br>
```

```
  Idade: <input type="text" name="idade"><br>
```

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

```
</form>
```



Atributo exclusivo de elementos input e textarea. Define uma dica para digitação no campo. Assim que o usuário começar a digitar, a dica é automaticamente retirada.

```
<form action="demo_form.asp">  
  <input type="text" name="user" placeholder="Digite seu Usuário"><br>  
  <input type="text" name="senha" placeholder="Digite sua Senha"><br>  
  <button type="submit">Enviar</button>  
</form>
```

<form action="processa.php" method="post">

Nome:

Sexo: ☒ M ☐ F

E-mail:

Assunto:

Curso:

Mensagem:

</form>

```
Processa.php
<?php
header('Content-Type: text/html; charset=utf-8');
echo "Dados recebidos do formulário!!!<br/><br/>";
echo "Você digitou:<br/>";
echo "Nome: " .$_POST["nome"]."<br />";
echo "Sexo: " .$_POST["sexo"]."<br />";
echo "E-mail: " .$_POST["email"]."<br />";
echo "Assunto: " .$_POST["assunto"]."<br />";
echo "Curso: " .$_POST["curso"]."<br />";
echo "Mensagem: " .$_POST["mensagem"]."<br />";
?>
```

Exibe no navegador

Dados recebidos do formulário!!!

Você digitou:

Nome: ALCIDES TEIXEIRA BARBOZA JUNIOR

Sexo: M

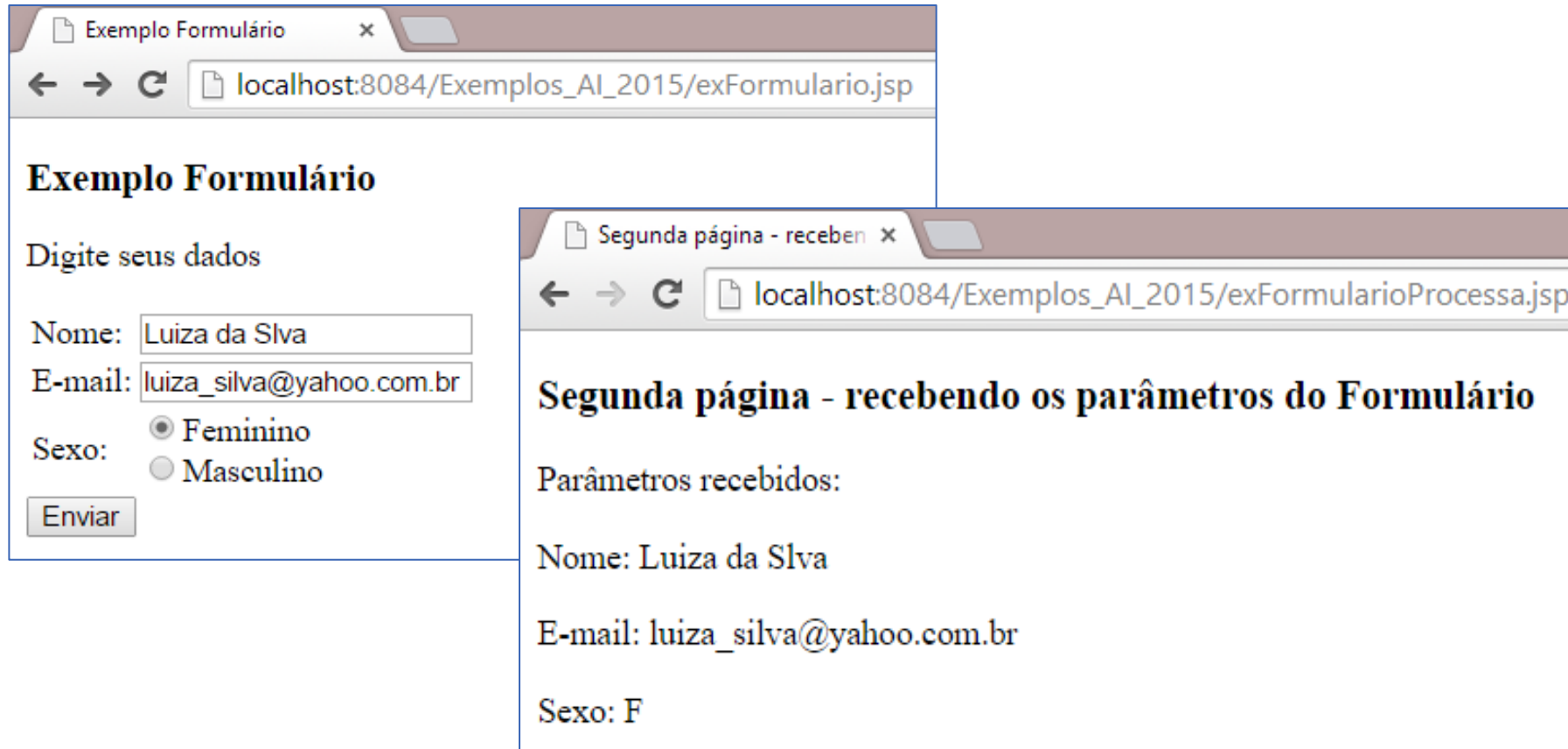
E-mail: alcidesprof@html.css

Assunto: Teste de Aula

Curso: TJD

Mensagem: Dúvida sobre o bloco de notas...

Para rodar esse exemplo completo precisamos ter o Servidor Web + PHP



The image displays two browser windows side-by-side. The left window, titled 'Exemplo Formulário', shows a web form at 'localhost:8084/Exemplos\_AI\_2015/exFormulario.jsp'. The form contains fields for 'Nome' (filled with 'Luiza da Silva'), 'E-mail' (filled with 'luiza\_silva@yahoo.com.br'), and 'Sexo' (with 'Feminino' selected). An 'Enviar' button is at the bottom. The right window, titled 'Segunda página - recebendo', shows the processing page at 'localhost:8084/Exemplos\_AI\_2015/exFormularioProcessa.jsp'. It displays the received parameters: 'Nome: Luiza da Silva', 'E-mail: luiza\_silva@yahoo.com.br', and 'Sexo: F'.

**Exemplo Formulário**

Digite seus dados

Nome:

E-mail:

Sexo: ☒ Feminino ☐ Masculino

**Segunda página - recebendo os parâmetros do Formulário**

Parâmetros recebidos:

Nome: Luiza da Silva

E-mail: luiza\_silva@yahoo.com.br

Sexo: F

Para rodar esse exemplo completo precisamos ter o Servidor Web Tomcat ou GlassFish

```
<body>
  <h3>Exemplo Formulário</h3>
  <form action="exFormularioProcessa.jsp" method="post">
    <p>Digite seus dados</p>
    <table>
      <tr>
        <td> Nome: </td>
        <td>
          <input name="nome" maxlength="20" autofocus="autofocus" required="required" />
        </td>
      </tr>
      <tr>
        <td> E-mail: </td>
        <td>
          <input name="email" maxlength="15" required="required" />
        </td>
      </tr>
      <tr>
        <td> Sexo: </td>
        <td>
          <input type="radio" name="sexo" value="F" checked />Feminino <br/>
          <input type="radio" name="sexo" value="M" />Masculino
        </td>
      </tr>
    </table>
    <input type="submit" value="Enviar" />
  </form>
```

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Segunda página - recebendo os parâmetros do Formulário</title>
  </head>
  <body>
    <h3>Segunda página - recebendo os parâmetros do Formulário</h3>
    <%
      String nome = request.getParameter("nome");
      String email = request.getParameter("email");
      String sexo = request.getParameter("sexo");
    %>
    <p>Parâmetros recebidos:</p>
    <p>Nome:  <%=nome%> </p>
    <p>E-mail: <%=email%> </p>
    <p>Sexo:  <%=sexo%> </p>
  </body>
</html>
```

Crie um formulário HTML para ler os dados abaixo, este formulário deve ser enviado para o endereço:

<http://aplicacoesparainternet.siteoficial.ws/validaDados.php>

Os dados que o código PHP (servidor) está preparado para tratar são:

- 1 – RGM
- 2 – Ano de nascimento
- 3 – Peso
- 4 – Altura
- 5 – Curso

OBS: o script PHP está preparado para receber somente dados por meio do método post.

form.html

RGM:

Ano de nascimento:

Peso:

Altura:

Curso:



validaDados.php

**Teste básico da aula de  
Programação Web (Prof Alcides)**

Variáveis necessárias: rgm, ano, peso, altura e curso

RGM: 123456

Idade: 41

IMC: 33.087076783026

Curso: SM-CCP.

Vamos testar este exemplo fazendo nosso form seguindo as definições acima.

```
! Ou html:5
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>Document</title>
  </head>
  <body>

  </body>
</html>
```

```
link:css
<link rel="stylesheet" href="style.css" />
```

```
ul>li*5
<ul>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
</ul>
```

```
ul>li.item$*5
<ul>
  <li class="item1"></li>
  <li class="item2"></li>
  <li class="item3"></li>
  <li class="item4"></li>
  <li class="item5"></li>
</ul>
```

```
style
<style></style>

script
<script></script>

script:src
<script src=""></script>
```

```
table>tr>td
<table>
  <tr>
    <td></td>
  </tr>
</table>
```

No arquivo cheatsheet-a5.pdf tem vários atalhos.