

Introdução às Linguagens de Programação

Recursos e Linguagens de Programação para Web
Linguagem de marcação xHTML

Prof^a. Me. *Aline de Campos*

HyperText Markup Language é uma
linguagem de marcação.

Criada por **Tim Berners-Lee** para comunicação e
divulgação de pesquisas, utilizando-se da **internet**.

São **interpretados** por navegadores web.

Primeiras versões: regras sintáticas flexíveis
Atualmente: a sintaxe muito mais rígida

Recebeu especificações formais na **década de 90**

Representado por um conjunto de tags

Comandos de formatação de linguagem

Padrão: <tag> conteúdo </tag>

Tags podem conter atributos e valores

Exemplo: conteúdo

↑ ↑
ATRIBUTO VALOR

eXtensible HyperText Markup Language

Evolução natural da tecnologia HTML.

Principal aspecto:

Melhoria da construção do código e separação de código de marcação de código de formatação visual, ou seja, para um código xHTML padrão W3, não se fazem mais ajustes visual em código fonte html e sim em CSS.

DEFINIÇÕES DO DOCUMENTO

CABEÇALHO (HEAD)

CORPO (BODY)

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"
      lang="pt-br" xml:lang="pt-br">

  <head>
    <meta http-equiv="Content-Type"
          content="text/html; charset=iso-8859-1" />

    <title>Strict</title>
  </head>
  <body>

    </body>
</html>
```

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"
      lang="pt-br" xml:lang="pt-br">

  <head>
    <meta http-equiv="Content-Type"
          content="text/html; charset=iso-8859-1" />

    <title>Strict</title>
  </head>
  <body>

    </body>
</html>
```

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"
      lang="pt-br" xml:lang="pt-br">

  <head>
    <meta http-equiv="Content-Type"
          content="text/html; charset=utf-8" />

    <title>Strict</title>
  </head>
  <body>

  </body>
</html>
```


Modelo Transitional UTF

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"
      lang="pt-br" xml:lang="pt-br">

  <head>
    <meta http-equiv="Content-Type"
          content="text/html; charset=utf-8" />

    <title>Strict</title>
  </head>
  <body>

    </body>
</html>
```

Não utilize espaços em branco, acentuação e caracteres especiais no nome de arquivos. O padrão de funcionamento da web não permite esta prática, caracteres serão convertidos em codificações que podem ocasionar erros.



HTML, xHTML

Formatação de textos



<html>

DEFINIÇÕES DO DOCUMENTO

Delimita o início e o fim do documento HTML

<head>

CABEÇALHO (HEAD)

Para indicação de parâmetros de configuração do documento

</head>

<body>

CORPO (BODY)

Corpo do documento onde será inserida toda a codificação.

</body>

</html>

```
<!-- Um comentário aqui -->
```

```
<!--
```

```
Um comentário aqui com mais  
de uma linha
```

```
-->
```

Parágrafos, Cabeçalhos e Títulos

Tipo	Codificação	Descrição
Parágrafo	<p> </p>	Abarca conteúdos inserindo um espaço antes e depois do parágrafo.
Títulos	<h<i>n</i>> </h<i>n</i>>	Introduzem títulos no documento, podem ser de seis níveis (1 até 6), seu tamanho de fonte varia de 24 à 10 pts.
Divisão	<div></div>	Utilizado com folhas de estilo na especificação de blocos e texto.
Horizontal Rule	<hr/>	Insere uma linha horizontal.
Preserve	<pre> </pre>	Utilizado para preservar a formatação do texto, não interpretando seu conteúdo.

Gerador de texto aleatório <http://www.lipsum.com/>

Parágrafos, Cabeçalhos e Títulos

<h1> </h1>

<h2> </h2>

<h3> </h3>

<h4> </h4>

<h5> </h5>

<h6> </h6>

Maior



Menor

Tags que abrem e fecham em si mesmo

Sintaxe: `<nomedatag />`

Tipo	Codificação	Descrição
Area	<area />	Na criação de mapeamento de imagens.
base	<base />	Aponta para o endereço base do site.
br	
	Insere uma quebra de linha.
hr	<hr />	Insere uma linha horizontal.
input	<input />	Insere um elemento que varia a apresentação quanto ao seu tipo
img		Insere uma imagem.
link	<link />	Inclui arquivos externos como por exemplo definições de CSS.
meta	<meta/>	Utilizado para definir uma meta-tag
col	<col/>	Utilizado na definição de tabelas.
param	<param/>	Utilizado na definição de parâmetros de objetos ou applets.
frame	<frame/>	Na criação de framesets.



Tipos variados de listas



Elementos para criação de listas

Tipo	Nome literal	Codificação	Descrição
Lista sem ordenação	Unordered list	 	Cria a estrutura que vai abarcar os itens de uma lista sem ordenação.
Lista com ordenação	Ordered list	 	Cria a estrutura que vai abarcar os itens de uma lista com ordenação.
Item de lista	List item	 	Cada estrutura de irá corresponder a um item da lista.
Lista de definição	Definition list	<dl> </dl>	Cria a estrutura que vai abarcar os itens de uma lista de definição.
Termo definição	Definition term	<dt> </dt>	Cria a estrutura de termos de definição de uma lista de definição.
Descrição de definição	Definition description	<dd> </dd>	Cria os itens de descrição de definição de uma lista de definição.

As listas não ordenadas são criadas fazendo-se uso do marcador ` `, e dos elemento ` `.

Estilos: atributo “type” pode apresentar três tipos:

- círculo vazio (circle),
- círculo cheio (disc)
- quadrado (square)

```
<ul>  
  <li type="circle">Item 1</li>  
</ul>
```

```
<!-- ul = unordered list -->
```

```
<ul>
```

```
    <!-- li - list item -->
```

```
    <li> Item 1 </li>
```

```
    <li> Item 2 </li>
```

```
    <li> Item 3 </li>
```

```
</ul>
```

Possuem elementos que são numerados, o marcador utilizado é o `` `` e para definição de cada elemento utiliza-se o `` ``.

Estilos: atributo “type” pode apresentar três tipos

arábico

“1” 1, 2, 3...

romano

“i” i, ii, iii...

“I” I, II, III...

alfabético

“a” a, b, c...

“A” A, B, C...

``

`<li type=“a”> Item 1 `

``

```
<!-- ol = ordered list -->
<ol>
    <!-- li - list item -->
    <li> Item 1 </li>
    <li> Item 2 </li>
    <li> Item 3 </li>
</ol>
```

Utilizadas na definição de termos, o marcador utilizados é o `<dl> </dl>`, o marcador a ser utilizado no texto a ser definido é o `<dt> </dt>` e a definição dos termos é feita através do marcador `<dd> </dd>`.

```
<dl>  
    <dt> Item 1 </dt>  
</dl>
```

```
<!-- dl = definition list -->
<dl>
  <!-- dt - definition term -->
  <dt> Item 1 </dt>
  <!-- dd - definition description -->
    <dd> Item 1.1 </dd>
    <dd> Item 1.2 </dd>
  <dt> Item 2 </dt>
    <dd> Item 2.1 </dd>
    <dd> Item 2.2 </dd>
</dl>
```




Imagens Hiperlinks Navegação entre páginas



src

Indica o caminho de onde está a imagem. Uma imagem tem geralmente as seguintes extensões: jpg, gif ou png.

width

Indica a largura que a imagem deve ter. Pode ser expressa em várias unidades. Geralmente pixel ou porcentagem.

height

Indica a altura que a imagem deve ter. Pode ser expressa em várias unidades. Geralmente pixel ou porcentagem.

alt

Texto que irá surgir quando o mouse for colocado sobre a imagem. Sua utilização é imprescindível para acessibilidade do site.

```
<!-- inserção de imagens -->  

```

Âncoras

São links para navegação dentro do mesmo documento. Assim uma ligação pode apontar para o mesmo documento no qual foi inserida

Links

As ligações hipertexto são figuras ou trechos de textos que quando clicados, ou acionados, re-direcionam o navegador ou leitor XHML à um outro documento, ou a outro ponto do documento atual. Neste contexto há o conceito de página de origem, na qual existe a ligação, e a página destino, para a qual a ligação aponta.

Criação de âncora:

```
<a name="ancora"> Texto </a>
```

Criação de referência:

```
<a href="#ancora"> Veja o texto </a>
```

Hiperlinks – Links absolutos e relativos

` Ligação `

Link relativo:

` Aulas `

Link absoluto:

` Aulas
`

Dica:

Preste atenção sempre a hierarquia das pastas e arquivos na hora de referências nos links.

href Referência do hiperlink. Indica o caminho para onde o link irá apontar.

name Nome do hiperlink. Utilizado para criação de âncoras para navegação em uma mesma página.

target De que maneira o link irá se comportar ao ser clicado. Ou seja, de que forma o caminho irá ser mostrado. Em geral na mesma página ou em uma nova página.

Resumindo as tags desta aula

Tipo	Codificação	Descrição		
Imagem		Inserção de imagens em documentos HTML.		
		src	fonte	caminho onde se encontra a imagem
		width	largura	indicação da largura da imagem (px ou %)
		height	altura	indicação da altura da imagem (px ou %)
		alt	texto	descrição da imagem
Hiperlink	<a> 	Inserção de ligações num mesmo documento ou para navegação entre diversos documentos HTML.		
		href	referência de hiperlink	
		target	indica o comportamento da ação de navegação (_blank, _top)	
		name	nome de referência de uma âncora	



Tabelas webstandards



O que são tabelas?

São um recurso para **estruturação de dados** de forma **tabular**.

Separadas por **linhas** e agrupadas por **colunas**. Linhas tem a finalidade de separar os elementos entre si, as colunas servem para apresentar separadamente informações pertinentes a cada um dos elemento.

Quando usar tabelas?

Recomendada quando houver **necessidade** de apresentação de **grandes quantidades de conteúdo**. Por exemplo, relatórios, planilhas, apresentação de resultados e etc.



O que é tableless?

Tableless (sem tabelas) é a adoção de outros elementos, que não tabelas, para criação de estruturas gerais de sites. Ou seja, para o desenvolvimento de sites não se deve utilizar tabelas para disposição de conteúdo na página.




table Tag para criação de uma tabela.

thead Tag para criação de uma estrutura de cabeçalho.

tbody Tag para criação de estrutura de corpo da tabela.

tfoot Tag para criação de uma estrutura de rodapé.

Atenção: Se inserir um cabeçalho e rodapé, deve-se obrigatoriamente inserir o corpo. Ou seja, Sem cabeçalho ou rodapé a estrutura tbody é desnecessária.

Estrutura de tabela com cabeçalho e rodapé

```
<table>
  <thead>
    <tr>
      <td></td>
    </tr>
  </thead>
  <tfoot>
    <tr>
      <td></td>
    </tr>
  </tfoot>
  <tbody>
    <tr>
      <td></td>
    </tr>
  </tbody>
</table>
```

Corresponde a uma linha com uma coluna no cabeçalho.

Corresponde a uma linha com uma coluna no rodapé.

Estrutura de tabela com cabeçalho e rodapé

Estrutura de tabela com cabeçalho e rodapé

• • •

```
<tbody>
```

```
  <tr>
```

```
    <td></td>
```

```
    <td></td>
```

```
  </tr>
```

```
  <tr>
```

```
    <td></td>
```

```
    <td></td>
```

```
  </tr>
```

```
</tbody>
```

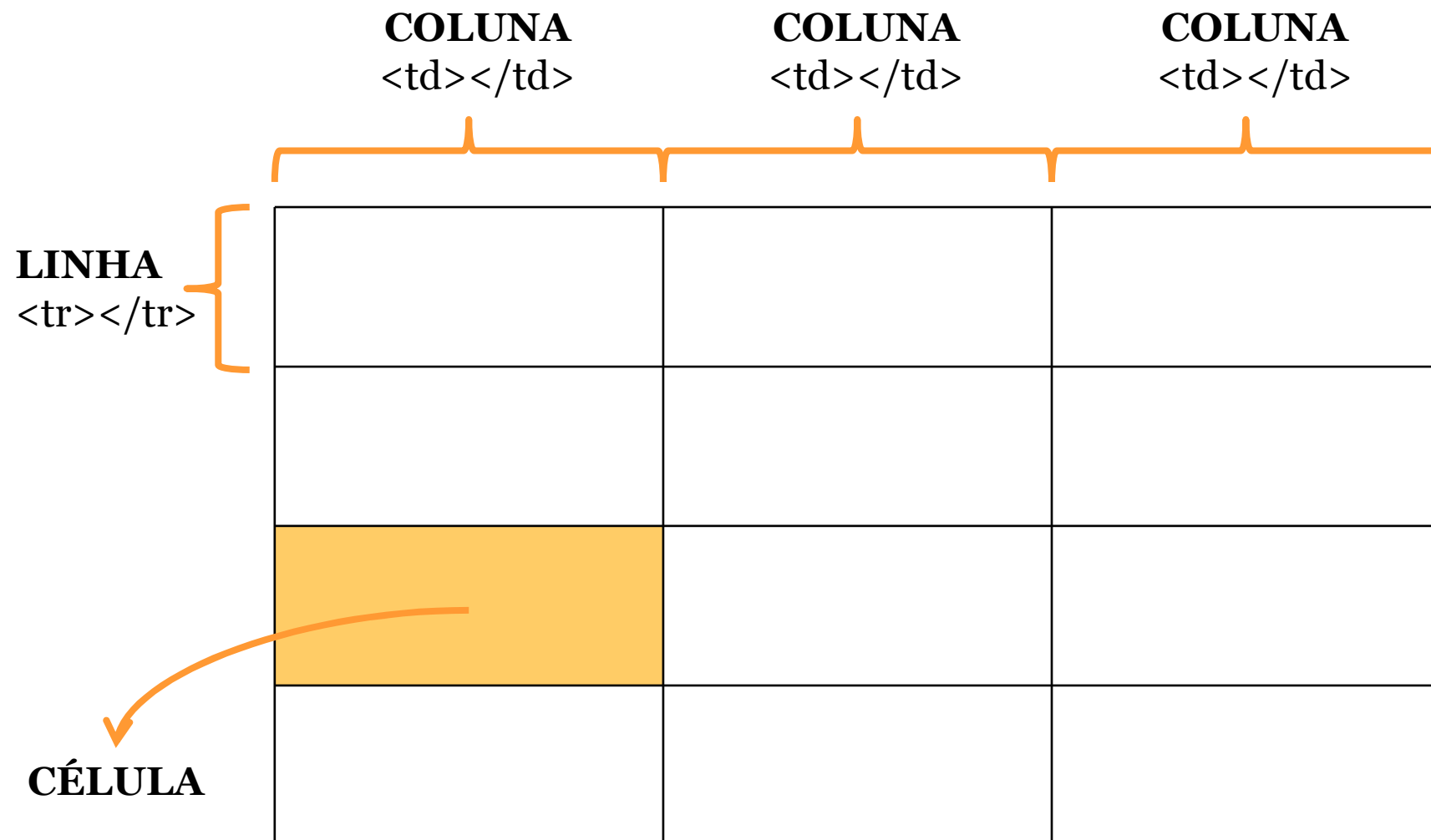
• • •

Corresponde a uma linha com duas colunas no corpo da tabela.

Corresponde a uma linha com duas colunas no corpo da tabela.

* Ou seja, no corpo dessa tabela existem duas linhas com duas colunas cada

Estrutura de tabela simples




Tags e atributos

Tipo	Codificação	Descrição		
Tabela	<table> </table>	Estrutura principal de uma tabela.		
		border	borda	
		width	largura	indicação da largura da tabela (px ou %)
Linha	<tr> </tr>	Insere uma nova estrutura de linha dentro da tabela.		
		align	alinhamento horizontal (left, center, right)	
		valign	alinhamento vertical (top, middle, bottom)	
Coluna	<td> </td>	Insere uma nova coluna dentro da estrutura de linha.		
		align	alinhamento horizontal (left, center, right)	
		valign	alinhamento vertical (top, middle, bottom)	
		colspan	número de colunas que a célula deverá se expandir	
		rowspan	número de linhas que a célula deverá se expandir	

Atributos colspan e rowspan

Expansão de coluna: `<td colspan="2"> Texto </td>`

Linha 1 – Coluna 1	Linha 1 – Coluna 2
Linha 2 – Coluna 1	Linha 2 – Coluna 2



Linha 1 – Coluna 1 EXPANDIDA	
Linha 1 – Coluna 1	Linha 1 – Coluna 1

```
<tr>
  <td> L1-C1 </td>
  <td> L1-C2 </td>
</tr>
<tr>
  <td> L2-C1 </td>
  <td> L2-C2 </td>
</tr>
```

```
<tr>
  <td colspan="2">
    L1 – C1 EXPANDIDA
  </td>
</tr>
<tr>
  <td> Linha 2 – Coluna 1 </td>
  <td> Linha 2 – Coluna 2 </td>
</tr>
```

Atributos colspan e rowspan

Expansão de linha: `<td rowspan="2"> Texto </td>`

Linha 1 – Coluna 1	Linha 1 – Coluna 2
Linha 2 – Coluna 1	Linha 2 – Coluna 2

Linha 1 – Coluna 1 EXPANDIDA	Linha 1 – Coluna 2
	Linha 2 – Coluna 2



```
<tr>
  <td> L1-C1 </td>
  <td> L1-C2 </td>
</tr>
<tr>
  <td> L2-C1 </td>
  <td> L2-C2 </td>
</tr>
```

```
<tr>
  <td rowspan="2">
    L1 – C1 EXPANDIDA
  </td>
  <td> L1-C2 </td>
</tr>
<tr>
  <td> L2-C2 </td>
</tr>
```

Exercícios desta aula

- Crie um arquivo chamado tabelas.html a partir do modelo Strict UTF

** Encontre seu modelo Strict UTF (msutf.html)*

** Copie-o e cole na pasta que desejar*

** Renomeie este arquivo para tabelas.html*

- Construa as tabelas apresentadas abaixo:

Matrícula	Nome	Curso
0001	João da Silva	Informática
0002	Maria Oliveira	Informática
0003	Pedro Pereira	Redes
Todos	Todos	Todos

thead

tfoot

** Tabela sem cabeçalho e rodapé*

Aluno	Curso	Média			Situação
João	Informática	Nota 1	Nota 2	Nota 3	
		9,5	10,0	8,0	Aprovado



Estruturas de formulários e componentes



Estrutura de tabela com cabeçalho e rodapé

Um formulário HTML é uma seção de um documento que composto por conteúdo, marcação, elementos especiais chamados “controles”, labels (rótulos) e seus controles. Em geral usuários completam um formulário modificando seus controles (inserindo textos, selecionando itens em um menu...), antes de submeter o formulário para um agente de processamento (um servidor web, um servidor de e-mail, uma aplicação em linguagem dinâmica...)

A tag HTML **form** tem como atributo o método que especifica a forma de passagem de parâmetros daquele formulário.

```
<form>  
    ...  
</form>
```

Sintaxe

Atributo	Valores
name	Identificador para o formulário
action	Indica para onde o formulário será enviado
method	POST ou GET

```
<fieldset>  
  <legend> título </legend>  
  ...conteúdo...  
</fieldset>
```


Métodos GET e POST

O método **GET** se utiliza da URL para passagem dos parâmetros, porém apresenta dois problemas:

- 1) Passagem limitada: máximo de 1024 caracteres, assim pode ocorrer perda de dados.
- 2) Segurança: todos os dados que serão passados pelo formulário podem ser visualizados na URL.

Um método que pode corrigir estes problemas é o **POST**, que não exibe as informações passadas em parâmetros da URL e tem uma capacidade de 2 kb de informações.

```
<html>
  <body>
    <!-- USANDO MÉTODO POST -->
    <form action="pagina.php" method="POST">
      <input name="arg" type="text" value="valor">
      <input name="enviar" type="submit" value="OK">
    </form>
  </body>
</html>
```

```
<html>
  <body>
    <!-- USANDO MÉTODO POST -->
    <form action="pagina.php" method="GET">
      <input name="arg" type="text" value="valor">
      <input name="enviar" type="submit" value="OK">
    </form>
  </body>
</html>
```

Após acionar o botão **submit** do primeiro formulário:

Usando o método **GET** a URL fica assim:

```
http://localhost/pagina.php?arg=valor&enviar=OK
```

Usando o método **POST** a URL fica assim:

```
http://localhost/pagina.php
```

- a) **buttons**: cria botões que podem ter suas funcionalidades configuradas
- b) **checkboxes**: cria elementos de seleção, permite selecionar mais de um
- c) **radio buttons**: cria elementos de seleção, permite selecionar apenas um
- d) **menus**: cria lista de itens
- e) **text input**: cria elementos para inserção de texto
- f) **file select**: cria elementos de seleção de arquivos
- g) **hidden controls**: elemento para passagem de dados ocultos no formulário

Controles – input text

```
<INPUT type="TIPO" name="ID" />
```

Tipo	Descrição
text	Inserção de texto
password	Formatação que oculta caracteres para inserção de senha
submit	Cria um botão que ao ser acionado envia as informações do form
radio	Cria estrutura de seleção que permite a escolha de apenas uma opção
checkbox	Cria estrutura de seleção que permite múltipla escolha
image	Cria um botão com imagem
reset	Cria um botão que ao ser acionado limpa os campos do formulário
button	Cria um botão para inserção de ação via linguagem de programação
file	Permite o carregamento de um arquivo
hidden	Elemento para passagem de dados ocultos no formulário

Cria estrutura de seleção que permite múltipla escolha

```
<input name="cb1"  
      type="checkbox"  
      value="1"/> Checkbox 1
```

Cria estrutura de seleção que permite seleção de um elemento dentre várias opções

```
<input name="rd"
      type="radio"
      value="1" /> Radiobutton 1

<input name="rd"
      type="radio"
      value="2" /> Radiobutton 2
```

Atenção: É importante aqui que o nome de todos os radio buttons seja exatamente o mesmo!

Cria uma estrutura que permite a seleção de um item

```
<select name="sel">  
  <option value="1"> Opção 1 </option>  
  <option value="2"> Opção 2 </option>  
  <option value="3"> Opção 3 </option>  
</select>
```

Controles – seleção com agrupamentos

Cria uma estrutura que permite a seleção de um item
O optgroup cria subdivisões entre as opções.

```
<select name="rd">
  <optgroup label="Grupo 1">
    <option value="1"> Opção 1 </option>
    <option value="2"> Opção 2 </option>
  </optgroup>
  <optgroup label="Grupo 2">
    <option value="3"> Opção 3 </option>
  </optgroup>
</select>
```

Cria estrutura de seleção que permite inserção de maior quantidade de textos com .

```
<textarea name="tx" cols="40" rows="5">  
    ...conteúdo...  
</textarea>
```

```
<div>  
    ...conteúdo...  
</div>
```