

FUND_WEB2 - Texto de apoio

Site: [EAD Mackenzie](#)
Tema: FUNDAMENTOS DE WEB {TURMA 01C} 2023/1
Livro: FUND_WEB2 - Texto de apoio

Impresso por: CAIO FRESSATTI PINHEIRO .
Data: domingo, 5 fev 2023, 00:55

Índice

1. TABELAS e FORMULÁRIOS
2. FORMULÁRIO
3. ATRIBUTOS NOVOS DO HTML5
4. Acessibilidade
5. Agrupamento de elementos
6. REFERÊNCIAS BIBLIOGRÁFICAS

1. TABELAS e FORMULÁRIOS

Tabelas

Representam informações em um formato de grade.

Estrutura básica de uma tabela

Perceba que a tag **table** não é utilizada sozinha. Ela necessita de, pelo menos, um ou mais elementos **tr** que, por sua vez, necessitam de, pelo menos, um ou mais elementos **th** ou **td**.

O que significam essas tags?

- **tr** – define uma linha da tabela
- **th** – define uma célula de cabeçalho
- **td** – define uma célula

Além disso, existem mais algumas tags: **thead**, **tfoot** e **tbody**, que ajudam a estruturar a tabela.

O que significam essas tags?

- **thead** – define o cabeçalho da tabela
- **tfoot** – define o rodapé da tabela
- **tbody** – define o corpo da tabela

Atributos de mesclagem para tabelas (**colspan** e **rowspan**) podem ser aplicados nos elementos **td** ou **th**.

- **rowspan** – indica por quantas linhas uma célula deve estender para baixo na tabela
- **colspan** – indica por quantas colunas a célula deve se estender

COLGROUP

- Define um grupo para determinada formatação, por meio da tag **col** e do atributo **span**.

CAPTION E SUMMARY

- É possível adicionar a tag **caption** para termos um título da tabela.
- O atributo **summary** é destinado às tecnologias assistivas, ou seja, à acessibilidade.

Exemplo:

```
<!Doctype html>
<html>
  <head>
    <title> Currículo</title>
    <meta charset="utf-8">
  </head>
  <body>
    <table border="2px">
      <thead>
        <tr>
```

```
<th colspan="4">Tabela de Preços</th>
</tr>
</thead>
<tbody>
<tr>
<td rowspan="2">NOVOS</td>
<td>DELL</td>
<td>Samsung</td>
<td>MacBook</td>
</tr>
<tr>
<td>10000</td>
<td>10000</td>
<td>$1800</td>
</tr>
</tbody>
<tfoot>
<tr>
<td colspan="4">Compre AGORA...</td>
</tr>
</tfoot>
</body></html>
```

Resultado no navegador/browser.

Tabela de Preços			
NOVOS	DELL	Samsung	MacBook
	10000	10000	\$1800
Compre AGORA...			

2. FORMULÁRIO

Formulário é um documento que contém espaços em branco para preencher dados.

Action

Mas afinal, como fazer com que os dados preenchidos pelo formulário fiquem salvos?

O atributo **action** é quem diz para onde os dados de um formulário devem ser enviados. Além disso, devemos informar a maneira como os dados serão enviados, ou seja, se queremos que sejam enviados por meio de uma requisição do tipo GET ou POST (métodos de envio definidos no protocolo HTTP).

```
<form action="http..." method="get/post">
```

Os elementos do formulário recebem algum tipo de entrada do usuário, seja por um clique, seja pelo dígito de algum valor.

- O elemento `<label>` significa uma etiqueta que define um rótulo para muitos elementos do formulário.
- O **for**, atributo do elemento `<label>`, deve ser igual ao id, atributo do elemento `<input>`, descrito a seguir.
- A tag **input** permite que o elemento que a contenha assuma diversas formas, dependendo de seu atributo **type**.
- Alguns atributos type:
 - **text** – entrada de texto
 - **password** – entrada de senha
 - **email** – entrada de email
 - **radio** – botão de opção
 - **checkbox** – caixa de seleção
 - **submit** – botão para submeter
 - **reset** – botão para cancelar
 - **number** cria uma lista de números
 - **range** cria um “slider control”
 - **email** verifica se a entrada corresponde a um endereço de email

SELECT

A tag **select** permite ao usuário escolher um ou mais itens de uma lista. O atributo **multiple**, quando presente, informa ao navegador que mais de um item pode ser selecionado.

A lista de itens deve ser informada por meio de elementos **option**. Tais elementos devem ser filhos diretos ou indiretos de elementos **select**. Além disso, cada item pode conter o atributo chamado **value** para informar o valor associado a uma determinada opção.

OPTGROUP

Caso exista a necessidade de agrupar as opções de um elemento **select**, podemos utilizar o elemento **optgroup** em conjunto com o atributo **label**.

TEXTAREA

A tag **textarea** exibe uma caixa de texto na qual o usuário poderá inserir um texto qualquer.

TIPOS NOVOS DE CAMPOS DO HTML5

O input **email** verifica se a entrada corresponde a um endereço de e-mail.

Usado para preencher campos de e-mail, nos quais o conteúdo digitado é validado automaticamente para o formato de um e-mail.

```
<input type="email">
```

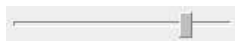


O input **number** cria uma lista de números em um “spinner”.

```
<input type="number">
```

O elemento **range** cria um “slider control”.

```
<input type="range">
```



O input **date** cria um calendário.

```
<input type="date">
```

O input **time** cria “control” com as horas.

```
<input type="time">
```

O elemento **datalist** cria um campo de texto com sugestões.

```
<input list="países">
```

```
<datalist id="países">
```

```
  <option value="Argentina">
```

```
  <option value="Brasil">
```

```
</datalist>
```

3. ATRIBUTOS NOVOS DO HTML5

Placeholder

Usado para deixar um campo com um preenchimento prévio para que o usuário saiba o que digitar no campo como entrada de dados.

```
<input type="text" placeholder="Digite seu nome e sobrenome">
```

Autofocus

Para deixar o cursor sobre determinado campo quando a página carregar, use o autofocus:

```
<input type="text" autofocus="">
```

Required

Para fazer com que um campo seja obrigatório, utilize required.

```
<input type="text" required="">
```

Pattern

pattern – define expressões regulares para validação de formulários.

```
<label for="CPF1">Digite o CPF com máscara: </label>
```

```
<input id="cpf1" type="text" pattern="\d{3}\.d{3}\.d{3}\d{2}" placeholder="__.__.__-__"><br><br>
```

```
<label for="CPF2">Digite o CPF sem máscara: </label>
```

```
<input id="cpf2" type="text" pattern="[0-9]{11}" placeholder="Apenas números"><br>
```

```
<label >Telefone: </label>
```

```
<input type="tel" pattern="^\d2\d{4}[\-s]\d{4}$" placeholder="(XX) XXXX-XXXX"><br>
```

max – define um valor máximo para os diferentes elementos que serão usados:

```
<p>Data Limite:08/10/2013</p>
```

```
<input type="date" max="2013-10-08"><br>
```

```
<p>Valor Máximo: 11</p>
```

```
<input type="number" max=11><br>
```

```
<p>Valor Máximo: 10</p>
```

```
<meter value="3" max=10> 3 de 10<br>
```

min – define um valor mínimo para os diferentes elementos que serão usados:

```
<p>Data Limite:08/10/2013</p>
```

```
<input type="date" min="2013-10-08"><br>
```

<p>Valor Máximo: 11</p>

<input type="number" min=11>

<p>Valor Máximo: 10</p>

<meter value="1" min=0> 3 de 10

4. Acessibilidade

Atualmente, a acessibilidade vem recebendo muita atenção na área da tecnologia. Cada detalhe é aprimorado para que pessoas com necessidades especiais consigam ter uma boa experiência ao usar algum serviço em um computador.

Como deixar os formulários HTML acessíveis?

Dois rótulos (**id** e **for**) com os mesmos valores são utilizados visando à acessibilidade.

id é um atributo que fornece um identificador único e é incrementado no elemento do formulário.

for é um atributo do elemento label.

Labels

São rótulos vinculados a seus respectivos elementos de formulário, como <input>, <select> e <textarea>. Esses elementos devem ter o atributo **id** com o mesmo valor que o atributo **for** do elemento label.

Isso é ótimo, porque não só dá um indicador visual da finalidade de cada campo de formulário na tela, como também explica melhor o significado de cada campo. Além disso, uma pessoa com deficiência visual que estaria usando a página com um leitor de tela agora poderia dizer o que é aquele campo, por meio dos rótulos implementados. Ids também podem ser utilizados com CSS para estilizar um elemento específico.

5. Agrupamento de elementos

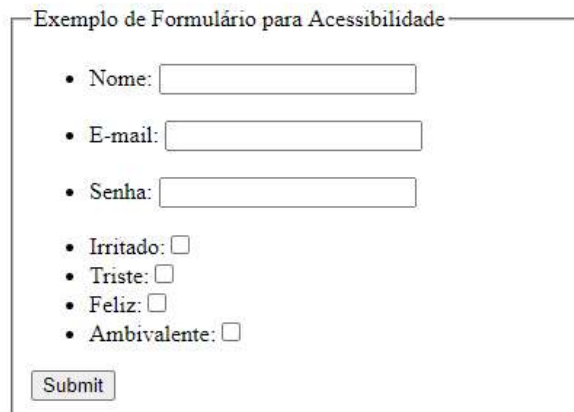
<fieldset>

Forma-se uma linha em torno da borda para mostrar o agrupamento dos elementos inseridos dentro dele.

<legend>

Vem logo depois da tag <fieldset> e exibe uma legenda no papel de identificar o agrupamento feito no formulário.

Um exemplo de formulário para acessibilidade:

A imagem mostra um exemplo de formulário web com o título "Exemplo de Formulário para Acessibilidade". O formulário está contido dentro de uma caixa com uma borda cinza. Dentro da caixa, há uma lista de itens: "Nome:" seguido de um campo de texto; "E-mail:" seguido de um campo de texto; "Senha:" seguido de um campo de senha; e quatro itens de lista com checkboxes: "Irritado:", "Triste:", "Feliz:" e "Ambivalente:". Abaixo da lista, há um botão "Submit".

<html>

<body>

<section>

```
<form method="post" action="" id="contato">
```

```
<fieldset>
```

```
<legend>Exemplo de Formulário para Acessibilidade</legend>
```

```
<ul>
```

```
<li><label for="name">Nome: </label><input type="text" name="name" id="name" value="" ></li><br>
```

```
<li><label for="email">E-mail: </label><input type="email" name="email" id="email" value="" ></li><br>
```

```
<li><label for="pwd">Senha: </label><input type="password" name="pwd" id="pwd" value="" ></li>
```

```
</ul>
```

```
<ul>
```

```
<li><label for="angry">Irritado:</label><input type="checkbox" name="angry" id="angry" value="angry" ></li>
```

```
<li><label for="sad">Triste:</label><input type="checkbox" name="sad" id="sad" value="sad" ></li>
```

```
<li><label for="happy">Feliz:</label><input type="checkbox" name="happy" id="happy" value="happy" ></li>
```

```
<li><label for="ambivalent">Ambivalente:</label><input type="checkbox" name="ambivalent" id="ambivalent" value="ambivalent" ></li>
```

```
</ul>
```

```
<input type="submit" value="Submit">
```

```
</form>
```

```
</fieldset>
```

```
</section>
```

```
</body>
```

</html>

Exemplo de Formulário:

Exemplo de Formulário

Nome:

Endereço:

E-mail:

Site:

Telefone: (xx) xxxx-xxxx

CEP: xxxxx-xxx


First name:

Last name:

E-mail:

Cor:

Criptografia:

Imagem: 

Idade

<html>

<body>

<section>

<header>

<p>Exemplo de Formulário</p>

</header>

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

<label for="nome">Nome: </label><input id="nome" type="text" required name="nome">

<!--

campo obrigatório - required

-->

<label for="end">Endereço: </label><input id="end" type="text" required name="end" autofocus>

<!--

campo obrigatório - required; campo com foco na abertura - autofocus

-->

```
<label for="email">E-mail: </label><input id="email" type="email" required name="email"><br><br>
```

```
<!--
```

campo obrigatório - required;

tipo:email verifica se o e-mail é valido e se o formato está correto, por exemplo: teste@teste.com

```
-->
```

```
<label for="site">Site: </label><input id="email" type="url" required name="site"> <br><br>
```

```
<!--
```

campo obrigatório - required;

tipo:url verifica o formato correto de uma URL, por exemplo: http://www.site.com.br, senão mostra: Insira uma URL.

```
-->
```

```
<label>Telefone: <input required name="tel" pattern="[0-9]2[\s][0-9]{4}-[0-9]{4}"></label> (xx) xxxx-xxxx <br><br>
```

```
<!--
```

campo obrigatório - required;

atributo pattern="dois números, de 0 a 9 entre parênteses, um espaço em branco e mais duas sequências de quatro números de 0 a 9.

```
-->
```

```
<label>CEP: <input required name="cep" pattern="[0-9]{5}-[0-9]{3}"></label> xxxxx-xxx <br><br>
```

```
<!--
```

```
<form action=" " novalidate="novalidate">
```

```
<label>nome: <input name="nome" required></label>
```

```
<label>email: <input name="email" type="email" required></label>
```

```
</form>
```

```
-->
```

```
<!--
```

campo obrigatório - required;

novalidate: quando, dentro de um form, os elem NÃO DEVEM SER VALIDADOS, mesmo que tenham required.

```
-->
```

```
<form action="" autocomplete="on">
```

```
First name: <input type="text" name="fname"><br>
```

```
Last name: <input type="text" name="lname"><br>
```

```
E-mail: <input type="email" name="email" autocomplete="off"><br>
```

```
<input type="submit" >
```

```
</form><br><br>
```

```
<!--
```

autocomplete: habilita o autopreenchimento, somente depois de conteúdo já postado.on é default.

campo e-mail: está com autocomplete=off, portanto não oferece uma lista de opções para preenchimento.

-->

<label>Cor: <input type="color" name="cor"></label>

<label>Criptografia: <keygen name="chave"></label>

<!--

keygen: proporciona uma forma segura de autenticar os usuários de um site por meio de um form. É gerador de par de chaves.

-->

<label>Imagem: <input type="image" src="computador.jpg" width="50" height="40"></label>

<script type="text/javascript">

```
function mostrarIdade(novoValor) {  
    document.getElementById("exibeldade").innerHTML = novoValor;  
}
```

</script>

<form id="app-idade" action="">

<fieldset>

<legend>Idade</legend>

<input id="idade" type="range" min="18" max="120" value="18" onchange="mostrarIdade(this.value)">

18

</fieldset>

</form>

<input type="submit" value="OK">

</section>

</form>

</body></html>

6. REFERÊNCIAS BIBLIOGRÁFICAS

CURSO de HTML5. Disponível em: <<https://www.w3c.br/Cursos/CursoHTML5>>.

DUCKETT, J. *HTML and CSS: design and build websites*. Rio de Janeiro: Alta Books, 2014. ISBN: 9788576089391.

HTML Tutorial. Disponível em: <<https://www.w3schools.com/html/>>.