



HTML, CSS, BOOSTSTRAP.



MATERIAL DE APOIO

Aula 5

HTML, CCS e Bootstrap

Formulários

Professora: Livia Gouvêa

Tutor(a):

Introdução

Formulários HTML são um dos principais pontos de interação entre um usuário e um web site ou aplicativo. Um desenvolvedor web pode criar formulários HTML baseados em um determinado modelo, usando os elementos HTML apropriados, pois são criados usando os elementos de formulário HTML. Estes elementos de formulário HTML são usados para criar diferentes tipos de campos de formulário, tais como caixas de texto, botões, caixas de seleção, etc.

Os formulários HTML são processados pelos navegadores e os dados são enviados para o servidor web. A maioria dos navegadores processa os formulários HTML de acordo com o W3C HTML5 especificação.

Objetivos da aula

- Definir o conceito de formulários HTML?
- Descrever a estrutura básica de um formulário em HTML
- Discutir os principais elementos aplicados a um formulário HTML.

Resumo

O que são formulários? Estrutura básica de um formulário

Formulários são estruturas utilizadas para coletar a entrada de dados do usuário e enviar para o servidor.

A construção de um formulário inicia-se sempre com a *tag* <form> e termina com a *tag* </form>. Esses elementos são um contêiner para diferentes tipos de campos de entrada de dados, tais como: campos de texto, caixas de seleção, botões de opção,

botões de envio, etc. O elemento `<form>` possui alguns atributos referente à forma como o formulário será submetido:

- O atributo *action* define recebe como valor o destino dos dados que serão submetidos no formulário, que, normalmente é uma página em uma linguagem de backend como PHP, Java, ASP, etc. Ex.: `<form action="/action_page.php">`
- O atributo *target* especifica onde exibir a resposta recebida após o envio do formulário, como por exemplo *_blank* (em uma nova janela), *_self* (na janela atual), *_parent* (no quadro pai), *_top* (em todo o corpo da janela) ou *framename* (em um iframe nomeado). Ex.: `<form action="/action_page.php" target="_blank">`
- O atributo *method* especifica o método HTTP a ser usado ao enviar os dados do formulário, que podem ser enviados como variáveis de URL (`method="get"`) ou como transação HTTP post (`method="post"`). Exemplo: `<form action="/action_page.php" method="post">`

Tipos de campos para envio de dados

O elemento HTML `<input>` é o mais comumente usado nos formulários. Um elemento `<input>` pode ser exibido de várias maneiras, dependendo do valor do atributo `type`, por exemplo:

- `<input type="text">` Exibe um campo de entrada de texto de uma linha.
- `<input type="radio">` Exibe um botão para cada item de uma série de opções, onde somente um item pode ser selecionado.
- `<input type="checkbox">` Exibe uma caixa de seleção para cada item de uma série de opções, onde pode-se selecionar mais de um item.
- `<input type="submit">` Exibe um botão enviar (para submeter o formulário)
- `<input type="button">` Exibe um botão clicável
- `<input type="date">` Exibe (ao clicar no campo) um calendário para selecionar uma data.

- `<input type="email">` Exibe um campo para entrada de um endereço de e-mail. O valor de entrada é validado automaticamente para garantir que seja um endereço de e-mail formatado corretamente.

Outro campo muito utilizado é a lista suspensa, que é definida pelo elemento `<select>` seguido das opções da listas, definidas pelos elementos `<option>`

```
<select id="carros" name="carros">
  <option value="gol">Gol</option>
  <option value="palio">Pálio</option>
  <option value="hb20">HB20</option>
  <option value="clio">Clio</option>
</select>
```

Todo campo do formulário deve ter um atributo *name* que deverá receber o nome do campo (sem espaços) para que o servidor possa pegar o valor enviado.

Para atribuir um rótulo ao campo, utiliza-se o elemento `<label>`. É importante a utilização deste elemento para que leitores de tela identifiquem o campo para pessoas com deficiência visual. Este elemento ajuda os usuários que têm dificuldade em clicar em regiões muito pequenas pois ao clicar no texto do `<label>`, ele seleciona o campo, mas, para isso, é necessário utilizar o atributo *for*, que deve receber o mesmo valor do atributo *id* do campo a que se refere, como no exemplo a seguir:

```
<label for="linguagem">Digite aqui a sua linguagem favorita</label>
<input type="text" id="linguagem" name="linguagemFavorita">
```

Abaixo um exemplo de formulário com campos de entrada de texto para nome e sobrenome. Veja que comumente, os valores dos atributos *for*, *id* e *name* são iguais.

```
<form>
  <label for="nome">Nome</label><br>
```

```
<input type="text" id="nome" name="nome"><br>
<label for="sobrenome">Sobrenome:</label><br>
<input type="text" id="sobrenome" name="sobrenome">
</form>
```

Podemos também usar o elemento `<fieldset>` para agrupar campos do formulário, como no exemplo a seguir:

```
<form action="/action_page.php">
  <fieldset>
    <legend>Pessoa:</legend>
    <label for="nome">Nome:</label><br>
    <input type="text" id="nome" name="nome" value="John"><br>
    <label for="sobrenome">Sobrenome:</label><br>
    <input type="text" id="sobrenome" name="sobrenome"
value="Doe"><br>
    <input type="submit" value="Submit">
  </fieldset>
</form>
```

Podemos utilizar o atributo `readonly` para especificar que um campo de entrada é somente leitura. Um campo de entrada somente leitura não pode ser modificado.

Podemos usar também atributo *maxlength* para limitar número máximo de caracteres que podem ser inseridos em um campo.

A seguir mais um exemplo:

```
<form>
  <label for="nome">Nome:</label>
  <input type="text" id="nome" name="nome" value="John" readonly>
  <label for="sobrenome">Sobrenome:</label>
  <input type="text" id="sobrenome" name="sobrenome" value="Doe">
```

```
maxlength>  
</form>
```

Como aplicar na prática o que aprendeu

- Vamos incluir na página web mais uma seção para conter um formulário. Para isso, abra a seção com `<section>`
- Dentro dela, inclua a abertura do formulário `<form action="" method="get">`
- Vamos incluir um campo para o usuário digitar seu nome. Para isso, inclua o rótulo `<label for="nome">Nome:</label>
` e em seguida, o campo `<input type="text" id="nome" name="nome" value="John" readonly>

`
- Agora vamos incluir um campo para o usuário digitar seu sobrenome. Para isso, vamos incluir o rótulo `<label for="sobrenome">Sobrenome:</label>
` e o campo `<input type="text" id="sobrenome" name="sobrenome" value="Doe" maxlength>

`
- Vamos criar mais um campo do tipo lista suspensa. Vamos adicionar um rótulo para a lista `<label for="linguagem">Escolha sua linguagem preferida:</label>
`
- E vamos abrir a lista `<select id="linguagem" name="linguagem">`
- e em seguida incluir os itens:
`<option value="html">HTML</option>`
`<option value="css">CSS</option>`
`<option value="javascript">Javascript</option>`
- Vamos fechar a lista suspensa `</select>

`
- Agora vamos incluir um botão para enviar os dados `<input type="submit" value="Enviar">`
- Vamos fechar o formulário `</form>` e em seguida, fechar a seção `</section>`
- Agora volte ao navegador e atualize a página para visualizar as alterações

Dica quente para você não esquecer

O elemento `<label>`:

- Define rótulos para os campos do formulário.
- É útil para usuários que necessitam de leitores de tela, porque o leitor lerá em voz alta o rótulo quando o usuário focar no campo dele.
- Ajuda os usuários que têm dificuldade em clicar em regiões muito pequenas (como botões de opção ou caixas de seleção) - pois quando o usuário clica no `<label>`, ele seleciona o campo deste rótulo
- O atributo *for* do `<label>` deve ser igual ao atributo *id* do campo a que ele se refere.

O atributo `method` do formulário

Observações sobre o método GET:

- Anexa os dados do formulário à URL, em pares nome/valor
- NUNCA use GET para enviar dados confidenciais! (os dados do formulário enviado são visíveis na URL!)
- O comprimento de um URL é limitado (2048 caracteres)
- Este método é útil para envios de formulários em que um usuário deseja marcar o resultado
- GET é bom para dados não seguros, como strings de consulta no Google

Observações sobre o POST:

- Anexa os dados do formulário dentro do corpo da solicitação HTTP (os dados do formulário enviado não são mostrados na URL)
- O POST não tem limitações de tamanho e pode ser usado para enviar grandes quantidades de dados.

Referência Bibliográfica

MDN Web Docs. [S. l.], 20 set. 2022. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Learn/Forms> Acesso em: 20 set. 2022.

Exercícios

1. Qual é o HTML correto para fazer uma caixa de seleção?

- a) `<input type="check">`
- b) `<checkbox>`
- c) `<check>`
- d) `<input type="checkbox">`
- e) `<check-input>`

Gabarito comentado: Letra D. Os elementos `<input>` do tipo *checkbox* são renderizados por padrão como caixas de seleção. Uma caixa de seleção permite selecionar mais de um item no grupo.

2. Em HTML, qual atributo é usado para especificar o que deve ser preenchido em um campo?

- a) `placeholder`
- b) `validate`
- c) `required`
- d) `formvalidate`
- e) `Value`

Gabarito comentado: Letra C. O atributo *required*, quando presente, indica que o campo do formulário deve ser obrigatoriamente preenchido. Esse atributo pode ser utilizado nos campos dos tipos *text*, *search*, *url*, *tel*, *email*, *password*, *date*, *month*, *week*, *time*, *datetime-local*, *number*, *checkbox*, *radio* e *file*.

3. Qual o elemento do HTML5 é usado para agrupar elementos dentro de um formulário?

- a) <group>
- b) <div>
- c) <section>
- d) <form>
- e) <fieldset>

Gabarito comentado: Letra E. O elemento *<fieldset>*, do HTML5 é usado para agrupar vários campos, bem como rótulos (*<label>*) dentro de um formulário web.

4. Qual atributo do <form> define o destino dos dados?

- a) method
- b) action
- c) POST
- d) GET
- e) Target

Gabarito comentado: Letra B. O atributo *action* especifica o destino de envio dos dados do formulário quando um formulário é submetido e tem 2 valores possíveis:

- Uma URL absoluta - aponta para outro site (como `action="http://www.example.com/example.htm"`)
- Uma URL relativa - aponta para um arquivo dentro de um site (como `action="example.htm"`).