



**INSTITUTO FEDERAL**

Catarinense

Campus Camboriú

# Desenvolvimento Web I

**Professor: Euclides Paim**

*[euclides.paim@ifc.edu.br](mailto:euclides.paim@ifc.edu.br)*



**INSTITUTO FEDERAL**

Catarinense

Campus Camboriú

# Formulários HTML

**Professor: Euclides Paim**

*[euclides.paim@ifc.edu.br](mailto:euclides.paim@ifc.edu.br)*

# Formulários HTML

## Elemento <form>

Um elemento HTML **<form>** define uma interface de entrada utilizada para interagir e recolher dados do usuário. Um formulário HTML contém **elementos de formulário**. Elementos de formulário são diferentes tipos de elementos de entrada, como: campos de texto, caixas de seleção, botões de rádio, botões de envio, etc...

Sintaxe:

<form>

•

*form elements*

•

</form>

# Formulários HTML

## O elemento `<input>`

O elemento `<input>` é o elemento mais importante de um formulário. Este elemento é exibido de várias maneiras, dependendo do **tipo de** atributo.

Tipo	Descrição
<code>&lt;Input type = "text"&gt;</code>	Define um campo de entrada de texto de linha única
<code>&lt;Input type = "radio"&gt;</code>	Define um botão de rádio (para a seleção de uma das muitas escolhas)
<code>&lt;Input type = "submit"&gt;</code>	Define um botão de envio (para enviar o formulário)



## Formulários HTML

Campos de texto `<input type="text">`

O elemento do tipo **"text"** `<input type="text">` define um campo de entrada de linha única para **entrada de texto**.

```
<form>
```

```
  <label for="fname">First name:</label><br>
```

```
  <input type="text" id="fname" name="fname"><br>
```

```
  <label for="lname">Last name:</label><br>
```

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

```
</form>
```

**Nota:** O `<form>` em si não é visível. Observe também que a largura padrão de um campo de entrada é de 20 caracteres.

## Formulários HTML

### O elemento <label>

Observe a utilização do elemento **<label>** do exemplo acima. Esta tag define um rótulo para muitos elementos do formulário. Este elemento é útil para usuários de leitores de tela, porque o leitor de tela irá ler o valor do rótulo quando o usuário estiver focado no elemento <input>.

O elemento **<label>** também auxilia aos usuários que têm dificuldade clicando em regiões muito pequenas (como botões de rádio ou caixas) - porque quando o usuário clica no texto dentro do elemento <label>, seleciona o botão de rádio/caixa.

O atributo **for** da tag <label> deve ser igual ao atributo **id** do elemento <input> para vincula-los.



# Formulários HTML

## Botões de rádio - Radio buttons

O elemento do tipo **"radio"** `<input type="radio">` define um **botão de rádio**. *Radio buttons* permitem que o usuário selecione uma opção de uma lista limitada de opções. No exemplo a seguir vemos um `<form>` com *radio buttons*.

```
<form>
```

```
  <input type="radio" id="homem" name="genero" value="homem">
```

```
  <label for="homem">Masculino</label><br>
```

```
  <input type="radio" id="mulher" name="genero" value="mulher">
```

```
  <label for="mulher">Feminino</label><br>
```

```
  <input type="radio" id="outros" name="genero" value="outros">
```

```
  <label for="other">Outros</label>
```

```
</form>
```



## Formulários HTML

### Botão enviar - *submit*

O elemento do tipo **"submit"** `<input type="submit">` define um botão para enviar os dados do formulário para o servidor. A forma de manipular os dados é normalmente através de uma página no servidor com um *script* para processar os dados de entrada. A maneira como esses dados serão manipulados pelo servidor é especificada no atributo **action** do `<form>`. Veja o exemplo:

```
<form action="/action_page.php">  
  <label for="fname">First name:</label><br>  
  <input type="text" id="fname" name="fname" value="John"><br>  
  <label for="lname">Last name:</label><br>  
  <input type="text" id="lname" name="lname" value="Doe"><br><br>  
  <input type="submit" value="Submit">  
</form>
```





## Formulários HTML

### O atributo *action*

O atributo ***action*** define a ação a ser executada quando o formulário é enviado. Normalmente, os dados do formulário são enviados para uma página no servidor quando o usuário clica no botão enviar.

No exemplo anterior, os dados do formulário são enviados para uma página no servidor chamada `"/action_page.php"`. Esta página contém um script do lado do servidor que lida com os dados do formulário:

```
<form action="/action_page.php">
```



# Formulários HTML

## O atributo *target*

Os atributos ***target*** especificam se os resultados apresentados serem abertos em uma nova aba do navegador ou na janela atual. O valor padrão é " `_self`", que significa o formulário será enviado para a janela atual. Para abrir o resultado do formulário em uma nova aba do navegador, use o valor " `_blank`". No exemplo a seguir o resultado será apresentado em uma nova aba:

```
<form action="/action_page.php" target="_blank">
```

# Formulários HTML

## O atributo *method*

O atributo ***method*** especifica o método HTTP ( **GET** ou **POST** ) para ser utilizado quando os dados do <form> forem enviados. Exemplo usando o método GET ao enviar o formulário:

```
<form action="/action_page.php" method="get">
```

Exemplo usando o método POST ao enviar o formulário:

```
<form action="/action_page.php" method="post">
```

# Formulários HTML

## Quando usar GET?

O método padrão no HTTP para enviar dados por formulários é o GET. Contudo, quando usamos GET, os dados do formulário serão visíveis no campo de endereço da página:

/action\_page.php?**firstname=John&lastname=Doe**

### Notas sobre o GET:

- Anexa dados de formulário na URL em pares nome/valor
- O tamanho da URL é limitado (2048 caracteres)
- Nunca use GET para enviar informações privadas (*sensitive data*)!
- Útil para enviar dados quando o usuário quer salvar os resultados
- GET é melhor para dados não sensíveis, como *strings* de consultas no Google



# Formulários HTML

## Quando usar POST?

Sempre use POST se os dados do formulário contiverem informações confidenciais ou pessoais (*sensitive data*). O método POST não exibe os dados do formulário no campo de endereço de página.

### **Notas sobre POST:**

- POST não tem limitações de tamanho, e pode ser usado para enviar grandes quantidades de dados.
- envios de formulários com POST não podem ser marcados



## Formulários HTML

### O atributo de *name*

Cada campo de entrada deve ter um atributo ***name*** para ser enviado. Se o atributo *name* for omitido, os dados desse campo de entrada não serão enviados completamente. Este exemplo não enviará o valor digitado no campo "*Primeiro nome*" :

```
<form action="/action_page.php">  
  <label for="fname">Primeiro nome:</label><br>  
  <input type="text" id="fname" value="Joao"><br><br>  
  <input type="submit" value="Enviar">  
</form>
```

Obs.: Ao clicar no botão "*enviar*", os dados deveriam ser enviados para a página "*/action\_page.php*", contudo os dados não serão processados devido à ausência do atributo *name* no elemento *input* .



# Formulários HTML

## O elemento *select*

O elemento `<select>` declara uma lista ***drop-down***.

```
<select id="cars" name="cars">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

Os elementos `<option>` definem as opções que podem ser selecionadas. Por padrão, o primeiro item da lista *drop-down* é selecionado. Para pré-definir uma opção, adicione o atributo *selected* ao *option*:

```
<option value="fiat" selected>Fiat</option>
```



# Formulários HTML

## O elemento *select*

Use o atributo *size* para especificar o número de valores visíveis:

```
<select name="cars" size="3">  
  <option value="volvo">Volvo</option>  
  <option value="saab">Saab</option>  
  <option value="fiat">Fiat</option>  
  <option value="audi">Audi</option>  
</select>
```

Use o atributo *multiple* para permitir que o usuário selecione mais de um valor:

```
<select name="cars" size="4" multiple>  
  <option value="volvo">Volvo</option>  
  <option value="saab">Saab</option>  
  <option value="fiat">Fiat</option>  
  <option value="audi">Audi</option>  
</select>
```



# Formulários HTML

## O elemento *textarea*

O elemento `<textarea>` define um campo de entrada multi-line (**área de texto**):

```
<textarea name="message" rows="10" cols="30">  
The cat was playing in the garden.  
</textarea>
```

- O atributo *rows* especifica o número visível de linhas em uma área de texto.
- O atributo *cols* especifica a largura visível de uma área de texto.

Obs.: Você também pode definir o tamanho da área de texto usando CSS:

```
<textarea name="message" style="width:200px; height:600px;">  
The cat was playing in the garden.  
</textarea>
```



# Formulários HTML

## O elemento *button*

O elemento `<button>` define um **botão** "clicável":

```
<button type="button" onclick="alert('Hello World!')">Click Me!</button>
```

**Nota:** Sempre especifique o atributo ***type*** para o elemento *button*. Diferentes navegadores podem adotar padrões diferentes para este elemento.

O elemento `<datalist>` especifica uma lista de opções pré-definidas para um elemento `<input>`. Os usuários verão uma lista *drop-down* de opções de dados pré-definidas a serem selecionadas. O atributo *list* do elemento `<input>`, deve se referir ao atributo *id* do elemento `<datalist>`.

```
<form action="/action_page.php">
  <input list="browsers">
    <datalist id="browsers">
      <option value="Internet Explorer">
      <option value="Firefox">
      <option value="Chrome">
      <option value="Opera">
      <option value="Safari">
    </datalist>
  </form>
```

# Formulários HTML

## O elemento *output*

O elemento `<output>` apresenta o resultado de um cálculo (realizado por um *script*). O exemplo a seguir executa um cálculo e mostra o resultado em um elemento `<output>`:

```
<form action="/action_page.php"
  oninput="x.value=parseInt(a.value)+parseInt(b.value)">
  0
  <input type="range" id="a" name="a" value="50">
  100 +
  <input type="number" id="b" name="b" value="50">
  =
  <output name="x" for="a b"></output>
  <br><br>
  <input type="submit">
</form>
```

# Formulários HTML

## Exercícios de Fixação

Exercícios...

# Referências

## Referências Básicas

Livro NIEDERAUER, Juliano. Desenvolvendo websites com PHP: aprenda a criar websites dinâmicos e interativos com PHP e bancos de dados. 2. ed. rev. e atual. São Paulo: Novatec, 2011.

Livro SILVA, Maurício Samy. HTML 5: a linguagem de marcação que revolucionou a web. São Paulo: Novatec, 2011.

Livro SILVA, Maurício Samy. CSS3: desenvolva aplicações web profissionais com uso dos poderosos recursos de estilização das CSS3. São Paulo: Novatec, 2012.

## Referências Complementares

Livro DEITEL, Paul J. Ajax,. Rich Internet applications e desenvolvimento Web para programadores. . Pearson Prentice Hall. 2009

Livro DALL'OGGIO, Pablo. PHP: programando com orientação a objetos. . Novatec. 2009

Livro SOARES, Wallace. PHP 5: conceitos, programação e integração com banco de dados. . Érica. 2010

Livro SILVA, Maurício Samy. Criando sites com HTML: sites de alta qualidade com HTML e CSS. . Novatec. 2010

Livro FLANAGAN, David. o guia definitivo. . O'Reilly. 2012

## Referências na Internet

<https://www.w3schools.com>

<https://developer.mozilla.org/pt-BR/docs/Web>