

O uso de Formulários é a forma mais simples, e a mais comum, para a prestação de serviços interativos. Também costuma ser o primeiro obstáculo sério na implementação da acessibilidade. Se acessar conteúdos informacionais de um sítio é uma atitude passiva do visitante, o acesso de um formulário exige que o usuário interaja com este.

Formulários costumam ser a principal dificuldade de desenvolvimento na formatação do código HTML, pois são ao mesmo tempo formatação e programação. Os usuários devem ter acesso às suas informações e poder interagir com elas, fornecendo as informações pertinentes para que a execução do serviço tenha sucesso. Para tanto o formulário deve ser compreensível, não só visualmente, mas também na sua formatação.

### **1. A estrutura correta do formulário**

---

O primeiro passo para tornar o formulário acessível é organizá-lo de forma a ser compreensível, tanto na sua formatação quanto visualmente. Lembrando que o formulário deve ser acessível não só pelos programas leitores de tela, mas pelas pessoas que utilizam navegadores gráficos.

#### **Os elementos:**

Os elementos básicos para a construção de um formulário são:

- FORM: o contêiner do formulário;
- INPUT: que especifica vários tipos de campos como text, submit, button, radio button e checkbox;
- TEXTAREA: campo de texto multi-linha, como observações e comentários;
- SELECT: Utilizado para listas selecionáveis (como um menu);
- BUTTON: Utilizado para botões.

Dado esses elementos básicos pode-se montar um formulário simples e organizado. O que muitos desenvolvedores não sabem é que os navegadores e leitores de texto possuem uma *ordem de leitura* desses elementos e de seus rótulos. Por exemplo, ao encontrar um campo `<input type="text"...`, os leitores de tela esperam que o rótulo esteja *antes* do campo, mas ao encontrar um `<input type="checkbox" ...`, o leitor espera encontrar o rótulo depois do campo.

Para facilitar a construção dos formulários, prevendo sua ordem de leitura, existem regras de usabilidade visando facilitar a sua interpretação.

Estrutura de elementos de formulário - Boas práticas em IHC (interação homem-computador):

▪ **input type="text"**

- Estrutura: Rótulo - Elemento
- Exemplo:

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

▪ **input type="password"**

- Estrutura: Rótulo - Elemento
- Exemplo:

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

▪ **input type="button"**

- Estrutura: Não se aplica
- Exemplo:

```
<input type="button" name="cmdSend" value="Envia" />
```

▪ **input type="submit"**

- Estrutura: Não se aplica
- Exemplo:

```
<input type="submit" name="cmdSend" value="Envia" />
```

▪ **input type="radio"**

- Estrutura: Elemento - Rótulo
- Exemplo:

```
<input type="radio" name="radsim" value="Yes" />Sim<br />
```

```
<input type="radio" name="radnao" value="No" />Não<br />
```

▪ **input type="checkbox"**

- Estrutura: Elemento - Rótulo
- Exemplo:

```
< input type="checkbox" name="chkSubscribe" value="Subscribe" /> Assinar Boletim
```

- **textarea**
  - Estrutura: Rótulo - Elemento
  - Exemplo:

```
Observações: <textarea name="txtObservacoes"></textarea>
```

- **select**
  - Estrutura: Rótulo - Elemento
  - Exemplo:

```
Estado: <select name="ddlestado">  
  
<option>AL - Alagoas</option>  
  
...
```

- **button**
  - Estrutura: Não se aplica
  - Exemplo:

```
<button name="cmdSalvar">Salvar</button>
```

### **Tabelas para diagramar formulários**

Apesar de desnecessário, tabelas ainda são utilizadas para a diagramação de formulários. Esta prática pode ocasionar problemas caso o elemento e o rótulo estejam localizados em células diferentes. Os leitores de tela ignoram a disposição visual sendo o conteúdo lido na ordem em que foi escrito no código. Portanto, se possível, mantenha o rótulo e o elemento na mesma célula, ou na célula seguinte de acordo com o código.

## **2. O formulário acessível**

---

A criação de uma estrutura que respeite as boas práticas de IHC é o primeiro passo para a criação de formulários acessíveis. O próximo passo é a adição de elementos que garantam a acessibilidade dos formulários. Esses elementos são: LABEL, FIELDSET, LEGEND e OPTGROUP

## O elemento LABEL

O elemento LABEL é utilizado, em conjunto com o atributo "id" para relacionar de forma clara o elemento ao seu rótulo.

O elemento LABEL pode ser usado em conjunto com todos os elementos de formulário, exceto o elemento BUTTON.

Ex:

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

## Os elementos FIELDSET e LEGEND

O elemento FIELDSET é utilizado para agrupar os itens de formulários com características em comum. Por exemplo, um formulário para inscrição em boletim pode ter seus elementos agrupados da seguinte forma:

- Dados de identificação: Nome, sobrenome, endereço físico, endereço de email;
- Preferências de recebimento: periodicidade (diária, semanal, mensal) e tipo de boletim (html, texto plano);
- Artigos de seu interesse: Lista de itens que podem constar no boletim.

O elemento FIELDSET é utilizado em conjunto com o elemento LEGEND, que melhora a acessibilidade do elemento FIELDSET quando este não é visível em navegadores gráficos.

Ex:

```
<fieldset>  
<legend>Quais suas cores preferidas?</legend>  
<input type="checkbox" name="chk0" id="chk0_0" value="vermelho">  
<label for="chk0_0">vermelho</label><br />  
<input type="checkbox" name="chk1" id="chk1_1" value="verde">  
<label for="chk1_1">verde</label><br />  
<input type="checkbox" name="chk2" id="chk2_2" value=" azul">  
<label for="chk2_2">azul</label><br />  
<input type="checkbox" name="chk3" id="chk3_3" value="amarelo">  
<label for="chk3_3">amarelo</label>  
</fieldset>
```

## O elemento OPTGROUP

O elemento OPTGROUP trabalha de forma semelhante ao FIELDSET, agrupando itens do elemento SELECT quando necessário. Cada elemento OPTGROUP recebe um atributo "label" que é o nome do grupo criado.

Ex:

```
<select name="Lista de mercado">  
  <optgroup label="Laticínios">  
    <option>leite desnatado</option>  
    <option>queijo ralado</option>  
    <option>queijo prato</option>  
    <option>requeijão</option>  
  </optgroup>  
  <optgroup label="Vegetais">  
    <option>alface</option>  
    <option>alho</option>  
    <option>Beringela</option>  
    <option>Cenoura</option>  
    <option>Pimentão</option>  
  </optgroup>  
  <optgroup label="Frutas">  
    <option>Abacaxi</option>  
    <option>Banana</option>  
    <option>Manga</option>  
  </optgroup>  
</select>
```

## Teclas de atalho (Accesskeys)

O uso de teclas de atalho através do atributo "accesskey", permite escolher um atalho de teclado para cada um dos campos do formulário, possibilitando o acesso rápido. O atributo pode ser aplicado tanto ao elemento LABEL como ao elemento INPUT.

Ex: tecla de atalho para a busca

```
<input type="text" name="txtbusca" accesskey="s" />
```

No exemplo, se o usuário pressionar as teclas "alt" e "S" ele irá diretamente para o campo de busca.

No entanto, deve-se tomar cuidado com a escolha das teclas de atalho, pois diversos navegadores utilizam-se delas para sua própria navegação, variando do navegador, versão e língua. Em caso de conflito, a preferência é do navegador.

### 3. O uso de scripts

---

Um formulário deve funcionar *independentemente dos scripts* estarem habilitados no navegador ou no leitor de tela.

Valide sempre a manipulação de dados no lado do servidor. Isso possibilita o funcionamento de formulários independentemente de scripts no lado do usuário.

**Prefira:** Validação pelo servidor

```
<input type="submit" value="Comprar" />
```

**Do que:** Validação por script

```
<input type="button" value=" Comprar"  
onclick="document.forms('main').submit();" />
```

#### Menu Javascript (menus drop-down)

Menus estilo drop-down, que utilizam o elemento SELECT com javascript, costumam ser inacessíveis a usuários que utilizam leitores de tela ou apenas o teclado para navegar.

Nesses casos é preferível usar a lista (elemento SELECT) junto a um botão de ação, que permita o usuário ativar sua seleção.

### 4. Mensagens de erro amigáveis

---

Muitas vezes o usuário desiste de preencher um formulário por não conseguir entender as mensagens de erro. Salvo algumas exceções, a culpa é, em geral, do mau uso e formatação das mensagens de erro em formulários.

Existem vários scripts do lado do usuário que detectam e tratam erros de preenchimento, mas, como visto anteriormente, o uso de scripts do lado do usuário é muitas vezes inacessível e não recomendado.

### **Mensagens de erros no lado do SERVIDOR**

Uma vez enviado o formulário e encontrado erros de preenchimento, a forma usual é apresentar no topo da página um sumário dos erros encontrados, destacando os locais onde encontrados os erros. Esse método costuma ser eficaz visualmente, mas costuma causar confusão nos leitores de tela, pois geralmente o nome da página de retorno é o mesmo. O problema se resolve colocando um alerta no elemento TITLE da página, repetindo o mesmo alerta no cabeçalho de conteúdo da página.

Ex:

```
<title> Alguns dados estão incorretos ou faltando no formulário. Por favor, corrija. </title>
```

Outra forma, mais amigável é capturar os dados preenchidos corretamente e retornar ao usuário *somente* os campos com erros para correção.

## **5. Leitura recomendada**

---

### **Accessible HTML/XHTML Forms: Beginner Level**

<http://www.webstandards.org/learn/tutorials/accessible-forms/beginner/>

### **Construindo formulários HTML/XHTML acessíveis - 1ª Parte**

*(tradução do link anterior)*

<http://www.maujor.com/tutorial/formac-a.php>

### **HTML 4.01 Specification - Forms**

<http://www.w3.org/TR/html4/interact/forms.html>