

# Guia visual de controles de formulário HTML

Márcio d'Ávila, 17 de dezembro de 2003.

Categoria: [Internet: Servidor: Formulários](#)

---

## Controle bem-sucedido

---

Um controle de formulário bem-sucedido é aquele válido para envio (submissão). Todo controle bem-sucedido deve estar **dentro** de um `form` e ter um **nome** (`name`) pareado com um valor atual, ambos enviados no conjunto de dados do formulário. Em outras palavras, um controle bem-sucedido é aquele que gera um par nome/valor nos dados do formulário.

- Um formulário pode ter vários botões `submit`, mas somente o que ativar o envio pode ser bem-sucedido. Já um botão `reset` nunca deve ser considerado pelo browser como bem-sucedido.
- Todos os botões de escolha (`checkbox`) "ligados" podem ser bem-sucedidos.
- Para botões de alternativa (`radio`) com o mesmo nome, somente o "ligado" pode ser bem-sucedido.
- Para lista de seleção (`select`), somente as opções selecionadas podem ser bem-sucedidas. Quando nenhuma opção é selecionada, o controle não é bem-sucedido e nem o nome nem seus valores são enviados quando o formulário é submetido.
- Controles ocultos (`hidden`) e controles não visíveis por configurações de estilo/CSS (`display: none;`) ainda assim podem ser bem-sucedidos.

## Formulário com exemplo de cada tipo de controle

---

O formulário a seguir exemplifica o código e a visualização dos elementos (tags) e atributos usados para a criação dos diversos tipos de controle de formulário HTML. Para mais informações, leia a especificação HTML 4.01, listada como referência.

---

### input type="text"

```
<label for="ptexto1" title="T&iacute;tulo do label de ptexto1">
  Input <u>T</u>exto:</label>
<input type="text" name="ptexto1" id="ptexto1" value="texto"
  size="10" maxlength="10" accesskey="T" />
```

Input Texto:

Este controle `input text` possui um elemento **rótulo** (`label`) associado a ele. Quando clica-se no rótulo, o controle associado recebe o foco. O elemento rótulo é utilizado com os controles `input`, exceto o do tipo `button` que já é rotulado implicitamente pelo seu atributo `value`. O `label` é associado pelo seu atributo `for`, que deve referenciar o identificador (atributo `id`) do controle `input` correspondente.

Todo controle, assim como rótulo e legenda, pode ter um **título** com o atributo `title` que define um texto explicativo complementar, normalmente exibido pelos browsers como uma *tooltip* (caixa de texto sob o mouse). O rótulo do texto acima e o controle de texto somente-leitura abaixo possuem `title`; passe o mouse sobre eles para ver o resultado.

```
<input type="text" name="ptexto2" readonly="readonly" value="somente leitura"
title="Título do ptexto2" />
```

Input texto:

Controles com o atributo `readonly` recebem foco, são incluídos na navegação por TAB e podem ser bem-sucedidos, mas não podem ser modificados pelo usuário.

```
<input type="text" name="ptexto3" disabled="disabled" value="desabilitado" />
```

Input texto:

Controles com o atributo `disabled` não recebem foco, são ignorados na navegação por TAB e não podem ser bem-sucedidos.

## input type="password"

```
<input type="password" name="psenha" id="psenha" value="senha" accesskey="S" />
```

Input Senha:

## input type="hidden"

```
<input type="hidden" name="poculto1" value="escondido 1" />
<input type="hidden" name="poculto2" value="escondido 2" />
```

Um campo oculto (`hidden`) serve para incluir nos dados do formulário um par nome/valor fixo, sem exibi-lo ao usuário na página (embora fique legível no código fonte do HTML).

## input type="file"

```
<input type="file" name="parquivo" id="parquivo" />
```

Nome de arquivo:  Nenhum arquivo selecionado.

O campo de entrada de arquivo permite ao usuário especificar um arquivo local cujo conteúdo será enviado junto com o formulário submetido, como *upload*.

Para isso, o elemento `form` do formulário que o conteúdo deve especificar os atributos `method="POST"` e `enctype="multipart/form-data"`. O elemento `form` pode opcionalmente especificar também o atributo `accept="..."` com uma lista (itens separados por vírgula) de [tipos de conteúdo MIME](#) aceitos pelo servidor para o upload de arquivos.

Devido a implicações em segurança, os navegadores web em geral não consideram um nome de arquivo pré-definido pelo atributo `value`, [nem permitem definir o valor programaticamente](#) via JavaScript. O arquivo só pode ser especificado interativamente pelo usuário.

## input type="checkbox"

```
<input type="checkbox" name="pcheck" id="pcheck1" value="1" />>
<label for="pcheck1">Um</label><br />
<input type="checkbox" name="pcheck" id="pcheck2" value="2" />
<label for="pcheck2">Dois</label><br />
```

```
<input type="checkbox" name="pcheck" id="pcheck3" value="3"
  disabled="disabled" />
<label for="pcheck3">Tr&ecirc;s</label>
```

- ☐ Um
- ☐ Dois
- ☐ Três

Nada impede de haver vários controles com o mesmo nome. Neste caso, simplesmente são enviados nos dados do formulário múltiplos pares de nome/valor para um mesmo nome. Para identificar unicamente cada controle com mesmo nome na página, visando referenciá-lo em um label, script ou estilo, deve ser usado o atributo `id`, este sim único para cada elemento HTML na página.

```
<input type="checkbox" name="pcheck4" id="pcheck4" checked="checked" value="x" />
```

- ☒ Outro

Controles `input checkbox` e `radio` com o atributo `checked` ficam inicialmente "ligados" (marcados/ativos).

## select & option

```
<select name="lista1" size="3">
  <option value="1" selected="selected">Um</option>
  <option value="2">Dois</option>
  <option value="3">Tr&ecirc;s</option>
  <option value="4">Quatro</option>
</select>
```

```
Um
Dois
Três
```

Quem define o nome (`name`) do controle é o elemento `select`, enquanto cada valor (`value`) é dado por um `option`. O rótulo de cada opção é dado pelo texto entre as tags de abre e fecha `option`.

O atributo `size` define o número de opções (linhas) exibidas da lista. A exibição da barra de rolagem é automática e fica a cargo do browser.

O atributo `selected` define uma opção pré-selecionada (inicialmente "ligada"). É recomendado sempre definir explicitamente o que estará pré-selecionado, para evitar ambigüidade de comportamento em diferentes browsers.

```
<select name="lista2" size="1">
  <option value="">-- escolha --</option>
  <option value="1">Um</option>
  ...
</select>
```

```
-- escolha --
```

Uma combobox é gerada quando o atributo `size` é 1 (padrão quando a lista não é `multiple`).

A maioria dos navegadores posiciona a seleção inicialmente na primeira opção, quando não é especificada nenhuma opção com o atributo `selected`. Quando é utilizada uma lista estilo combobox em um formulário e deseja-se que nenhuma opção esteja efetivamente pré-selecionada inicialmente, é usual criar uma primeira opção com valor em branco (`value=""`), com o texto de rótulo vazio ou um indicativo como por exemplo "-- escolha --". Note que definir um valor em branco é diferente de omitir o atributo `value`, pois neste último caso o rótulo (texto) da opção é assumido como seu valor.

```
<select name="lista2" multiple="multiple">
  ...
```

```
</select>
```

Um  
Dois  
Três  
Quatro

A lista com o atributo `multiple` pode ter mais de uma opção selecionada, normalmente pressionando-se **CONTROL** e/ou **SHIFT** ao selecionar outro elemento.

Obviamente, somente em uma lista múltipla é válido ter mais de uma opção pré-selecionada.

```
<select name="lista4" size="5">
  <option selected label="Sem grupo" value="0">Sem grupo</option>
  <optgroup label="Grupo 1">
    <option label="1.1" value="1.1">Op&ccedil;&atilde;o 1.1</option>
    <option label="1.2" value="1.2">Op&ccedil;&atilde;o 1.2</option>
  </optgroup>
  ...
</select>
```

Sem grupo  
**Grupo 1**  
  Opção 1.1  
  Opção 1.2  
**Grupo 2**

O elemento `optgroup` serve para criar grupos de opções, dentro de uma lista `select`, formando listas hierárquicas. O texto exibido pelo `optgroup` é apenas um rótulo (gerado pelo atributo `label`) e não pode ser selecionado, ou seja, não é uma opção válida da lista.

## input type="radio" & fieldset

```
<fieldset style="width: 50%;">
<legend title="Informa&ccedil;&atilde;o complementar da legenda">
  Legenda do Fieldset</legend>
<input type="radio" name="pradio" id="pradio1" value="1" checked="checked" />
<label for="pradio1">Um</label><br />
<input type="radio" name="pradio" id="pradio2" value="2" />
<label for="pradio2">Dois</label><br />
<input type="radio" name="pradio" id="pradio3" value="3" disabled="disabled" />
<label for="pradio3">Tr&ecirc;s</label><br />
</fieldset>
```

Legenda do Fieldset

☒ Um  
☐ Dois  
☐ Três  
☐ Quatro

Um grupo de botões de alternativa (`radio`) é definido colocando-se todas as opções com o mesmo nome do controle (`name`).

**Bug:** No Internet Explorer testado (6 SP1 Windows), a navegação entre as alternativas do mesmo grupo com o teclado é feita com as setas, e não com a tecla **TAB**, refletindo o comportamento usual deste tipo de controle no Windows. Porém, quando um dos controles `radio` de um grupo está desabilitado, a navegação com as teclas de seta não vai além do controle desabilitado, tornando as alternativas do grupo seguintes ao controle desabilitado inacessíveis quando usado apenas o teclado. Este problema pôde ser observado no exemplo acima.

## textarea

```
<textarea name="parea" rows="5" cols="60" wrap="off">
```

Eis o conteúdo da área de texto.

O controle `textarea` é um campo de texto multi-linha. Os atributos `rows` e `cols` definem o número de linhas de altura e o de "caracteres" de largura, respectivamente. O atributo `wrap` define se a mudança de linha será automática durante a digitação ("on"), ou somente quando for pressionado ENTER ("off"). Este exemplo tem `wrap="off"`.

Um elemento `textarea` pode ter atributo `readonly` ou `disabled`.

## input type="image"

```
<input type="image" name="pimagem" id="pimagem"
  src="img/icone.gif" alt="I estilizado" value="valor" />
<label for="pimagem">Imagem</label>
```



Imagem

Um `input image` funciona como um botão `submit`, porém é exibida uma imagem, cuja localização é especificada pelo atributo `src`. É importante lembrar que, como regra de acessibilidade, toda imagem deve ter um texto alternativo (atributo `alt`).

Quando o envio do formulário é feito por um `input image`, são enviadas as coordenadas x (`pimagem.x`) e y (`pimagem.y`) da posição clicada da imagem.

## input type="submit|reset|button"

```
<input type="submit" name="pbotao1" value="Input Enviar" accesskey="E" />
<input type="submit" name="pbotao2" value="Input Enviar 2" accesskey="N" />
<input type="reset" value="Input Limpar" accesskey="L" />
<input type="button" id="pbotao3" value="Input Cliente" accesskey="C" />
```

Input Enviar

Input Enviar 2

Input Limpar

Input Cliente

O atributo `name` no `submit` só é necessário se houverem múltiplos botões para envio, de forma que o `name` (quando os nomes são diferentes) ou o `value` (para nomes iguais) possa ser usado para identificar qual dos botões foi pressionado para submeter o formulário. Sem o atributo `name`, o controle `submit` não se caracteriza como bem-sucedido, o que significa que apesar de realizar a ação de enviar o formulário, não gera um par nome/valor para ser enviado nos dados do formulário.

Um botão `reset` serve apenas para limpar o formulário, restaurando os valores dos campos ao seu padrão inicial, e não deve ser bem-sucedido.

Botões do tipo `button` não têm comportamento pré-definido, sendo usados apenas em scripts no lado cliente.

## button type="submit|reset|button"

```
<button type="submit" name="pbotao4" value="Button Enviar">
   Button Enviar
</button>
<button type="reset">Button Limpar</button>
<button type="button" id="pbotao5" accesskey="B">Button Cliente<br />
  Mais uma linha
```

```
</button>
```



Button Enviar

Button Limpar

Button Cliente  
Mais uma linha

Botões criados com o elemento `button` funcionam da mesma maneira que os criados com o elemento `input`, mas o `button` oferece possibilidades mais ricas de apresentação, pois possui conteúdo HTML como rótulo, definido entre as tags de abertura e fechamento do botão.

**Bug:** O Internet Explorer testado (6 SP1 Windows) parece ter dois bugs no elemento `button` do tipo `submit`; primeiro, o IE sempre o considera bem-sucedido quando ele possui um `name`, quando a especificação diz que isso deveria acontecer apenas se ele for o botão ativado para o envio do formulário; além disso, o IE envia o texto HTML como valor nos dados do formulário, quando o correto seria o conteúdo do atributo `value`. Já no navegador Mozilla testado (1.5 Windows), o comportamento deste controle funcionou corretamente.

## Tecla de acesso - accesskey

O atributo `accesskey` associa uma tecla de acesso a um elemento. Uma tecla de acesso deve ser um único caractere. Quando a tecla de acesso é ativada (no Windows, combinada com ALT), o controle correspondente recebe o foco. Os seguintes elementos suportam o atributo `accesskey`: `button`, `input`, `label`, `legend` e `textarea`, além dos elementos HTML `a` e `area`.

Ao longo destes exemplos foram definidas algumas teclas de acesso, que podem ser testadas: T (texto), S (senha), E (input enviar), N (input enviar 2), L (input limpar), C (input cliente), B (botão cliente 2).

Faltam ao padrão HTML controles mais sofisticados, como `grid` (tabela de campos), `tree` (árvore com abertura/fechamento de ramificações) entre outros. Em vários casos, esta ausência pode ser contornada, com algum esforço, simulando-se os controles desejados com o uso combinado de DHTML (HTML, JavaScript e CSS) e uma composição de componentes básicos disponíveis.

## Referências:

- [HTML 4.01 Forms](#), Especificação W3C.
- [How to write HTML forms](#), por Jukka "Yucca" Korpela. Em especial, veja o tópico [File input \(or "upload"\) in HTML forms](#).



ALGUNS DIREITOS RESERVADOS

© 2003-2007, Márcio d'Ávila, [mhavila.com.br](http://mhavila.com.br), direitos reservados. O texto e código-fonte apresentados podem ser referenciados, distribuídos e utilizados, desde que expressamente citada esta fonte e o crédito do(s) autor(es). A informação aqui apresentada, apesar de todo o esforço para garantir sua precisão e correção, é oferecida "como está", sem quaisquer garantias explícitas ou implícitas decorrentes de sua utilização ou suas consequências diretas e indiretas.

