

Input Controls

The input element is used to create a variety of form input controls, including:

- Single-line text entry fields
- Password entry fields
- Hidden controls
- Checkboxes
- Radio buttons
- Submit and reset buttons
- File upload mechanisms
- Custom and image buttons

The type attribute in the input element specifies the control type. The value of the type attribute also determines which other attributes may be used with the element. The input element and all of its accepted attributes appears here. Control-specific attribute listings appear along with the discussion of each control type.

input

<input />

Attributes

Core (id, class, style, title), *Internationalization*, *Events*, onfocus, onblur, onselect, onchange
alt="text"
accept="MIME type"
accesskey="character"
checked="checked"
disabled="disabled"
maxlength="number"
name="text" (*Required by all input types except submit and reset*)
readonly="readonly"
size="number"
src="URL"
tabindex="number"
type="text|password|checkbox|radio|submit|reset|file|hidden|image|button"
value="text"

Text entry field

The simplest type of form element is the text entry field (type="text"). Text is the default setting for the input element.

input type="text"

Attributes

Core (id, class, style, title), *Internationalization*, *Events*,
Focus (accesskey, tabindex, onfocus, onblur)
disabled="disabled"
maxlength=number
name="text" (*Required*)
readonly="readonly"
size="number"
value="text"

This field allows the user to enter a single word or a line of text. By default, the browser displays a text-entry box that is 20 characters wide, but you can set it to be any length using the size attribute.

By default, the user can type an unlimited number of characters into the field (the display scrolls to the right if the text exceeds the width of the supplied box), but you can set a maximum number of characters using the maxlength attribute.

Use the value attribute to specify the initial value, that is, the text to appear when the form is loaded. The user can change this default text. If you have a form that consists of only one text input element, pressing the Enter key submits the form without requiring a specific Submit button in the form. The following markup creates a text field with a size of 15 characters, a maximum length of 50 characters, and the text “enter your name” displayed in the field (Figure 15-2).

```
<p>What is your name?</p>  
<input type="text" name="name" size="15" maxlength="50" value="enter your  
name" />
```

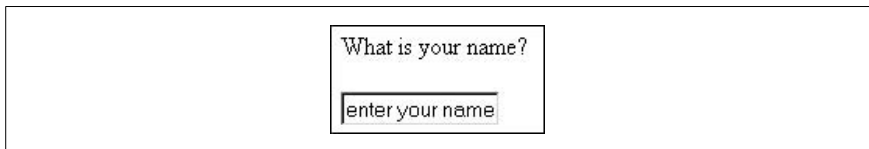


Figure 15-2. Text entry input control

Password text entry

A password field (type="password") works just like text entry, except the characters are obscured from view using asterisk (*) or bullet (•) characters (or another character determined by the user agent).

input type="password"

Attributes

Core (id, class, style, title), *Internationalization*, *Events*,
Focus (accesskey, tabindex, onfocus, onblur)
disabled="disabled"

```

maxlength="number"
name="text" (Required)
readonly="readonly"
size="number"
value="text" (Required)

```

The attributes and syntax for password entry fields are the same as for the text input type. The only difference is that values (such as the one provided as an initial value in this markup) are replaced with neutral characters, as shown in Figure 15-3.

```

<p>What is your password?</p>
<input type="password" name="password" size="8" maxlength="8"
value="abcdefg" />

```

Figure 15-3. Password input control



Although the characters entered into the password field are not visible to casual onlookers, the form does *not* encrypt the information entered and should not be considered to be a real security measure.

Hidden entry (type="hidden")

The hidden input (type="hidden") adds a control that isn't displayed in the browser, but is supplied to the form processor when the form is submitted.

input type="hidden"

Attributes

```

accesskey="character"
tabindex="number"
name="text" (Required)
value="text" (Required)

```

Hidden controls are useful for sending information to be processed along with the user-entered data, such as labels used by the script to sort forms. Users cannot see or alter hidden controls. Some scripts require specific hidden fields be added to the form in order to function properly. Here is a hidden element (Figure 15-4):

```

<p>This is a hidden element</p>
<input type="hidden" name="extra_info" value="important" />

```

Figure 15-4. Hidden input

Checkbox (type="checkbox")

Checkboxes (type="checkbox") are like on/off switches that can be toggled by the user. Several checkboxes in a group may be selected at one time, which makes them useful for multiple-choice questions where more than one answer is acceptable. When a form is submitted, only the “on” checkboxes submit values to the server.

input type="checkbox"

Attributes

Core (id, class, style, title), *Internationalization*, *Events*, *Focus* (accesskey, tabindex, onfocus, onblur)
align="left|right|top|texttop|middle|absmiddle|baseline|bottom|absbottom"
checked="checked"
disabled="disabled"
name="text" (*Required*)
readonly="readonly"
value="text" (*Required*)

Checkboxes can be used individually to transmit specific name/value coordinates to the server when checked. By default, a checkbox is not checked; to make it checked when the page loads, simply add the checked attribute to the corresponding input element. In XHTML, you must provide a value for every attribute, so the correct syntax is checked="checked".

When the box is checked, the corresponding value is transmitted with the form to the processing program on the server. The values for unchecked boxes are not sent.

If you assign a group of checkboxes the same name, they behave like a multiple-choice list in which the user can select more than one option for a given property, as shown in the following markup and in Figure 15-5.

```
<p>Which of the following operating systems have you used?</p>
<input type="checkbox" name="os" value="WinXP" /> Windows XP
<input type="checkbox" name="os" value="Linux" checked="checked" /> Linux
<input type="checkbox" name="os" value="OSX" checked="checked" /> Macintosh
OSX
<input type="checkbox" name="os" value="DOS" /> DOS
```

Which of the following operating systems have you used?

☐ Windows XP ☒ Linux ☒ Macintosh OSX ☐ DOS

Figure 15-5. Multiple checkboxes in a group may be selected

Radio button

Radio buttons (`type="radio"`) are another kind of button that users can toggle on and off. Unlike checkboxes, when a group of radio buttons share the same control name, only one button within the group can be “on” at one time, and all the others are “off.” They are used when the options are mutually exclusive.

input type="radio"

Attributes

Core (id, class, style, title), *Internationalization*, *Events*,
Focus (accesskey, tabindex, onfocus, onblur)
`checked="checked"`
`disabled="disabled"`
`name="text"` (*Required*)
`readonly="readonly"`
`value="text"` (*Required*)

In this example (Figure 15-6), only one operating system may be selected. The `checked` attribute makes the button “on” by default when the page loads. Only data from the “on” radio button is sent when the form is submitted.

```
<p>Which of the following operating systems have you used?</p>
<input type="radio" name="os" value="WinXP" /> Windows XP
<input type="radio" name="os" value="Linux" /> Linux
<input type="radio" name="os" value="OSX" checked="checked" /> Macintosh OSX
<input type="radio" name="os" value="DOS" /> DOS
```

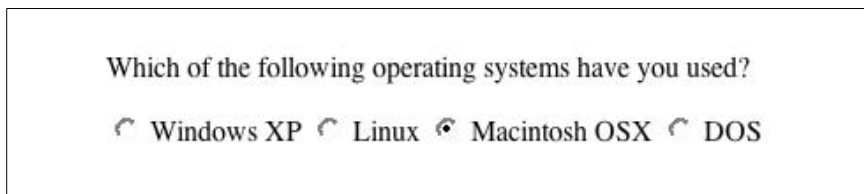


Figure 15-6. Only one radio button in a group may be selected

Submit and reset buttons

Submit buttons, used for sending the form data to the processing agent, are added with the `submit` input element type. Reset buttons return all form controls to their initial values and are added with the `reset` input element type.

input type="submit"

Creates a submit button control; pressing the button immediately sends the information in the form to the server for processing.

Attributes

Core (id, class, style, title), *Internationalization*, *Events*,
Focus (accesskey, tabindex, onfocus, onblur)
disabled="disabled"
name="text"
value="text"

input type="reset"

Creates a reset button that clears the contents of the elements in a form (or sets them to their default values).

Attributes

Core (id, class, style, title), *Internationalization*, *Events*,
Focus (accesskey, tabindex, onfocus, onblur)
disabled="disabled"
value="text"

Every form (unless it consists of exactly one text field) needs a submit button control to initiate the transmission of information to the server. A form may have more than one submit button. By default, the submit button (type="submit") says "Submit" or "Submit Query," but you can change it by adding your own text after the value attribute.

The reset button (type="reset") reverts all form controls back to the state they were in when the form loaded (either blank or with values provided by the author with the value attribute). The default value (and hence the label for the button) is "Reset," but like the submit button, you can change its text by specifying its value, as shown in Figure 15-7.

```
<p>You have completed the form.</p>  
<input type="submit" /><input type="reset" value="Start Over" />
```

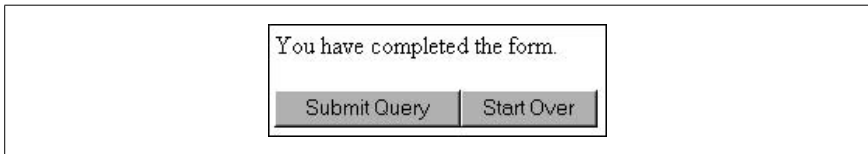


Figure 15-7. Submit and reset buttons

Some developers opt to leave the reset button out entirely, because there is no error-checking mechanism. If a user presses it accidentally, all the data already entered is lost. This isn't an uncommon occurrence.

Custom button

Authors may create a custom "push" button for use with client-side scripting (JavaScript) controls by setting the input type to button.

input type="button"

Attributes

Core (id, class, style, title), *Internationalization*, *Events*,
Focus (accesskey, tabindex, onfocus, onblur)
align="left|right|top|texttop|middle|absmiddle|baseline|bottom|
absbottom"
disabled="disabled"
name="text"
value="text"

This button (type="button") has no predefined function, but rather is a generic tool that can be customized with a scripting language such as JavaScript (the scripting language should be declared with a meta element). Use the value attribute to write your own text on the button, as shown in the following markup and in Figure 15-8. The data from a type="button" input element is never sent when a form is submitted; this type is useful only with script programs on the browser.

```
<p>This does something really exciting.</p>  
<input type="button" value="Push Me!" />
```

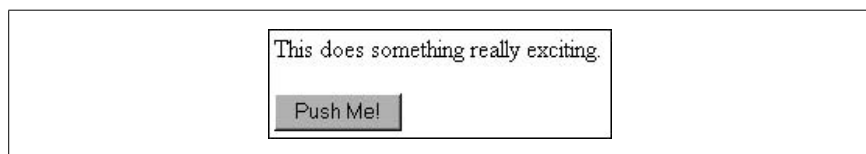


Figure 15-8. Custom button

Image button

If you want to use your own image for a submit button, use the image input type.

input type="image"

Attributes

Core (id, class, style, title), *Internationalization*, *Events*,
Focus (accesskey, tabindex, onfocus, onblur)
align="top|middle|bottom"
alt="text"
disabled="disabled"
name="text" (*Required*)
src="URL"

You can replace the submit button with a graphic of your choice by using the image input (type="image"), as shown in the markup example and in Figure 15-9. Clicking on the image submits the form to the server and includes the coordinates of the mouse click with the form data. You must provide the URL of the

graphic with the `src` attribute. It is recommended that you use alternative text (with the `alt` attribute) for image buttons.

```
<input type="image" src="graphics/sendme.gif" alt="Send me" />
```



Figure 15-9. Using an image for a button

File selection

The file input type allows users to submit external files with their form submission. The form control includes a text field and a “Browse” button that accesses the contents of the local computer.

input type="file"

Attributes

Core (`id`, `class`, `style`, `title`), *Internationalization*, *Events*, *Focus* (`accesskey`, `tabindex`, `onfocus`, `onblur`)
`accept`="MIME type"
`disabled`="disabled"
`maxlength`="number"
`name`="text" (*Required*)
`readonly`="readonly"
`size`="number"
`value`="text"

The file-selection form field (`type="file"`) lets users select a file stored on their computer and send it to the server when they submit the form. It is displayed as a text entry field with an accompanying “Browse” button for selecting the file, as shown in the following markup and in Figure 15-10. As for other text fields, you can set the `size` and `maxlength` values as well as the field’s default text. When using the file input type, you should specify `enctype="multipart/form-data"` in the form element.

```
<form enctype="multipart/form-data">  
<p>Send this file with my form information:</p>  
<input type="file" size="28" />  
</form>
```

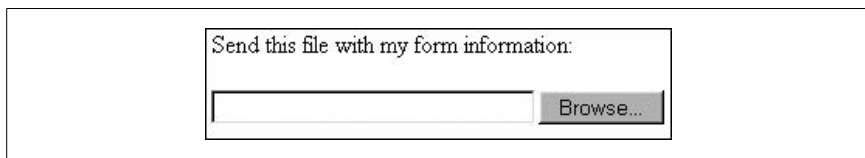


Figure 15-10. The file-selection form field