

# Create an HTML form

1. Open a page and place the insertion point where you want the form to appear.
2. Select Insert > Form, or select the Forms category in the Insert bar and click the Form icon.

In Design view, forms are indicated by a dotted red outline. If you don't see this outline, select View > Visual Aids > Invisible Elements.

3. Set the properties of the HTML form in the Property inspector (Window > Properties):

- a. In the Document window, click the form outline to select the form.
- b. In the Form Name box, type a unique name to identify the form.

Naming a form makes it possible to reference or control the form with a scripting language, such as JavaScript or VBScript. If you do not name the form, Dreamweaver generates a name using the syntax `formn`, and increments the value of *n* for each form added to the page.

- c. In the Action box, specify the page or script that will process the form data by typing the path, or clicking the folder icon to navigate to the appropriate page or script.
- d. In the Method pop-up menu, specify the method to transmit the form data to the server. Set any of the following options:

### **Default**

Uses the browser's default setting to send the form data to the server. Typically, the default value is the GET method.

### **GET**

Appends the value to the URL requesting the page.

### **POST**

Embeds the form data in the HTTP request.

Do not use the GET method to send long forms. URLs are limited to 8192 characters. If the amount of data sent is too large, data will be truncated, leading to unexpected or failed processing results.

Dynamic pages generated by parameters passed by the GET method can be bookmarked because all the values needed to regenerate the page are contained in the URL displayed in the browser's Address box. In contrast, dynamic pages generated by parameters passed by the POST method cannot be bookmarked.

If you collect confidential user names and passwords, credit card numbers, or other confidential information, the POST method may appear more secure than the GET method. However, the information sent by the POST method is not encrypted and can easily be retrieved by a hacker. To ensure security, use a secure connection to a secure server.

- e. (Optional) In the Enctype pop-up menu, specify the MIME encoding type of the data submitted to the server for processing.

The default setting of `application/x-www-form-urlencoded` is typically used in conjunction with the POST method. If you are creating a file-upload field, specify the `multipart/form-data` MIME type.

- f. (Optional) In the Target pop-up menu, specify the window in which to display the data returned by the invoked program.

If the named window is not already open, a new window with that name opens. Set any of the following target values:

### **\_blank**

Opens the destination document in a new unnamed window.

### **\_parent**

Opens the destination document in the parent window of the one displaying the current document.

### **\_self**

Opens the destination document in the same window as the one in which the form was submitted.

### **\_top**

Opens the destination document in the body of the current window. This value can be used to ensure that the destination document takes over the full window even if the original document was displayed in a frame.

4. Insert form objects in the page:

- a. Place the insertion point where the form object should appear in the form.
- b. Select the object in the Insert > Form menu, or in the Forms category of the Insert bar.
- c. Set the properties of the objects.
- d. Enter a name for the object in the Property inspector.

Every text field, hidden field, check box, and list/menu object must have a unique name that identifies the object in the form. Form object names cannot contain spaces or special characters. You can use any combination of alphanumeric characters and an underscore (\_). The label you assign to the object is the variable name that stores the value (the entered data) of the field. This is the value sent to the server for processing.

**Note:** All radio buttons in a group must have the same name.

- e. To label the text field, check box, or radio button object on the page, click beside the object and type the label.
5. Adjust the layout of the form.
- Use line breaks, paragraph breaks, preformatted text, or tables to format your forms. You cannot insert a form in another form (that is, you cannot overlap tags), but you can include more than one form in a page.
- When designing forms, remember to label the form fields with descriptive text to let users know what they're responding to—for example, "Type your name" to request name information.
- Use tables to provide structure for form objects and field labels. When using tables in forms make sure all the `table` tags are included between the `form` tags.

For a tutorial on creating forms, see [www.adobe.com/go/vid0160](http://www.adobe.com/go/vid0160).

For a tutorial on styling forms with CSS, see [www.adobe.com/go/vid0161](http://www.adobe.com/go/vid0161).

### Insert a group of radio buttons

1. Place the insertion point inside the form outline.
2. Select Insert > Form > Radio Group.
3. Complete the dialog box and click OK.
  - a. In the Name box, enter a name for the radio button group.

If you set the radio buttons to pass parameters back to the server, the parameters are associated with the name. For example, if you name the group `myGroup` and set the form method to `GET` (that is, you want the form to pass URL parameters instead of form parameters when the user clicks the submit button), the expression `myGroup="CheckedValue"` is passed in the URL to the server.
  - b. Click the Plus (+) button to add a radio button to the group. Enter a label and checked value for the new button.
  - c. Click the up or down arrows to reorder the buttons.
  - d. To set a particular radio button to be selected when the page opens in a browser, enter a value equal to the radio button's value in the Select Value Equal To box.

Enter a static value or specify a dynamic one by clicking the lightning bolt icon beside the box and selecting a recordset that contains possible checked values. In either case, the value you specify should match the checked value of one of the radio buttons in the group. To view the checked values of the radio buttons, select each radio button and open its Property inspector (Window > Properties).
  - e. Select the format in which you want Dreamweaver to lay out the buttons.

Lay out the buttons using line breaks or a table. If you select the table option, Dreamweaver creates a single-column table, and places the radio buttons on the left and the labels on the right.
4. You can also set the properties in the Property inspector or directly in Code view.

### Insert an image button

You can use images as button icons. Using an image to perform tasks other than submitting data requires attaching a behavior to the form object.

1. In the document, place the insertion point inside the form outline.
2. Select Insert > Form > Image Field.  
The Image Source dialog box appears.
3. Select the image for the button in the Select Image Source dialog box, and click OK.
4. Set any of the following options in the Property inspector:

**ImageField**

Assigns a name to the button. Two reserved names, Submit and Reset, tell the form to submit the form data to the processing application or script, or to reset all the form fields to their original values, respectively.

**Src**

Specifies the image you want to use for the button.

**Alt**

Lets you enter descriptive text in case the image fails to load in the browser.

**Align**

Sets the align attribute of the object.

**Edit Image**

starts your default image editor and opens the image file for editing.

**Class**

Lets you apply CSS rules to the object.

5. To attach a JavaScript behavior to the button, select the image, and then select the behavior in the Behaviors panel (Window > Behaviors).

 **Insert file-upload fields**

You can create a file-upload field that lets users select a file on their computer—such as a word processing document or graphics file—and upload the file to the server. A file field looks like other text fields except it also contains a Browse button. The user either manually enters the path to the file they want to upload, or uses the Browse button to locate and select the file.

Before you can use file-upload fields, you must have a server-side script or a page capable of handling file submissions. Consult the documentation of the server technology you use to process form data. For example, if you use PHP, see “Handling files uploads” in the online PHP Manual at <http://us2.php.net/features.file-upload.php>.

File fields require that you use the POST method to transmit files from the browser to the server. The file is posted to the address you specify in the form’s Action box.

**Note:** Contact your server’s administrator to confirm that anonymous file uploads are allowed before using the file field.

1. Insert a form in the page (Insert > Form).
2. Select the form to display its Property inspector.
3. Set the form Method to POST.
4. From the Enctype pop-up menu, select `multipart/form-data`.
5. In the Action box, specify the server-side script or page capable of handling the uploaded file.
6. Place the insertion point inside the form outline, and select Insert > Form > File Field.
7. Set any of the following options in the Property inspector:

**File Field Name**

Specifies the name for the file field object.

**Char Width**

Specifies the maximum number of characters that can be displayed in the field.

### **Max Chars**

Specifies the maximum number of characters the field will hold. If the user browses to locate the file, the filename and path can exceed the specified Max Chars value. However, if the user attempts to type in the filename and path, the file field will only allow the number of characters specified by the Max Chars value.

## **HTML text field object properties**

Select the text field object, and set any of the following options in the Property inspector:

### **Char Width**

Specifies the maximum number of characters that can be displayed in the field. This number can be less than Max Chars, which specifies the maximum number of characters that can be entered in the field. For example, if the Char Width is set to 20 (the default value) and a user enters 100 characters, only 20 of those characters will be viewable in the text field. Although you cannot view the characters in the field, they are recognized by the field object and are sent to the server for processing.

### **Max Chars**

Specifies the maximum number of characters that the user can enter in the field for single-line text fields. Use Max Chars to limit zip codes to 5 digits, limit passwords to 10 characters, and so on. If you leave the Max Chars box blank, users can enter any amount of text. If the text exceeds the character width of the field the text will scroll. If a user exceeds the maximum number of characters, the form produces an alert sound.

### **Num Lines**

(Available when the Multiline option is selected) Sets the height of the field for multiple-line text fields.

### **Wrap**

(Available when the Multiline option is selected) Specifies how the user's input will be displayed when the user enters more information than can be displayed within the defined text area. The wrap options are as follows:

#### **Off or Default**

Prevents text from wrapping to the next line. When the user's input exceeds the right boundary of the text area, text scrolls toward the left. Users must press Return to move the insertion point to the next line in the text area.

#### **Virtual**

Sets word wrap in the text area. When the user's input exceeds the right boundary of the text area, text wraps to the next line. When data is submitted for processing, word wrap isn't applied to the data. It is submitted as one string of data.

#### **Physical**

Sets word wrap in the text area, as well as to the data when it is submitted for processing.

### **Type**

Designates the field as a single-line, multiple-line, or password field.

#### **Single-line**

Results in an `input` tag with its `type` attribute set to `text`. The Char Width setting maps to the `size` attribute, and the Max Chars setting maps to the `maxLength` attribute.

#### **Multi-line**

Results in a `textarea` tag. The Char Width setting maps to the `cols` attribute, and the Num Lines setting maps to the `rows` attribute.

#### **Password**

Results in an `input` tag with its `type` attribute set to `password`. The Char Width and Max Chars settings map to the same attributes as in single-line text fields. When a user types in a password text field, the input appears as bullets or asterisks to protect it from observation by others.

### **Init Value**

Assigns the value displayed in the field when the form first loads. For example, you might indicate that the user enters information in the field by including a note or example value.

### **Class**

Lets you apply CSS rules to the object.

## **Button object options**

### **Button Name**

Assigns a name to the button. Two reserved names, Submit and Reset, tell the form to submit the form data to the processing application or script, or to reset all the form fields to their original values, respectively.

### **Label**

Determines the text that appears on the button.

### **Action**

Determines what happens when the button is clicked.

#### **Submit Form**

Submits the form data for processing when the user clicks the button. The data is submitted to the page or script specified in the form's Action property.

#### **Reset Form**

Clears the contents of the form when the button is clicked.

#### **None**

Specifies the action to be performed when the button is clicked. For example, you can add a JavaScript behavior that opens another page when the user clicks the button.

### **Class**

Applies CSS rules to the object.

## **HTML check box object options**

### **Checked Value**

Sets the value to be sent to the server when the check box is checked. For example, in a survey you might set a value of 4 for strongly agree and a value of 1 for strongly disagree.

### **Initial State**

Determines whether the check box is selected when the form loads in the browser.

### **Dynamic**

Lets the server dynamically determine the initial state of the check box. For example, you can use check boxes to visually present the Yes/No information stored in a database record. At design time, you don't know that information. At run time the server reads the database record and selects the check box if the value is Yes.

### **Class**

Applies Cascading Style Sheets (CSS) rules to the object.

## **Single radio button object options**

### **Checked Value**

Sets the value to be sent to the server when the radio button is selected. For example, you might type `skiing` in the Checked Value box to indicate a user chose skiing.

### **Initial State**

Determines whether the radio button is selected when the form loads in the browser.

### **Dynamic**

Lets the server dynamically determine the initial state of the radio button. For example, you can use radio buttons to visually present information stored in a database record. At design time, you don't know that information. At run time the server reads the database record and checks the radio button if the value matches one you specified.

### **Class**

Applies CSS rules to the object.

## **Menu options**

### **List/Menu**

Assigns a name to the menu. The name must be unique.

### **Type**

Indicates whether the menu drops down when clicked (the Menu option) or displays a scrollable list of items (the List option). Select the Menu option if you want only one option to be visible when the form is displayed in a browser. To display the other choices, the user must click the down arrow.

Select the List option to list some or all the options when the form is displayed in a browser, to let users select multiple items.

**Height**

(List type only) Sets the number of items displayed in the menu.

**Selections**

(List type only) Indicates whether the user can select multiple items from the list.

**List Values**

Opens a dialog box that lets you add the items to a form menu:

1. Use the Plus (+) and Minus (-) buttons to add and remove items in the list.
2. Enter label text and an optional value for each menu item.

Each item in the list has a label (the text that appears in the list) and a value (the value that is sent to the processing application if the item is selected). If no value is specified, the label is sent to the processing application instead.

3. Use the up and down arrow buttons to rearrange items in the list.

Items appear in the menu in the same order as they appear in the List Values dialog box. The first item on the list is the selected item when the page is loaded in a browser.

**Dynamic**

Lets the server dynamically select an item in the menu when it is first displayed.

**Class**

Lets you apply CSS rules to the object.

**Initially Selected**

Sets the items selected in the list by default. Click the item or items in the list.

 **Hidden field object options****HiddenField**

Specifies the name for the field.

**Value**

Assigns a value to the field. This value is passed to the server when the form is submitted.

**Class**

Applies CSS rules to the object.

[http://livedocs.adobe.com/en\\_US/Dreamweaver/9.0/help.html?content=WSc78c5058ca073340dcda9110b1f693f21-7936.html](http://livedocs.adobe.com/en_US/Dreamweaver/9.0/help.html?content=WSc78c5058ca073340dcda9110b1f693f21-7936.html)