

## Creating a form

As a web developer you will be asked many times throughout your career to create a web form. These forms can gather various types of visitor information and implement functionality such as:

- Membership registration
- Update contact information
- Purchase merchandise
- Provide a simple feedback form

The list can continue quite extensively. Creating a web form which provides various ways to gather visitor information is easy to do in most cases. In this tutorial, we'll examine the following:

- Form tag
  - Input
  - Drop down list
  - Text area
  - Radio button
  - Checkboxes
  - Submit & reset button

If you would like to follow creating a form step-by-step development or try your hand at creating a web form, you will find the project files link helpful.

Finished Page

<http://midwestwebdesign.net/tutorials/forms/forms.html>

Project Zip

<http://midwestwebdesign.net/tutorials/forms/forms.zip>

### Form tag

In your text editor create a new file named **forms.html** and add the following code:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-
1">
</head>
<body>
<form action="" method="post" name="simple" >
</form>
</body>
</html>
```

## Forms

When creating forms you use a form tag which serves as a container for other form fields used to gather visitor information. Forms are similar to tables or DIVS because they have an opening and closing tag:

- <form>
  - Opening tag
- </form>
  - Closing tag

The individual attributes inside the opening form tag are discussed below:

**Action:** This attribute allows you to set another file used in processing visitor information. Developers usually use the following:

- CGI
- ASP or ASP.NET
- PHP
- Coldfusion

When a visitor presses the submit button, it causes the action attribute to execute and call the server-side file. This file contains code to do the following:

- Process the results into a readable format
- Perform any server-side validation needed
- Send an email to a specified recipient(s)

**Method:** This attribute deals with how information with your form is used. Possible values are:

- Get
- Post

Nine times out of ten, you'll use post, which hides information from the visitor, and passes the information internally. Get, will pass visitor information through the address bar and will be readily seen by all who use the application. If you're passing sensitive information such as banking or social security numbers, you should always use the post method.

**Name:** This attribute is the name of your form.

**ID:** This attribute gives your form a unique name, which is then used in styling a web form with CSS.

### Creating the form

Inside the opening and closing form tag, we need a way to lay out a form. We have two options:

- CSS
- Table

There are strong debates on which is more appropriate. While neither method is right or wrong, most of the time forms are designed in a tabular format, which means they are a perfect candidate to be placed inside a form. General rules of thumb, if the form represents tabular data, use a table, if not, then use CSS. If you decide to use CSS, make sure you exhaustively test your form in all modern browsers to account for any browsers inconsistencies. For this tutorial's purpose, we'll use a table.

## Create the table

Inside the opening and closing form tag, add the following code:

```
<form action="" method="post" name="simple">
<table width="500" cellpadding="0" cellspacing="0" align="center">
</table>
```

Inside the opening table tag, we have the following:

- Width
  - Set to a fixed width of 500 pixels
- Cell padding
  - Set to 0
- Cell spacing
  - Set to 0
- Align
  - Set to center

The last attribute/value will center our table in the browser. Save your file.

## Input

In order to collect the name of our visitor, we need:

- One cell to ask the name of our visitor
- One cell for an input text field

Add the following code:

```
<table width="500" cellpadding="0" cellspacing="0" align="center">
<tr>
<td>Name:</td>
<td><input name="name" type="text" size="30"></td>
</tr>
</table>
```

We add one row with two columns. The first column asks our visitor for their name. The second column has an input text field for the visitor to input their name. The attributes/values of the input tag are discussed below:

**Type:**

- Specifies the type of input, in this example, text

**Name:**

- Specifies the name of our input field, in this example, name

**Size:**

- Specifies the width of our input field, in this example, 30

**Maxlength:**

- Optional. Specifies the number of characters which are allowed in text input field

Save your file and preview the results.

[http://midwestwebdesign.net/tutorials/forms/forms\\_2.html](http://midwestwebdesign.net/tutorials/forms/forms_2.html)

Since the email field uses the same input tag, we simply add the following code:

```
<tr>
<td>E-mail</td>
<td><input name="Email" type="text" size="30"></td>
</tr>
```

Since we've covered this input tag, we'll move forward. Save your file and preview the results.

[http://midwestwebdesign.net/tutorials/forms/forms\\_3.html](http://midwestwebdesign.net/tutorials/forms/forms_3.html)

**Select**

Many times when creating forms, there will be times when you are squeezed for space in your web page. This presents a problem when you need to offer the visitor multiple options to choose. In this scenario, a select tag is an obvious choice and simple to create. We add the following code:

```
<tr>
<td>Rate this site: </td>
<td><select name="ratesite">
<option value="null">-----</option>
<option value="excellent">Excellent</option>
<option value="good">Good</option>
<option value="poor">Poor</option>
</select>
</td>
</tr>
```

You create a drop down (select list) by using the select tag. Inside the opening select tag we have the following attribute/value:

- name=ratesite
  - Specifies the name of our drop down list

Inside the select tag, to provide our visitors with multiple options to choose, we use the option tag. Inside the option tag, we have the following attribute/value:

- value=null
  - Specifies the value of each option tag

After the closing the opening option tag, we provide descriptive text for each option. The reason our first option is set to null and a series of dashes is to allow our visitor to select an option instead of making the first option, excellent, as a default option. Save your file and preview the results.

[http://midwestwebdesign.net/tutorials/forms/forms\\_4.html](http://midwestwebdesign.net/tutorials/forms/forms_4.html)

## Text Area

In most web forms, you want visitors to enter comments and/or suggestions into a text box. We can do this easily in a web form by using a text area. Add the following code:

```
<tr>
<td>Referred By: </td>
<td><textarea name="referredby" cols="25" rows="4"></textarea></td>
</tr>
```

The text area tag is one of the few form fields which requires an opening and closing tag, at least in the HTML 4.01 specification. In XHTML, the remaining form fields must have an opening and closing tag as well. Continuing, attributes/values of the text area tag are as follows:

### Name:

- Specifies the name of our text area, in our case, referredby

### Cols:

- Specifies the width of your text area, in our case, 25

### Rows:

- Specifies the height of your text area, in our case, 4

Save your file and preview the results.

[http://midwestwebdesign.net/tutorials/forms/forms\\_5.html](http://midwestwebdesign.net/tutorials/forms/forms_5.html)

## Radio Button

Sometimes in web forms, you want to list multiple choices for your visitors and allow them to choose only one. If space on your web page isn't an issue, we can easily provide our visitors this option with radio buttons. Add the following code:

```
<tr>
<td valign="top">Favorite Search Engine:</td>
<td>
<input type="radio" name="searchengine" value="google">
Google<br />
<input type="radio" name="searchengine" value="msn">
MSN<br />
<input type="radio" name="searchengine" value="altavista">
Alta Vista
</td>
</tr>
```

You might notice radio buttons are created using the input tag as text field. What makes a text field become a radio button is specifying the type as radio. The attribute/values of the input tag are:

### Type:

- Specifies the type of input, in this example, radio button

### Name:

- Specifies the name of our input field, in this example, searchengine

### Value:

- Specifies the value of each radio button, in this example, Google is first, followed by msn and Alta Vista

You might be wondering why all our radio buttons share the same name. The reason is simple – radio buttons are treated as a group. Thus they share the same name, in our case, search engine. The only difference is the value, which is the option the visitor chose. Save your file and preview the results.

[http://midwestwebdesign.net/tutorials/forms/forms\\_6.html](http://midwestwebdesign.net/tutorials/forms/forms_6.html)

## Check boxes

Sometimes in web forms, you want to list multiple choices for your visitors and allow them to choose multiple answers. If space on your web page isn't an issue, we can easily provide our visitors this option with checkboxes. Add the following code:

```
<tr>
```

```

<td valign="top">Which products do you use?</td>
<td><input type="checkbox" name="products" value="mx 2004">
Macromedia Studio MX 2004<br />
<input type="checkbox" name="products" value="ms office 2004">
MS Office 2003<br />
<input type="checkbox" name="products" value="vs 2004">
Visual Studio .NET<br />
<input type="checkbox" name="products" value="ad aware">
Ad-Aware</td>
</tr>

```

You might notice checkboxes are created using the same input tag as a text field. What makes a text field become a radio button is specifying the type as radio. The attribute/values of the input tag are:

**Type:**

- Specifies the type of input, in this example, checkbox

**Name:**

- Specifies the name of our input field, in this example, products

**Value:**

- Specifies the value of each radio button, in this example, Macromedia is first, followed by Office, Visual Studio, and Ad-Aware

You might be wondering why all our check boxes share the same name. The reason is simple – check boxes are treated as a group. Thus they share the same name, in our case, products. The only difference is the value, which are the options the visitor chooses. Save your file and preview the results.

[http://midwestwebdesign.net/tutorials/forms/forms\\_7.html](http://midwestwebdesign.net/tutorials/forms/forms_7.html)

**Submit and Reset buttons**

Since we have completed our assessment of the various form fields, we need a way to submit or reset this information. Add the following code:

```

<tr>
<td colspan="2" align="center"><br>
<input type="submit" name="submit" value="submit">
<input type="reset" name="reset" value="reset">
</td>
</tr>

```

First, we want our buttons to span the form and table; we set the last table cell to span two columns by using colspan with a value of 2. To center our buttons, we add align with a value of center. To provide a bit of vertical separation between the previous form field and the buttons, a break tag is added. Again, you might notice our buttons are created using the same input tag as a

text field, radio button or checkbox. What makes an input become a button is specifying the type as either submit or reset. The attribute/values of the input tag are:

**Type:**

- Specifies the type of input, in this example, submit and reset

**Name:**

- Specifies the name of our input field, in this example, submit and reset

**Value:**

- Specifies the value of each button, in this example, submit and reset

More importantly, the value is what enables the form to either submit the information or reset the value of our forms fields to empty values. Save your file and preview the results.

[http://midwestwebdesign.net/tutorials/forms/forms\\_8.html](http://midwestwebdesign.net/tutorials/forms/forms_8.html)

**Summary**

In this article, you learned how to create a form. You also learned about:

- Attribute/value pairs of form tags
- Various form fields used, which include:
  - Text fields
  - Drop down (select)
  - Text area tags (comments)
  - Radio buttons
  - Checkboxes
  - Submit and reset buttons

Take the knowledge gained here and create any form imaginable!

If you have questions, please follow the link below.

<http://midwestwebdesign.net/tutorials/contact.aspx>