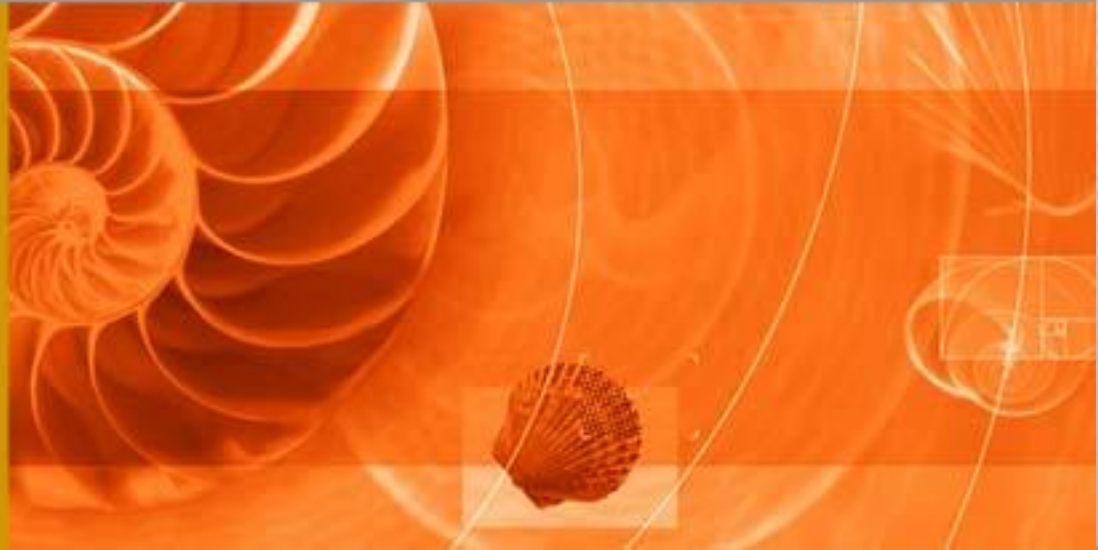


**Fundamentals
of
WEB
DESIGN**



Form Tags

XHTML

About Forms

- Forms allow you to collect information from visitors to your Web site. The example below is a typical tech support form for a visitor to ask a question. More complex forms can be used to place online orders, give online exams, make database updates, and more.

Tech Support	
Name	<input type="text" value="Mort Wiggy"/>
Email Address	<input type="text" value="Mort@wiggy.com"/>
Question	<input type="text" value="I have lost my password, what should I do?"/>

Press when you have completed this form.

Viewing Form Results

- To see the results of a submitted form requires an program on the server to process the form. Though it is not within the scope of this course to cover server-side programming, we will describe a server-side program we can use to display the form results without doing any programming.

Processed form looks like this:

```
username : Mort Wiggy
userpass :
email    : Mort@wiggy.com

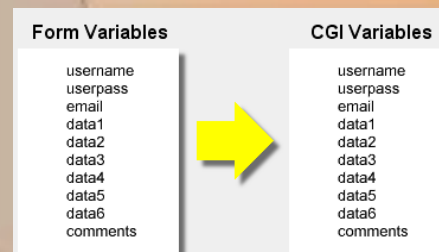
DATA:
-----
data1 : I have lost my password, what should I do?
data2 :
data3 :
data4 :
data5 :
data6 :

COMMENTS:
-----

cgimail 1.5
```

Forms and Form Variables

- Forms are used to collect user data and submit it to a CGI script or other program on the server capable of processing form data. In order for the form data to be processed correctly, the variable names in the form and in the CGI script must be identical. Form variables that do not have matching variables in the CGI script will be ignored during processing. The example below illustrates the variable names that will be used in the form examples in this lesson and then passed to a CGI script.



CGlecho and CGlemail

- Unless you are a programmer, you will need a prewritten CGI Script to handle form processing. A very popular choice for processing form data is CGlecho/CGlemail. The CGlecho script is used for testing a form's output, while CGlemail is used for sending the processed form data via email. We will use CGlecho for testing all the examples and exercises in this lesson.

Formatting the Template File

- CGlecho/CGlemail requires a simple text file called a template to format form data. The template file contains both variables and free form text. Notice below that the text within the template is not restricted. Square brackets are placed around variable names "*[variablename]*", which the CGI script will replace with the variable values passed by the form. The name of the template file, "form.txt", can be any valid filename.

```
form.txt

username : [username]
userpass : [userpass]
email    : [email]

DATA:
-----
data1 : [data1]
data2 : [data2]
data3 : [data3]
data4 : [data4]
data5 : [data5]
data6 : [data6]

COMMENTS:
-----
[comments]
```

<form> Tag

- The <form> tag starts a form and the </form> tag ends it. All other tags associated with forms will not function unless they are inside <form></form> tags. Note that when editing forms, you often must hold down the "Shift" key while hitting "Reload"/"Refresh" in order to see your changes. On rare occasions, you will need to restart the browser to see edit changes to forms. If restarting the browser does not fix a form problem, then there is a mistake in the form itself. The form below is not complete; therefore the submit button does not function.

```
<body bgcolor="#bdbdbd" text="black" link="blue" vlink="purple">
<center>
<h2>Basic Form Example</h2>
<form>
Enter your name:
<input type="text" name="username" value="" /><br />
<br />
<input type="submit" value="Submit Form" />
</form>
</center>
</body>
```

Method & Action Attributes

- The previous example would not submit information because there were no form attributes describing what to do on submit. The "method" attribute, which determines how your form data is sent, can have a value of "get" or "post," but "method="post"" is the most common.
- Next, the "action" attribute points to the script that will process the form. This example points to "cgiecho," which is a common testing tool for processing form email. Submitting this form example results in a list of all the variable names supported by this test script.

```
<body bgcolor="#bdbdbd" text="black" link="blue" vlink="purple">
<center>
<h2>Basic Form Example</h2>
<form method="post"
action="http://www.webyoda.com/cgi-bin/cgiecho/form.txt">
Enter your name:
<input type="text" name="username" value="" /><br />
<br />
<input type="submit" value="Submit Form" />
</form>
</center>
</body>
```

Type Attribute

- With the form framework in place, we can now add functionality to the form. By default, the `<input />` tag indicates a text input field. In the example below, the "type" attribute has been set to "text," but it is not required.
- The `<input />` tag has no closing tag, thus the " /" (there is a space before the slash) is placed at the end of the tag to make it XHTML compliant.

```
<form method="post"
      action="http://www.webyoda.com/cgi-bin/cgiecho/form.txt">
Enter your name:
<input type="text" name="username" /><br />
<br />
<input type="submit" value="Submit Form" />
</form>
```

Name Attribute

- The "name" attribute is used to define the variable name that will pass data. For the data in the variable name to be processed, it must match the variable name in the processing script exactly. Variable names are case sensitive.

```
<form method="post"
      action="http://www.webyoda.com/cgi-bin/cgiecho/form.txt">
Enter your name:
<input type="text" name="username" /><br />
<br />
<input type="submit" value="Submit Form" />
</form>
```

Value Attribute

- The "value" attribute defines the default value for the input variable. If the value attribute is not used, a default value of "null" is assigned to the variable. The "size" attribute is used to set the length of the text input window, whereas the "maxlength" attribute defines how many characters of input will be accepted.

```
<form method="post"
      action="http://www.webyoda.com/cgi-bin/cgiecho/form.txt">
Enter your name:
<input type="text" name="username" value=""
      size="20" maxlength="5" />
<br />
<br />
<input type="submit" value="Submit Form" />
</form>
```

<input> Tag

- To submit form data, use the `<input />` tag, with the "type" attribute set to "submit." You can control the text on the submit button by setting the "value" attribute.

```
<form method="post"
      action="http://www.webyoda.com/cgi-bin/cgiecho/form.txt">
Enter your name:
<input type="text" name="username" value="" />
<br />
<br />
<input type="submit" value="Submit Form" />
</form>
</center>
</body>
</html>
```

Type Attribute

- To input a text field as a password, use the `<input />` tag with the "type" attribute set to "password." The input field will display an "*" each time a key is pressed.

```
<form method="post"
      action="http://www.webyoda.com/cgi-bin/cgiecho/form.txt">
Enter your name:
<input type="text" name="username" /><br />
Enter password:
<input type="password" name="userpass" /><br />
<br />
<input type="submit" value="Submit Form" />
</form>
</center>
</body>
</html>
```

Type Attribute

- In any form, to give the user a way to reset form data to the defaults, use the `<input />` tag with the "type" attribute set to "reset." You can control the text on the reset button by setting the "value" attribute.

```
<form method="post"
      action="http://www.webyoda.com/cgi-bin/cgiecho/form.txt">
<font size="+1"> Please enter your comments:</font><br />
<center>
<br />
<textarea name="comments" rows="6" cols="60"></textarea>
<br />
<br />
<input type="submit" value="Submit Form" />
<input type="reset" value="Clear Form" />
</center>
</form>
```

Form Check Boxes

- To input data using check boxes, use the `<input />` tag with the "type" attribute set to "checkbox." The variable names used with each check box must be the same as the ones used in the processing script, or the data will not be collected. By default, any check boxes that are checked submit a value of "on," while unchecked boxes send no value. Notice that only the values for the check boxes under "Kitchen" show up in the results, since they use variable names that are recognized by the test script ("data1" through "data5").

```
<form method="post"
      action="http://www.webyoda.com/cgi-bin/cgiecho/form.txt">
<font size="+2">Kitchen:</font><br />
<input type="checkbox" name="data1" />Spoons<br />
<input type="checkbox" name="data2" />Forks<br />
<input type="checkbox" name="data3" />Plates<br />
<input type="checkbox" name="data4" />Napkins<br />
<input type="checkbox" name="data5" />Cups<br />
<br />
<font size="+2">Bathroom:</font><br />
<input type="checkbox" name="Towels" /> Towels<br />
<input type="checkbox" name="Razors" /> Razors<br />
<input type="checkbox" name="Shampoo" /> Shampoo<br />
<input type="checkbox" name="Q-Tips" /> Q-Tips<br />
<input type="checkbox" name="Soap" /> Soap<br />
<br />
<input type="submit" value="Submit Form" />
</form>
```

Value Attribute

- Add the "value" attribute to define the data sent when a check box is checked. In this example, all items send descriptive values instead of "on."

```
<form method="post" action=
    "http://www.webyoda.com/cgi-bin/cgiecho/form.txt">
<font size="+2">Kitchen:</font><br />
<input type="checkbox" name="data1" value="Spoons" />Spoons<br />
<input type="checkbox" name="data2" value="Forks" />Forks<br />
<input type="checkbox" name="data3" value="Plates" />Plates<br />
<input type="checkbox" name="data4" value="bowls" />Bowls<br />
<input type="checkbox" name="data5" value="Cups" />Cups<br />
<br />
<font size="+2">Bathroom:</font><br />
<input type="checkbox" name="Towel" value="Towel" />Towel<br />
<input type="checkbox" name="Soap" value="Soap" />Soap<br />
<input type="checkbox" name="Razor" value="Razor" />Razor<br />
<input type="checkbox" name="Meds" value="Meds" />Meds<br />
<input type="checkbox" name="Q-Tips" value="Q-Tips" />Q-Tips<br />
<br />
<input type="submit" value="Submit Form" />
</form>
```

Checked Attribute

- By default, check boxes are unchecked. You can establish the check box default as checked by using the "checked" attribute.

```
<form method="post"
      action="http://www.webyoda.com/cgi-bin/cgiecho/form.txt">
<font size="+2">Kitchen:</font><br />
<input type="checkbox" name="data1" /> Spoons<br />
<input type="checkbox" name="data2" /> Forks<br />
<input type="checkbox" name="data3" /> Plates<br />
<input type="checkbox" name="data4" /> Napkins<br />
<input type="checkbox" name="data5" checked="checked"/>Cups<br />
<br />
<font size="+2">Bathroom:</font><br />
<input type="checkbox" name="Towels" /> Towels<br />
<input type="checkbox" name="Razors" /> Razors<br />
<input type="checkbox" name="Meds" /> Meds<br />
<input type="checkbox" name="Q-Tips" /> Q-Tips<br />
<input type="checkbox" name="Soap" checked="checked"/>Soap<br />
<br />
<input type="submit" value="Submit Form" />
</form>
```

Form Radio Buttons

- To collect data using radio buttons, use the `<input />` tag with the "type" attribute set to "radio." The variable names used with radio buttons must be the same as the ones used in the processing script or the data will not be collected. By default, checked radio buttons submit a value of "on," while unchecked buttons send no value. Notice that all items under "Kitchen" use the same variable name (the same is true for the items under "Bathroom"). This is because a group of radio buttons permit the selection of only one choice from a list of options.

```
<form method="post"
      action="http://www.webyoda.com/cgi-bin/cgiecho/form.txt">
  <font size="+2">Kitchen:</font><br />
  <input type="radio" name="data1" />Spoons<br />
  <input type="radio" name="data1" />Forks<br />
  <input type="radio" name="data1" />Plates<br />
  <input type="radio" name="data1" />Napkins<br />
  <input type="radio" name="data1" />Cups<br />
  <br />
  <font size="+2">Bathroom:</font><br />
  <input type="radio" name="data2" /> Towels<br />
  <input type="radio" name="data2" /> Razors<br />
  <input type="radio" name="data2" /> Shampoo<br />
  <input type="radio" name="data2" /> Q-Tips<br />
  <input type="radio" name="data2" /> Soap<br />
  <br />
  <input type="submit" value="Submit Form" />
</form>
```

Value Attribute

- The "value" attribute can be added to the <input /> tag to define the data sent when a radio button is checked. In the example below, all items have descriptive values instead of "on."

```
<form method="post"
      action="http://www.webyoda.com/cgi-bin/cgiecho/form.txt">
  <font size="+2">Kitchen:</font><br />
  <input type="radio" name="data1" value="Spoons" /> Spoons<br />
  <input type="radio" name="data1" value="Forks" /> Forks<br />
  <input type="radio" name="data1" value="Plates" /> Plates<br />
  <input type="radio" name="data1" value="Napkins" /> Napkins<br />
  <input type="radio" name="data1" value="Cups" /> Cups<br />
  <br />
  <font size="+2">Bathroom:</font><br />
  <input type="radio" name="data2" value="Towels" />Towels<br />
  <input type="radio" name="data2" value="Razors" />Razors<br />
  <input type="radio" name="data2" value="Shampoo" />Shampoo<br />
  <input type="radio" name="data2" value="Q-Tips" />Q-Tips<br />
  <input type="radio" name="data2" value="Soap" />Soap<br />
  <br />
  <input type="submit" value="Submit Form" />
</form>
```

Checked Attribute

- By default, radio buttons are unchecked. A radio button's default can be set as checked with the "checked" attribute, however, only one radio button per group should be checked (a group of radio buttons uses the same variable name). In this example, "Cups" is checked for the data1 variable and "Soap" is checked for the data2 variable.

```
<form method="post"
      action="http://www.webyoda.com/cgi-bin/cgiecho/form.txt">
  <font size="+2">Kitchen:</font><br />
  <input type="radio" name="data1" value="Spoons" />Spoons<br />
  <input type="radio" name="data1" value="Forks" />Forks<br />
  <input type="radio" name="data1" value="Plates" />Plates<br />
  <input type="radio" name="data1" value="Napkins" />Napkins<br />
  <input type="radio" name="data1" value="Cups"
        checked="checked" />Cups
<br /><br />
  <font size="+2">Bathroom:</font><br />
  <input type="radio" name="data2" value="Towels" />Towels<br />
  <input type="radio" name="data2" value="Razors" />Razors<br />
  <input type="radio" name="data2" value="Shampoo" />Shampoo<br />
  <input type="radio" name="data2" value="Q-Tips" />Q-Tips<br />
  <input type="radio" name="data2" value="Soap"
        checked="checked" />Soap
<br /><br />
  <input type="submit" value="Submit Form" />
</form>
```

Form Pull-Down Menu

- To input data using a pull down menu, use the `<select>` tag with the "name" attribute set to a variable used in the processing script. Next, define the items in the pull down menu using the `<option>` tag. The closing `</option>` tag is required in XHTML.
- By default, the top item in the menu is selected, but you can set the default to any item in the menu by using the "selected" attribute inside one of the `<option>` tags contained within a `<select>` tag.

```
<form method="post"
      action="http://www.webyoda.com/cgi-bin/cgiecho/form.txt">
<table align="center" border="1" cellpadding="6">
<tr><td align="center" valign="top">
<font size="+2">Eyes:</font><br />
<select name="data1">
<option> Blue </option>
<option> Brown </option>
<option> Green </option>
<option> Hazel </option>
</select>
```

Form Scrollable Menu

- To input data using a scrollable menu, add the "size" attribute to the previous example. With "size="4", the first column doesn't scroll because it has only 4 items, the second scrolls since it has 5 items, and the third makes room for four items, but only displays two items on the menu. In this case, add the "selected" attribute if you wish to set a default value.

```
<form method="post"
      action="http://www.webyoda.com/cgi-bin/cgiecho/form.txt">
  <table align="center" border="1" cellpadding="6">
  <tr><td align="center" valign="top">
  <font size="+2">Eyes:</font><br />
  <select name="data1" size="4">
  <option selected="selected"> Blue </option>
  <option> Brown </option>
  <option> Green </option>
  <option> Hazel </option>
  </select>
```

Form Text Areas

- To input data using a text area (aka "edit window"), use the `<textarea>` tag. Place any default text for the edit window between the opening and closing `<textarea>` `</textarea>` tags. The "rows" attribute defines how many text rows tall the edit window is, while the "cols" attribute defines how wide it is.

```
<form method="post"
      action="http://www.webyoda.com/cgi-bin/cgiecho/form.txt">
  <font size="+1">Please enter your comments:</font><br />
  <textarea name="comments" rows="6" cols="50">
  This is where the default text goes.
  </textarea>
  <br />
  <br />
  <input type="submit" value="Submit Form" />
</form>
```

Form "Cancel" Button

- There is no defined "Cancel" feature in HTML, so we will improvise. To add a cancel button to any form, you make a separate form with the "method" attribute set to "get" and the "action" attribute pointing to an HTML file to be loaded when the cancel button is clicked. The cancel button is simply a submit button with the button text set to "Cancel Form."

```
<input type="submit" value="Submit Form" />  
<input type="reset" value="Clear Form" />  
</form>  
<form method="get" action="cancel.htm">  
<input type="submit" value="Cancel Form" />  
</form>  
</center>  
</body>  
</html>
```

Create a form using "mailto:"

- Using the "mailto:" protocol is the only way to send form information without a server script. A "mailto" compatible form is nothing more than a form where the "action" attribute points to the recipient of the mail.
- By default the "mailto" protocol sends data as one long line.
- This method of processing form data should be avoided because functionality is not consistent in all browsers.

```
<form method="post" action="mailto:sales@mortwiggy.com">
```

<button> tag

- You can also use the <button> tag to create your own image buttons and your own graphics.

```
<button type="submit" name="submit" value="submit">  
    
</button>
```

```
<button type="reset" name="reset" value="reset">  
    
</button>
```