HTML Form

In HTML, form tags can be used to interact with users. Things that can be added to a form are the likes of textboxes, radio buttons, checkboxes, drop down lists, text areas, and submit buttons. A basic HTML code for a form with a textbox and a Submit button looks like this:

```
<html>
<head>
<title> A BASIC HTML FORM </title>
</head>
<body>
 <form name="form1" method="" action="">
  <input type="text" value="username">
  <input type="submit" name="submit1"
value="login">
 </form>
</body>
</html>
```

The Form Tag

You don't have to tell your browser about any form elements on your web page. If you just want a simple form element like a text box, you can

insert the text box tag by itself. But if you want to do something with the information on your form, like send it somewhere or to someone, you have to "tell" your browser about the form elements on your page. You do this with the Form tag:

```
<form> </form>
```

Like most html tags, the form tag comes as a pair, the forward slash preceding the second form tag to indicate that the form tag ends. Any form elements you need then go between these two form tags.

A name attribute is a useful addition to the form tag. When the form has a name, you can then refer to it in scripts.

```
<form name="frmTest"> </form>
```

To give your form a name, type a space after form then type the word name, followed by an equals sign (=). Finally, add a name for your form.

You can call it anything you like. Here, we've called the form `frmTest`.

If you want your form to go somewhere or to someone, two other attributes are needed in the form tag: `method` and `action`. Like this:

```
<form name="frmTesst" method="post"
action="mailto:mryoung@presmaltra.com">
</form>
```

Method is the way to send the data. There are two options, `post` and `get`. POST sends the data in lines of text; GET squashes all the data in a single line and adds it to the URL in the Action part. If the URL was an internet address, you'd see all the data you're sending in the address bar of your browser. This sort of thing:

```
ProcessSurvey.html?text1=John&text2=Smith
```

The first name John was typed into the text box called "text1" and the surname Smith went into the textbox called "text2". Form elements are separated by an ampersand (&). That is a direct result of using the Get method to send

data. The POST method doesn't add all that code to the address bar of your browser.

It is most preferred to use POST method to send data rather than GET.

Action is used to specify the address where you want to send the data. Here, we're using an Email link to send the data straight to an email address:

```
action = "mailto:me@me.com"
```

But the form can be processed by a scripting language like CGI, ASP.NET, PHP, etc. In which case you'd specify the address of the script in question:

```
action = "processSurvey.php"
```

To ensure that data in your forms is not sent anywhere, you can add a pair of double quotes to the `action` attribute:

```
action = ""
```

This is good for testing purposes.

Form Elements

There are quite a few different form elements you can add to a form:

> textboxes, textareas, option buttons, check boxes, drop down list/fixed lists, password boxes, command buttons, submit buttons, reset buttons, labels, image command buttons, hidden form values.

HTML5, however, adds even more:

> placeholder, email, url, number, range, data/time, search, colour.

Some of these new form elements are only supported in certain browsers, however. For example, color refers to a popup colour picker. Only the Opera web browser supports this, at the time of writing.

Textboxes, Submit and Reset Buttons

For most form elements, the word INPUT is used to set up the element. Next, you type a space followed by the word TYPE. This tells the browser what type of form elements to draw on your page. If you want a text box, the TYPE to use is "text":

```
<input type = "text" />
```

Next you add the attributes you want. There are quite a lot of attributes for text boxes. The three most common ones are Size, Value, and Name. Size refers to how long the text box is in pixels rather than the number of characters it can contain. It's a good idea to limit many text characters can fit into your text box, and for this the maxlength attribute is used:

```
<input type="text" size="20" value= "" name =
"text1" maxlength="15"  />
```

The `value` attribute means the text that will be typed in your text box. You can set some default text, if you like:

```
<input type="text" size="20" value="default text here" name = "text1" />
```

Whatever is between the two quote marks of value what you get back when the form is sent somewhere.

A `name` attribute should be added so that you can refer to it in scripts. It distinguishes it from any other text box you may have on your form.

Some other textbox attributes are: accesskey, height, tabindex, width.

Height and width are self-explanatory. `accesskey` refers to a keyboard shortcut you can add, while `TabIndex` is a number value and refers to where the cursor will end up when you press the Tab key on your keyboard. As an example, take a look at the following three text boxes:

```
<input type="text" Tabindex="2" />
<input type="text" Tabindex="3" />
<input type="text" Tabindex="1" />
```

The third of the three text boxes above will be activated first when the tab key is pressed. Press the tab key again and the cursor will appear in the top textbox, followed by the middle one.

The Submit Button

If you want your form to be sent somewhere, a Submit button is needed (though you can write code for your own submit button). When the Submit button is clicked the browser will check the ACTION attribute of the form tag to see where you want the data sent. It then checks the method attribute to see what method you want to use, POST or GET. The browser will then try to send the form's data for you.

The code for a Submit button is this:

```
<input type="Submit" value="Submit" />
```

This time, the TYPE is set to "Submit". The `value` attribute is the text that will appear on the button itself. The width of the button is determined by the width of the text for the `value` attribute.

Here, spaces are added to the left and right of the text. The browser will see the spaces as text and adapt the width accordingly.

The Reset Button

The Reset button returns the form to the state it was originally in when it was loaded. Any default values you added will be retained. The code for a rest button is this:

```
<input type="Reset" value="Clear">
```

Note that the `type` is now "Reset". The value attribute works in the same way that the Submit button does - it's the text that will appear on the button.

Click to Ask Question(s)



MR YOUNG

IGWURUTA | ISIOKPO | ELELE | IKIRI | OMOKU

mryoung.presmaltra.com.ng