Reading 23: Forms

Note: you do not have to have a form in your template – you can simply have a link to an email address and they can email you.

1. How Forms Work

Forms are used on your site to allow you to gain information from visitors.

The most common forms are in the “contact us” pages of sites.

They operate in the following way:

1. A visitor to your site will see the form when it loads of their computer when they look at that page of your website

2. They complete the form and click on the “submit” button.

The information is then sent to the web server that hosts your website.

3. The hosting web server then runs a process that utilises a “computer language” called a “cgi script” that processes the information submitted on the form.

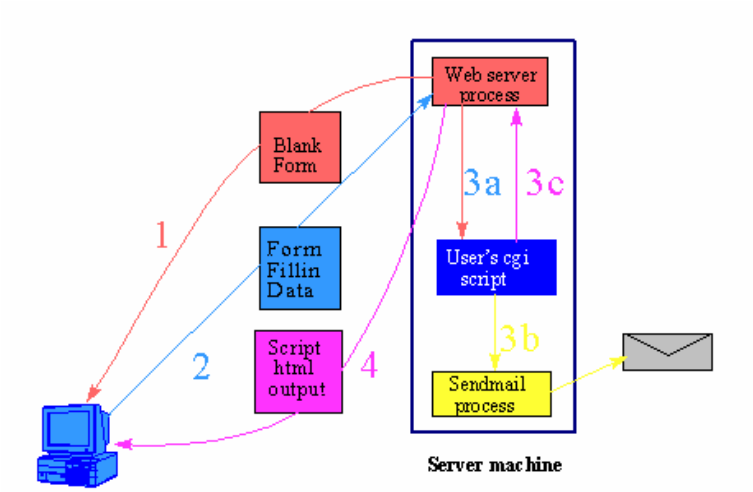
Note if you want a form – you need a hosting computer that supports “cgi script” or similar script languages.

The cgi script triggers steps 3a, 3b and 3c in the diagram below.

3A. The host computer “reads” the data submitted
3B. The host computer send you an email message that contains the data from the form submitted by the customer.
3C. The host computer then prepares an acknowledgement to the customer by way of a “thank you webpage” or similar. This is sent as step 4 below

4. The visitor should then receive a page informing them that the message has been sent.

Note – you determine what this webpage acknowledgement says – this is addressed later.



Source: http://www.indstate.edu/itdept/web/webformdoc.html

2. Creating a Form

Select a page on your website that you want the form to be on.

Then insert the form detail in a table you have set aside for the form.

Note – most templates have a form already in them as the contact us page.

The following instructions will show you how to edit the form to meet your requirements.

3. The Tags for Forms

FORM tag
You must start with a form tag. This tag has an opening and closing tag.

All of the form elements need to be between these opening and closing tags.

Tags:

<form>	opening tag
</form>	closing tag

4. Input Types on a Form

The form can have a number of different ways to get visitors to input data.

The most common are:

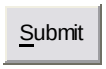
- Text a textbox allowing free text to be input
- Radio a circular "button" that is clicked on: ⊙ or off: ○.

Note a radio box will only allow ONE option to be selected

- Checkbox a square box that is clicked on: ☑ or off: ☐.

Note a checkbox will allow multiple choices to be selected

- Password a rectangular box area where a password is required
- Reset a button allowing users to clear or reset the form: 

- Submit the submit button: 

- Select box allows a choice to be chosen from the dropdown list
- Text area allows a preset size text area – a bigger version of a text box

Note that for most input types you will also have a label ie the text or instructions that the visitor to the site will see.

5. Generic Tag Format for the Input Types on the Form

INPUT TYPE tag
The INPUT TYPE tag is for the parts to the form where you will be getting users to input some sort of data.

The INPUT TYPE tag has an opening tag only. It does not need a closing tag except in XHTML - /INPUT>

We also give the input type a name – for internal website use only. For example when we get the email saying that the form has been submitted and what the details are – the name is used to show us what sections they responded to.

Note the “name” is NOT the same as the label or text that the visitor will see. We will add that shortly.

The basic format for the INPUT TYPE tag is:

```
“Label” <INPUT TYPE = “TYPE” NAME=”the name for this input”>
```

An example where the type of input is a text box that we want to call it “name_box” with the label that the visitors will see of “Name:”

```
Name: <INPUT TYPE = “TEXT” NAME=”name_box”>
```

Name:

If you want the label in the line above, simply add a line break

```
Name:<br> <INPUT TYPE = “TEXT” NAME=”name_box”>
```

Name:

The INPUT TYPE tag can have a number of different types of properties and values, for which we select the attribute or style we are after. For example, the size of the box or the colour ….

The basic format for the input type tags with properties is:

```
“Label” <INPUT TYPE = “TYPE” NAME=”the name for this input”
property1=”value”; property2=”value2” >
```

Details for how to use this format for the common input types for forms follow.

6. Tag Format for Text Boxes

Example: Name Text Box

Here is an example of a input tag setting up a text box that we have labelled “Name”

```
Name <INPUT TYPE="TEXT" NAME="name_box" SIZE="25" CLASS="shadeform">
```

In this example the INPUT tag has the following properties and values:

SIZE: How wide the textbox is
CLASS: You can use your “style sheet” to shade or colour your text box areas. In this example we would look up our css file and see the specific class (selector) called shadeform.

The following would be created with the above example:

Name:

Example: Template Code for Text Boxes
The following is the type of code that you will find in your template for text boxes for name, company, phone number and email. Note that a table has been created to fit the form details into:

```
<TABLE BORDER="0" cellpadding="2" cellspacing="0" width="100%">
<TR><TD VALIGN="TOP">
Name:<br>
<INPUT TYPE="TEXT" NAME="your_name" SIZE="30" CLASS="shadeform"><br>
</TD></TR>
<TR><TD>
Company:<br>
<INPUT TYPE="TEXT" NAME="company_name" SIZE="30" CLASS="shadeform"><br>
</TD></TR>
<TR><TD>
Phone:<br>
<INPUT TYPE="TEXT" NAME="phone" SIZE="30" CLASS="shadeform"><br>
</TD></TR>
<TR><TD>
E-Mail:<br>
<INPUT TYPE="TEXT" NAME="email" SIZE="30" CLASS="shadeform"><br>
</TD></TR>
<TR><TD>
Please Confirm E-Mail:<br>
<INPUT TYPE="TEXT" NAME="email_confirm" SIZE="30" CLASS="shadeform"><br>
```

7. Tag Format for Radio Buttons

Example: Yes or No Radio Button
Here is an example of an input tag setting up 2 radio buttons – one for yes and one for no. Note that the labels in this case are after each button, but they could have been put before.

```
Do You Agree? <input type="radio" name="agree" value="Yes"> Yes
<input type="radio" name="agree" value="No"> No
```

It would display as:

Do You Agree? ⭘ Yes ⭘ No

With radio buttons you get one choice only for each “name”. It is best for YES/NO, MALE/FEMALE type responses. People can only choose ONE option from the buttons provided/

Example: Male / Female Radio Button
Here is an example of an input tag setting up 2 radio buttons – one for male and one for female. Note that the labels in this case are before each button, but they could have been put after.

```
Male <input type="radio" name="sex" value="male">
Female<input type="radio" name="sex" value="female">
```

It would display as:

Male ⭘ Female ⭘

8. Tag Format for Check Boxes

Check boxes are useful where you want to allow people to pick more than one of the options you provide.

Example: Available Contact Times Checkbox
Here is an example of an input tag setting up a checkbox that allows visitors to tell you when they are available to be contacted by you – noting they can choose none, one, many or all of the options you provide.

```
Available contact hours: <br>
<input type="checkbox" name="contact" value="7am-9am”> 7am - 9am <br>
<input type="checkbox" name="contact" value="9am-5pm"> 9am – 5pm <br>
<input type="checkbox" name="contact" value="5am-8pm"> 5pm -8pm <br>
<input type="checkbox" name="contact" value="8am-10pm"> 8pm – 10pm <br>
```

This would display as:

Available contact hours:
☐ 7am - 9am
☐ 9am – 5pm
☐ 5pm – 8pm
☐ 8pm - 10pm

With checkboxes you get multiple choices. It is best for when you want to quickly gather a large range of responses.

9. Tag Format for Password

A password is different to a text box simply because asterisks will appear when a person types in their password rather than the letters they are actually typing. It is therefore useful to use this for any information that you don’t want to be visible on their screen when you are entering it.

Note this system of asterixes does NOT protect their password or data. You would require SSL secure socket layer protection to add protection to any submitted data.

Example: Password Tags
Here is an example of an input tag setting up a password.

```
<input type="password" name="password" SIZE="10” >
```

It would display as:

Password: *****

Note the size property refers to maximum characters in the box that displays NOT the password length.

10. Tag Format for Text Area or Comments Boxes

There are situations where a textbox will not be large enough for the information you want.

In this case a textarea input type is used. In this situation you define the number of rows and size of the columns for the text area.

Note the input tag is slightly different and requires a close tag.

Example Comments Section
Here is an example of an input tag setting up a text area that allows visitors to leave you a lengthy comment.

```
Please provide your comments:<br>
<TEXTAREA NAME="comments" ROWS="6" COLS="35" </TEXTAREA>
```

The following would display:

Please provide your comments:

Example Template Comments Section
Your template will have the following comments section code, using a class in the css file to further define attributes:

```
Please provide your comments:<br>
TEXTAREA NAME="comments" ROWS="6" COLS="35"
CLASS="textarea"></TEXTAREA>
```

11. Tag Format for Select Boxes

There are times when you want to have a list that you have visitors select from. In this case you create a select box and the range of options they can choose.

Note the different tags, and the option tags and close tags:

Example Select Box for Choosing State

Here is an example of a select box used to let people select the state they live in. Note the use of a “None selected” value:

```
<select name="states">
<option selected value="None selected">Please Select State ----></option>
<option value="NSW">NSW</option>
<option value="QLD">QLD</option>
<option value="VIC">VIC</option>
<option value="NT">NT</option>
<option value="ACT">ACT</option>
<option value="WA">WA</option>
<option value="TAS">TAS</option>
</select>
```

Example Select Box for Choosing State with a Default Selection

If you wanted a default state to show in the list use the following bold code for the one you want as the default, in this case NSW:

```
<select name="states">
<option value="NSW"> selected="selected">NSW</option>
<option value="QLD">QLD</option>
<option value="VIC">VIC</option>
<option value="NT">NT</option>
<option value="ACT">ACT</option>
<option value="WA">WA</option>
<option value="TAS">TAS</option>
```

Example Select Box from Templates – How You Found Us

The following is an extract showing the typical select box used in the templates, where a class in the css file is used to define aspects eg the dropdown call:

```
<select name="through" class="dropdown">
<option selected value="None selected">Please Select One ----></option>
<option value="Google">Google</option>
<option value="Friend">Friend</option>
<option value="website">Website</option>
<option value="Magazine Newspaper">Magazine/Newspaper</option>
<option value="other">Other</option>
</select> <B>Found Us Where?</B><BR>
```

12. Tag Format for Submit Buttons

There are common buttons you may have in your forms. The most common one is to submit the data / form.

When you click on the button it looks and finds the ACTION for that form (we will cover this later) and does what it says.

The following is the code for creating the buttons NOT the action. This is addressed later.

Note that the value is the word/s you want inside the button. The input type is the type of button, ie a submit button.

Example: Submit Button
Here is an example of an input tag setting up a submit button.

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

The following button will display:

Example: Template Submit Button
The templates use a slightly different coding for submit buttons, again using a class in the css file.

```
<input type="submit" value="Submit Query" class="submitbutton"> 
```

You will notice outside the tag for the submit button is the code
This creates a “non-line breaking space” ie a small gap eg like what a spacebar does in a word document. This is used to keep 2 buttons apart.

13. Tag Format for Reset Buttons

Example: Reset Button
Here is an example of an input tag setting up a reset /clear button. The type of button: reset is a “code” that creates the action of clearing the form fields.

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

The following button will display:

Example: Template Submit Button
The templates use a slightly different coding for submit buttons, again using a class in the css file. This time the words that will appear in the button are “Clear Form”:

```
<input type="reset" value="Clear Form" class="submitbutton">
```

14. “CGI” Code for Accepting the Form Details

Forms need direction for what it is to do. This is steps 3 & 4 in your diagram in the first section of this handout.

The form directions are supplied by the format of METHOD and the ACTION.

The METHOD specifies what the form will do, the ACTION how it will do it.

NOTE: that the detail of how this is done is determined by your web hosting provider. You need to edit your template to change / add the code that they require.

The following is a generic example of the most common form action scripts which appear at the beginning of forms:

```
<form method="POST" action="http://your-web-domain.com/cgi-bin/formmail.cgi"
target="_top">
```

In this form the **METHOD** used is POST. This is the most common method and will be used for 90% of all forms.

The **ACTION** is telling the form to do go to a particular web server and run a specific program file – in this case a file called formmail.cgi

The location of the web server is specified in the section http://your-web-domain.com.

It is then telling it to go to a specific folder called the “cgi-bin”

Then to access the file, or in this case to run a small program or script called “formmail.cgi”.

Most hosting companies will have a CGI (Common Gateway Interface) folder and most will also have a script that handles forms.

Note it may be called formmail.cgi, or mailform.cgi or webform.cgi. Other names are common as well.

Check with your hosting provider for the path (folder) and name of the file that they use.

The file or script acts to accept the form, and then do what you have told it to do (often in your “Console” area that you set up with your web hoster eg it will send a message to your site to have a “thanks.htm” or “thankyou.htm” page displayed.

It will also advise you of the data in the form. It can do this multiple ways. The most common is to either send you an email containing the information from the form. If you have your form linked to a database, it can add the data to the database (subject to your selected hosting service supporting the database.

Source: http://www.w3schools.com/html/html_forms.asp

Updating Your Template for Form Actions

To update your template – first of all update the contact page in notepad with the correct domain name and file path and file ie the code in bold below:

<form method=POST action="**http://your-domain.com/cgi-bin/formmail.cgi**" target="_top">

Ensure that you check this detail with the requirements of your web hosting business.

Secondly update the email address that you want the message sent to. This is the text in bold:

<INPUT TYPE="HIDDEN" NAME="recipient" VALUE="**info@your-domain.com**">

Many businesses use a specific email address for web enquiries eg contact@your-domain.com or enquiries@your-domain.com

Ensure that you have created this email address in the console provided by your web hoster AND have an email account on your computer that picks up these emails.

Step 3 – specify the subject heading that will be used when your web server creates and forwards the email to you:

<INPUT TYPE="HIDDEN" NAME="subject" VALUE="**WebSite Contact**">

Make the value to any words you would like to appear in the email subject line.

Step 4 specify the web page that you want sent back to the visitor so that they know the form was received. In most templates you have a page called thanks.htm to be used for this purpose.

<INPUT TYPE="HIDDEN" NAME="redirect" VALUE="http://**your-domain.com/thanks.htm">**

Update the name of your domain name and the name of the return page if it isn't called thanks.htm.

Ensure that you have edited the "thanks.htm" page so that it says what you want it to say when it goes back to the visitor.

Step 5 – captures the data that the sender has submitted. At this stage the email address and their name that the sender submitted as part of the form. Note that you must have these “parts” in your form.

Most cgi scripts will use this detail in the FROM section of the email, so you can capture and save their email address and name:

```
<INPUT TYPE="HIDDEN" NAME="required" VALUE="email,your_name">
```

In the above example you are seeking the data from the input properties that you named as “email” and as “your_name”.

The following shows the template code used in the form that named these sections:

```
Name:<br>
<INPUT TYPE="TEXT" NAME="your_name" SIZE="30" CLASS="shadeform"><br>
</TD></TR>
<TR><TD>
Company:<br>
<INPUT TYPE="TEXT" NAME="company_name" SIZE="30" CLASS="shadeform"><br>
</TD></TR>
<TR><TD>
Phone:<br>
<INPUT TYPE="TEXT" NAME="phone" SIZE="30" CLASS="shadeform"><br>
</TD></TR>
<TR><TD>
E-Mail:<br>
<INPUT TYPE="TEXT" NAME="email" SIZE="30" CLASS="shadeform"><br>
</TD></TR>
<TR><TD>
Please Confirm E-Mail:<br>
<INPUT TYPE="TEXT" NAME="email_confirm" SIZE="30" CLASS="shadeform"><br>
```

Step 6 gets the rest of the data that you want from their form, in the format that you specify

```
<input type=hidden name="sort"
value="through,your_name,phone,company_name,email,comments">
```

Note – ensure that the values are updated to reflect the names of the parts to your form.