07 Forms

1 About Forms

For a website to be successful, it is important to be able to get feedback from visitors to your site. This could be a request for information, general comments on your site or even a product order. This could be done with a `MAILTO:` link but providing a form has several advantages over a simple email.

- It makes it easier for people to send the information
- A form gives you greater control over the information that is sent.
- Form results can be organised in a way that makes them easy to store in a spreadsheet or database.

1.1 Form Handlers

When a form is filled in by a visitor to your site and sent, the results of the form need to be processed in some way. One of the most common ways to process form results is with a CGI (Common Gateway Interface) script. This is a small program that the information entered in to the form is sent to as soon as the form is submitted. The purpose of this script is to accept the results of a form, organise the results and send the results to an appropriate location, such as a text file, database or email address. In these exercises we won't go into creating CGI scripts since that is quite different from HTML and requires some programming knowledge. There are many free CGI scripts on the Internet that you can download and use for your own forms.[1] Many ISPs (Internet Service Providers) also provide scripts for the use of their users so you can often get by without having to learn how to create your own scripts.
In addition to CGI there are some other technologies available for linking forms with databases such as Microsoft's Active Server Pages (ASP), PHP and Allaire's Cold Fusion.

2 The FORM Tag

Forms are placed in your document using a `<FORM>` tag which must have a closing `</FORM>` tag. You can have more than one form in a document as long as they don't overlap. I.e. one form must finish before the next one begins. A `<FORM>` tag specifies two main things.

- The location of the program or script used to process the results of the form and send them to an appropriate location
- The method that will be used to send data from the form

The layout of the form is specified by the form fields. These can be placed anywhere between the `<FORM>` and `</FORM>` tags.

A `<FORM>` tag usually contains the following attributes.

`ACTION=` Specifies the location of a program to process the results of the form. An email address can also be specified, though this is only supported by some browsers and can give you messy results. Eg.
`ACTION="MAILTO:email@address.com"` would send the form results to this address.

`METHOD=` get / post - Specifies the method used to submit the form results. *Get* for search forms and forms where results need to be retrieved. *Post* for feedback forms and forms where results need to be sent.

`NAME=` Name that can identify the form (E.g. in scripts).

[1] Matt's Script Archive is a good place to start - http://www.worldwidemart.com/scripts/

3 Form Elements

Your form may contain several elements that enable a user to input information. Each of these is described in the following sections. All of them use the `<INPUT>` tag except for select lists and text areas, which use `<SELECT>` and `<TEXTAREA>` respectively. For the `<INPUT>` elements, a `TYPE` attribute is used to specify what type of element it is.

3.1 Input fields

Input fields can have the following attributes. The attributes required will depend on the type of input being used – specified by the `TYPE` attribute.

`ALIGN=` If the input type is an image, this is used the same as for an `IMG` tag.

`CHECKED` For checkbox and radio input types, this attribute will cause the option to be pre-selected by default.

`MAXLENGTH=` Determines the maximum number of characters that can be typed into this field.

`NAME=` Provides a name for the field. This is used to identify the contents of this field in the form results.

`SIZE=` Sets the width of the field in characters.

`SRC=` If the input type is an image, this specifies the location of the image file the same as the `SRC` attribute in an `<IMG>` tag.

`TYPE=` Specifies the type of the input field. The choices are text, password, checkbox, radio, submit, image, reset, file and hidden.

`VALUE=` For text fields, an optional attribute that sets an initial value for the field (if you want it to already be filled in).

For check boxes and radio buttons, this specifies what to send with the form results if the option is selected.

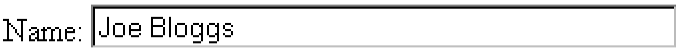
Each of the various input types is described below.

TYPE="text"

A text input element is for entering a small amount of text. It uses an <INPUT> tag with a TYPE="text" attribute. It uses the NAME=, SIZE=, MAXLENGTH= and VALUE= attributes.

3.2 Example

Name: <INPUT TYPE="text" NAME="name" SIZE="40" MAXLENGTH="80" VALUE="Joe Bloggs">



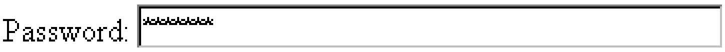
Name: Joe Bloggs

TYPE="password"

This is the same as a text field except that any characters entered will appear as * as they are typed. It has all the same attributes as a text input field except that there is no VALUE attribute.

3.3 Example

Password: <INPUT TYPE="password" NAME="password" SIZE="40" MAXLENGTH="80">



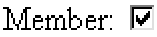
Password: ********

TYPE="checkbox"

A checkbox input can be used for boolean fields where there are only two choices. It can use the NAME=, VALUE= and CHECKED attributes. When several checkbox inputs share the same name, their results will be put into the same field. This allows users to select more than one value for a category.

3.4 Example

```
Member: <INPUT TYPE="checkbox" CHECKED NAME="member" VALUE="yes">
```



Member: ☑

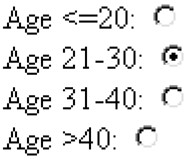
TYPE="radio"

Radio inputs allow a user to select from several options where only one can be selected. Each option in a list has its own <INPUT> tag and each must have the same name. One option can be pre-selected with the CHECKED attribute.

Example

Note that the &lt; and &gt; special characters have been used to display the less than < and greater than > symbol.

```
<BR>Age &lt;=20: <INPUT TYPE="radio" NAME="age" VALUE="20">
<BR>Age 21-30: <INPUT TYPE="radio" NAME="age" VALUE="21-30" CHECKED>
<BR>Age 31-40: <INPUT TYPE="radio" NAME="age" VALUE="31-40">
<BR>Age &gt;40: <INPUT TYPE="radio" NAME="age" VALUE="40">
```



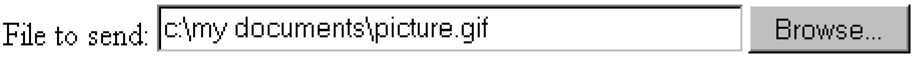
Age <=20: ○
Age 21-30: ●
Age 31-40: ○
Age >40: ○

TYPE="file"

This field allows a user to specify the name of a file to be sent as an attachment with the form results. Normally a browse button will appear next to the field to allow the user to browse for the location of the file on their computer. The NAME=, SIZE= and MAXLENGTH= attributes may be used. File inputs are not widely supported in browsers.

3.5 Example

```
File to send: <INPUT TYPE="file" NAME="file" SIZE="40">
```



File to send: c:\my documents\picture.gif

TYPE="hidden"

Some CGI scripts make use of hidden fields within the form to accept additional parameter information (such as an email address to send the results to or a subject for the email). These are passed to the server when the form is submitted. Normally they will contain a NAME= and VALUE= attribute.

3.6 Example

```
<INPUT TYPE="hidden" NAME="subject" VALUE="feedback">
```

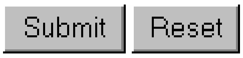
This might be used to provide a subject for the form results when they are sent to an email address. Forms may also have hidden fields to do things like specify an email address to send the results to or specify an html file to display once the results are submitted (confirmation page).

TYPE="submit" and TYPE="reset"

These inputs both provide buttons that affect the results of the form. Both have a VALUE= attribute which determines the text to appear on the button. A submit button can also have a NAME= attribute. When a user clicks a submit button, the results of the form will be sent. When a user clicks a reset button, the contents of the form’s fields are set to their initial state.

Example

```
<INPUT TYPE="SUBMIT" NAME="Submit" VALUE="Submit">
<INPUT TYPE="RESET" NAME="Reset" VALUE="Reset">
```

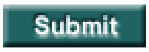


TYPE="image"

This type of input can be used in place of a submit button. Instead of a button, it will show an image, which will submit the form results when it is clicked. This type of input typically has a NAME= and SRC= attribute but it can also include attributes common to IMG tags such as ALIGN=. Although an image can look a lot better than a plain submit button, image inputs can cause difficulties with text browsers.

Example

```
<INPUT TYPE="image" NAME="Submit" SRC="submit.gif" WIDTH="70" HEIGHT="20">
```



Note Netscape Navigator 4 will typically treat an image button as an image link by putting a blue border around the image. To prevent this, include BORDER=”0” as you would for an <IMG> tag.

Tip You can’t use an image in place of a reset button but there is a workaround. Have an image that is a link to the page the form is on. When a user clicks the image, it will cause the browser to reload the page, which in turn clears the form.

3.8 Select Menus

As the name implies, select menus allow a user to select from a list of options. The menu begins with a `<SELECT>` tag and ends with a `</SELECT>` tag. The select tag has the following attributes.

`MULTIPLE` This attribute enables more than one option to be selected by holding down the **[CTRL]** key, which is useful for certain types of forms. Particularly when used with an online database.

`NAME=` Name of the list that will be used in the results to identify the field.

`SIZE=` Number of rows to appear in the list. The default is 1, which will mean that one option will be visible but an arrow will appear to the right. This arrow will bring down a list when clicked. If values of more than 1 are set, the list will show that number of rows with a scroll bar on the side.

Within the select tag, each item in the list is set with an `<OPTION>` tag. An option tag may have the following attributes.

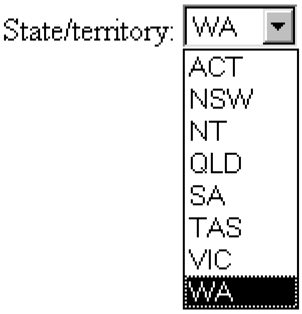
`SELECTED` One option in the list can have this attribute, which will mean that particular option will be initially selected.

`VALUE=` Value that will be submitted with the form. If no value is specified, the text in the list itself will be used.

Example 1

One row displayed with a dropdown list.

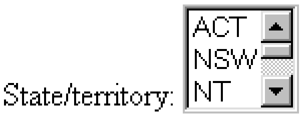
```
<SELECT NAME="state">
      <OPTION>ACT
      <OPTION>NSW
      <OPTION>NT
      <OPTION>QLD
      <OPTION>SA
      <OPTION>TAS
      <OPTION>VIC
      <OPTION SELECTED>WA
</SELECT>
```



Example 2

Three rows displayed with a scrollbar on the side.

```
<SELECT NAME="state" Size="3">
      <OPTION>ACT
      <OPTION>NSW
      <OPTION>NT
      <OPTION>QLD
      <OPTION>SA
      <OPTION>TAS
      <OPTION>VIC
      <OPTION>WA
</SELECT>
```



3.9 Text Areas

Text areas are used where a large amount of text needs to be entered. Common examples are where you want to provide a space on the form for comments to be entered. Text areas begin with a `<TEXTAREA>` tag and end with a `</TEXTAREA>` tag. Anything between the start and end tag will appear in the field on the page. If there is nothing between the start and end tag, the field will be blank to start with. Text areas can have the following attributes.

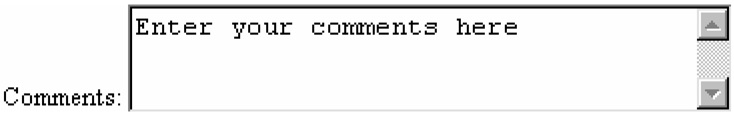
`COLS=` Width of the field in characters. The width of a text area will be quite different in Internet Explorer and Netscape Navigator so don't count on getting them consistent.

`ROWS=` Number of rows that will appear.

`NAME=` Name of the text area that will be used in the results to identify the field.

3.10 Example

```
Comments:
<TEXTAREA NAME="comments" ROWS=3 COLS=35>
Enter your comments here
</TEXTAREA>
```

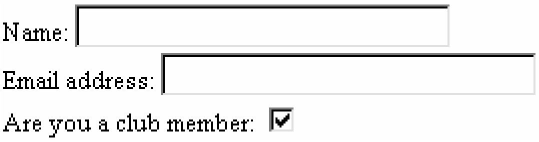


Comments: Enter your comments here

3.11 Form Layout

A common technique with form layout is to place the form components in a transparent table so everything lines up neatly. Below are examples of a form created so each field is placed one under the other with `<P>` tags and a form that uses a table to line up the form components. In our exercise we will use an existing table to line up the various fields in our form.

3.11.1 Without a table

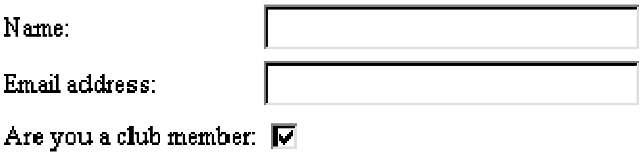


Name:

Email address:

Are you a club member: ☑

3.11.2 With a table



Name:	
Email address:	
Are you a club member: ☑	

Exercise 5 Creating an Online Form

In this exercise we will create a form to be used to send feedback about the site.

1 Open the file feedback.html. Look at the HTML and preview the document. To save time, a table has already been prepared for the form. Tables can be a good way to line up the fields in a form.

2 The first thing we will do is insert the <FORM> tag before the beginning of the table. We will also insert some hidden fields that will be used by the form handler. Your instructor will tell you what to put in the form tag and what hidden fields to use. An example is shown below. These should be placed before the <TABLE BORDER="0"> tag.

```
<FORM NAME="feedback" ACTION="http://www.iinet.net.au/bin/mail" METHOD="POST">
<INPUT TYPE="hidden" NAME="subject" VALUE="feedback">
<INPUT TYPE="hidden" NAME="destination" VALUE="cougar@footy.net.au">
```

(You will need to change the email address to one that you can check to test the form results)

3 Next, add a </FORM> tag after the end of the table (after the </TABLE> tag).

4 We will now add the form components to the table in the appropriate places. Edit the HTML so it looks similar to the HTML below. The modified HTML is highlighted in bold text.

```
<TABLE BORDER="0" ALIGN="center">
<TR><TD>Name: </TD>
<TD><INPUT TYPE="text" NAME="Name" SIZE="40" MAXLENGTH="80"></TD></TR>

<TR><TD>Email address:</TD>
<TD><INPUT TYPE="text" NAME="Email" SIZE="40" MAXLENGTH="80"></TD></TR>

<TR><TD>Are you a club member:</TD>
<TD><INPUT TYPE="checkbox" CHECKED NAME="Member" VALUE="yes"></TD></TR>

<TR><TD VALIGN="top">Your age:</TD>
<TD><INPUT TYPE="radio" NAME="Age" VALUE="20"> &lt;= 20
<BR><INPUT TYPE="radio" NAME="Age" VALUE="21-30" CHECKED> 21-30
<BR><INPUT TYPE="radio" NAME="Age" VALUE="31-40"> 31-40
<BR><INPUT TYPE="radio" NAME="Age" VALUE="40"> &gt; 40
</TD></TR>

<TR><TD>Your state:</TD>
<TD>
<SELECT NAME="state">
<OPTION VALUE=”act”>Australian Capital Territory
<OPTION VALUE=”nsw”>New South Wales
<OPTION VALUE=”nt”>Northern Territory
<OPTION VALUE=”qld”>Queensland
<OPTION VALUE=”sa”>South Australia
<OPTION VALUE=”tas”>Tasmania
<OPTION VALUE=”vic”>Victoria
<OPTION VALUE=”wa” SELECTED>Western Australia
</SELECT>
</TD></TR>
```

Continued on the next page

```
<TR><TD VALIGN="top">How did you<BR>find our site:</TD>
<TD>
<SELECT NAME="found" SIZE="3">
<OPTION>Web Search
<OPTION>Link on another site
<OPTION>Told about it
<OPTION>Other
</SELECT></TD></TR>

<TR><TD VALIGN="top">Site suggestions:</TD>
<TD><TEXTAREA NAME="suggestion" ROWS=2 COLS=35>Type your suggestion here
</TEXTAREA></TD></TR>

<TR><TD VALIGN="top">Other comments:</TD>
<TD><TEXTAREA NAME="comment" ROWS=2 COLS=35>Additional comments here
</TEXTAREA></TD></TR>

<TR><TD COLSPAN="2" ALIGN="center">
<INPUT TYPE="SUBMIT" NAME="Submit" VALUE="Submit">
<INPUT TYPE="RESET" NAME="Reset" VALUE="Reset"></TD></TR>
</TABLE>
```

5 Save and preview the file. Fill in some of the fields and then test the reset button.

6 We’ll finish by replacing our buttons with graphics. Find the lines with the buttons and change them as shown below.

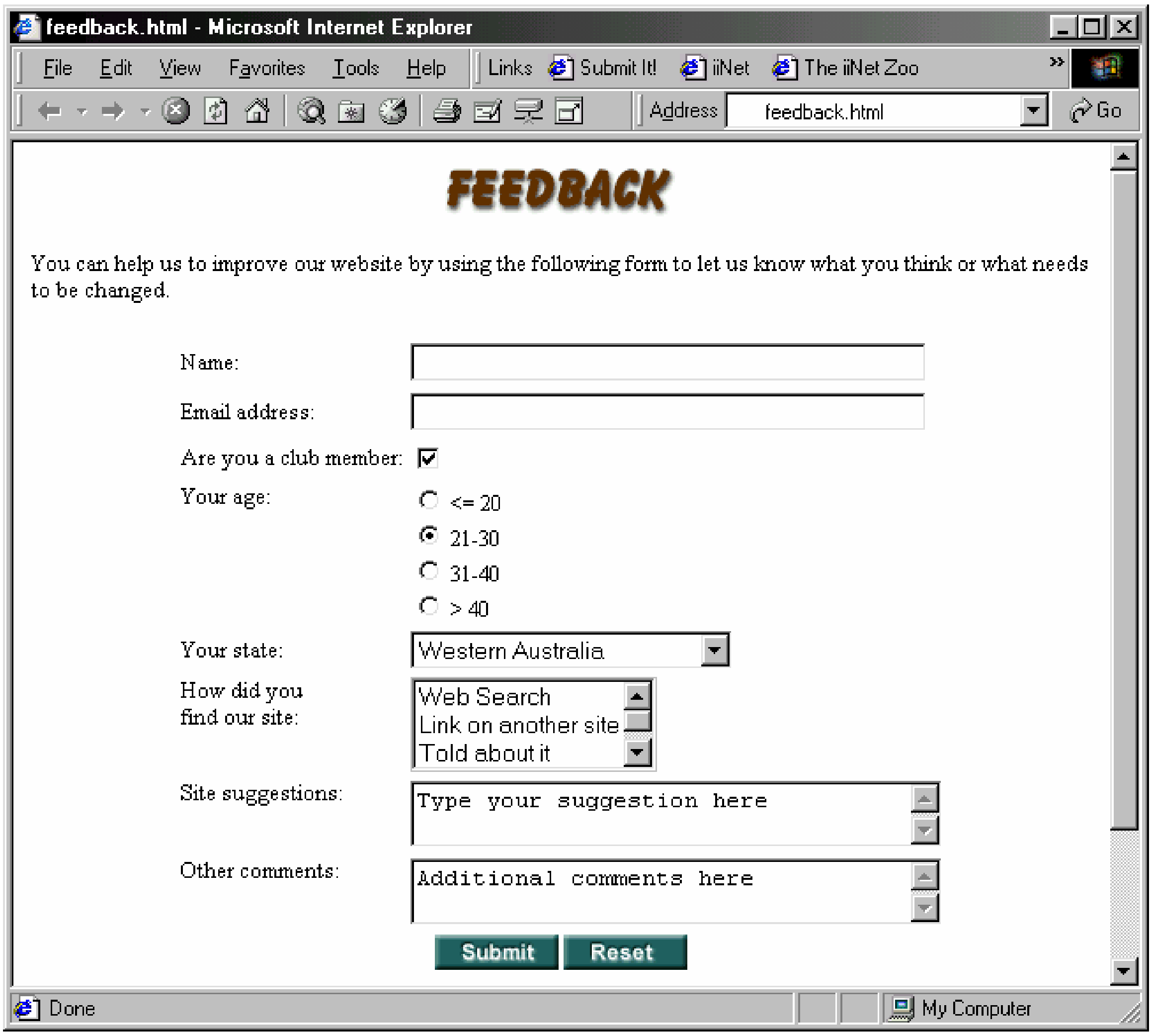
7 Find the following lines of HTML.

```
<TR><TD COLSPAN="2" ALIGN="center">
<INPUT TYPE="SUBMIT" NAME="Submit" VALUE="Submit">
<INPUT TYPE="RESET" NAME="Reset" VALUE="Reset"></TD></TR>
</TABLE>
```

8 Change them as follows

```
<TR><TD COLSPAN="2" ALIGN="center">
<INPUT TYPE="image" NAME="Submit" SRC="pics/submit.gif" WIDTH="70" HEIGHT="20"
BORDER="0">
<A HREF="feedback.html"><IMG SRC="pics/reset.gif" WIDTH="70" HEIGHT="20" ALT="Reset"
BORDER="0"></A></TD></TR>
</TABLE>
```

Your form should look similar to the one on the following page.



FEEDBACK

You can help us to improve our website by using the following form to let us know what you think or what needs to be changed.

Name:	
Email address:	
Are you a club member:	☑
Your age:	○ <= 20 ● 21-30 ○ 31-40 ○ > 40
Your state:	Western Australia
How did you find our site:	Web Search Link on another site Told about it
Site suggestions:	Type your suggestion here
Other comments:	Additional comments here
	Submit Reset

Exercise continued on the next page.

9 If time permits, your instructor may guide you through the steps of testing the form and viewing the results. The results for this form should be sent as an email and look similar to the results shown below.

```
This is the output of a form completed on
the World Wide Web.

name:    Joe Bloggs
member:    yes
age:    31-40
state:    Western Australia
found:    Told about it
suggestion:    Needs plenty of testing
comment:    Coming along nicely
Submit.x:    16
Submit.y:    11

The following information is provided by the server, and
may be useful for statistical purposes:

  User Agent:    Mozilla/4.0 (compatible; MSIE 5.5; Windows 98)
  Protocol:      HTTP/1.0
  Peer:          retro.iinet.net.au (203.59.24.149)
  Performed at:  Sat Jan 29 18:21:52 1999
  Handled by:    iiNet Technologies, http://www.iinet.net.au/
```

10 Open the file navbar.html. Add a new row to your table so we can place a “feedback” link between “Contact Us” and “Links”. The HTML for the new row should look similar to the HTML below. (Hint: copy and paste one of the other rows and then modify it)

```
<TR><TD><A HREF="feedback.html" TARGET="contents">
<IMG SRC="pics/nav_feedback.gif" ALT="Feedback" WIDTH="110" HEIGHT="25" BORDER="0"
HSPACE="5"></A></TD></TR>
```

11 Save the file. Open and preview frames.html.

Chapter 4 Forms Revision Questions

1. What are some uses for an online form?

2. How many forms can you have in a page?

3. What happens to the results of a form when it’s sent?