

# Further web design: HTML forms

## Practical workbook

### **Aims and Learning Objectives**

The aim of this document is to introduce HTML forms.

By the end of this course you will be able to:

- use existing forms on the web;
- create a simple form;
- add extra code to make forms more accessible and usable.

## Document information


### Course documents and files

If you are working in the training room the files you need are in **C:\User\WWW\Web Design 3\Resources\forms**.

If you want to work through the practical tasks shown in this document on your own system, obtain the necessary files from:

[www.bristol.ac.uk/is/learning/documentation/docs-current.html#web-t17](http://www.bristol.ac.uk/is/learning/documentation/docs-current.html#web-t17)

You will see a link to Practice files. This contains the files you need to work through the course notes. In Internet Explorer you extract the files as follows:

- Click on **Practice files**;
- Choose **Save**.
- Browse to where you want to save the file and click on **Save** (the practice files are grouped together, and you need to extract them to be able to use the individual files);
- Go to the file you have just saved, and **Open** or **Run** it to show the **Self-extracting Archive** box;
- Select where to extract (save) the files on your system using the browse button (as shown  opposite);
- Click **Start** to save the files.

### Related documentation

Other related documents are available from the web at:

Further web design: Cascading style sheets (document number web-t15)

[www.bristol.ac.uk/is/learning/documentation/docs-current.html#web-t15](http://www.bristol.ac.uk/is/learning/documentation/docs-current.html#web-t15)

Further web design: Meta tags (document number web-t16)

[www.bristol.ac.uk/is/learning/documentation/docs-current.html#web-t16](http://www.bristol.ac.uk/is/learning/documentation/docs-current.html#web-t16)

---

Further web design: HTML forms (August 06)

[www.bristol.ac.uk/is/learning/documentation/web-t17/web-t17.doc](http://www.bristol.ac.uk/is/learning/documentation/web-t17/web-t17.doc)

If you have any comments or queries about this document mail [iser-docs@bristol.ac.uk](mailto:iser-docs@bristol.ac.uk).

This document may be used wholly or in part by the academic community, providing suitable acknowledgment is made. It may not be used for any commercial or profit-making purpose without permission. © 2006 University of Bristol. All rights reserved.

## **Introduction**

This course module aims to extend participants' knowledge of HTML to include the use and construction of forms within web pages. Participants are also shown how to add code to increase form accessibility and usability.

## **Prerequisites**

This document assumes that you are familiar with the use of a computer keyboard and mouse, Microsoft Windows software including file management, and a web browser.

You should also be able to create simple web pages using a text editor and be able to transfer files to a web server, having attended a relevant course, eg Introduction to Web page creation in XHTML (code CC1WPC).

# Contents

## Document information

<b>Task 1</b>	<b>Using an HTML form .....</b>	<b>1</b>
<b>Task 2</b>	<b>Writing a simple form.....</b>	<b>3</b>
<b>Task 3</b>	<b>Including selection boxes and buttons .....</b>	<b>5</b>
<b>Task 4</b>	<b>Text input .....</b>	<b>6</b>
<b>Task 5</b>	<b>Creating menus .....</b>	<b>7</b>
<b>Task 6</b>	<b>Making forms more accessible .....</b>	<b>8</b>
<b>Appendix A</b>	<b>How do I put a form on the web? .....</b>	<b>10</b>

## Task 1 Using an HTML form

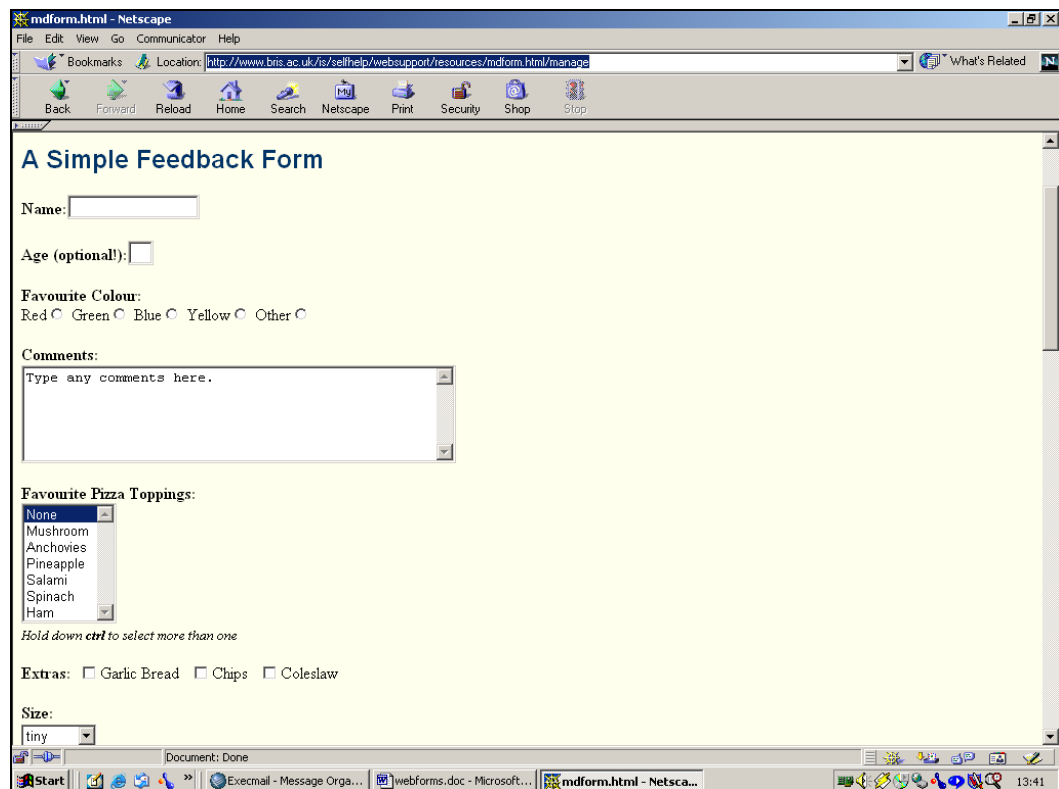
**Objectives** To introduce the use of forms in web pages and identify corresponding HTML code.

**Comments** This form uses all of the simple form boxes, buttons and input elements.

1.1 Open the following URL in your web browser:

[www.bristol.ac.uk/is/info/websupport/resources/mdform.html](http://www.bristol.ac.uk/is/info/websupport/resources/mdform.html)

The following form will be displayed:



The screenshot shows a Netscape browser window with the address bar containing the URL <http://www.bristol.ac.uk/is/info/websupport/resources/mdform.html>. The browser interface includes a menu bar (File, Edit, View, Go, Communicator, Help), a toolbar with buttons for Back, Forward, Reload, Home, Search, Netscape, Print, Security, Shop, and Stop, and a status bar at the bottom showing 'Document: Done' and the time '13:41'. The main content area displays a form titled 'A Simple Feedback Form' with the following elements:

- Name:** A text input field.
- Age (optional):** A text input field.
- Favourite Colour:** Radio buttons for Red, Green, Blue, Yellow, and Other.
- Comments:** A large text area with the placeholder text 'Type any comments here.'
- Favourite Pizza Toppings:** A dropdown menu with options: None, Mushroom, Anchovies, Pineapple, Salami, Spinach, and Ham.
- Extras:** Checkboxes for Garlic Bread, Chips, and Coleslaw.
- Size:** A dropdown menu with the option 'tiny' selected.

Figure 1 - simple feedback form

1.2 Enter some data into the form, using the **Reset** button to start again, if required.

Notice that the form results are returned in the form of a web page. Note the URL:

[www.bristol.ac.uk/cgi-bin/test\\_form.pl](http://www.bristol.ac.uk/cgi-bin/test_form.pl).

This is the URL of the **CGI (Common Gateway Interface)** script used to parse (process) the form. The **.pl** file extension tells us that the script was written in **PERL (Practical Extraction and Reporting Language)**, the most commonly used language for such applications. A typical script will email the form results to an address specified within the form or script. The above example replicates the content of such an email message.

**Note** Unlike other form elements, the **Reset** button has a predetermined function: to clear a form.

1.3 Minimise the form results page and, on the original form page, scroll down to the source code used to write the form.

- Examine the code and try to identify the functions of the different HTML form elements. For example, how does a check box differ from a radio button?
- 1.4** Maximise the form results page and resize the window, so you can view the form results and HTML code simultaneously. Try to identify the roles of the HTML elements which appear on the results page.

## Task 2 Writing a simple form

**Objectives** To write a simple HTML input form and link it to a CGI script in order to return the results as a web page.

**Comments** You will notice that some of the elements include attributes that do not appear in the form itself, but in the results page.

---

**2.1** Open the form template file:

- To open Notepad, from the **Start** menu select **Programs / Accessories** and click on **Notepad**.
- Browse to folder: **C:\User\WWW\Web Design 2\** and open file **formtemplate.html**

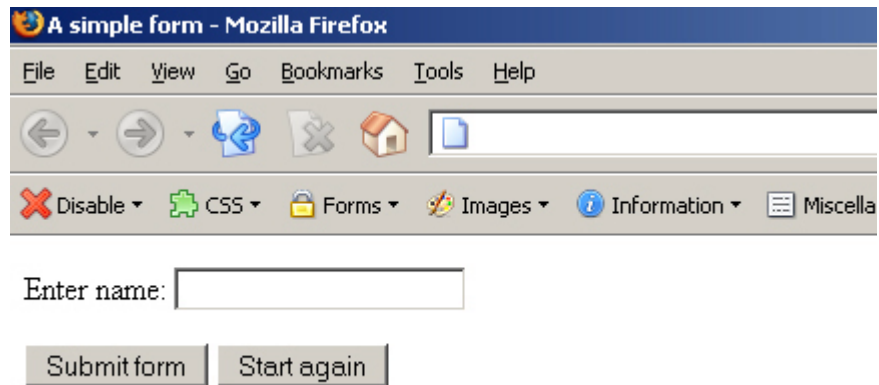
**2.2** Type in the following code, shown in bold italic:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<HTML xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>A simple form</title>
</head>
<body>
<form method="post" action="http://www.bristol.ac.uk/cgi-bin/test_form.pl"
target="_blank">
<p>
Enter name: <input type="text" name="MyName1" />
</p>
<p>
<input type="submit" value="Submit form" />
<input type="reset" value="Start again" />
</p>
</form>
</body>
</HTML>
```

- Save this file as **form.html** in your working area: **C:\User\WWW\Web Design 2\**.

**2.3** Open the file in your web browser: **File / Open File** (in Firefox ), or **File / Open / Browse** (in Internet Explorer).

- You should see the following: (see over)



**Figure 2 - creating a simple form**

- Fill in the form as in task 1 and submit. The results will appear in an independent window. This is because the **target="\_blank"** attribute has been used within the **<form>** tag.

**2.4** Compare the results with your form code. Try to identify the functions of name, type and value attributes.

**Note** An attribute is a descriptive element of an HTML tag.

## Task 3 Including selection boxes and buttons

**Objectives** To add selection boxes and buttons to a form.

**Comments** When you amend forms you may find you need to force a complete page reload in order to view changes. To do this, hold down the **<Shift>** key whilst clicking on the **Reload** icon.

---

**3.1** Returning to Notepad, add the following code, shown in bold italic, to your form:

```
Enter name: <input type="text" name="MyNames" />
</p>
<p>
Radio button 1 <input type="radio" name="IHaveChosen"
value="the first button" /><br />
Radio button 2 <input type="radio" name="IHaveChosen"
value="the second button" />
</p>
<input type="submit" value="Submit form" />.
```

Unlike the **input type**, the **name** and **value** are defined by the user, in order to make them relevant to the form. Although spaces should not be left within attributes, the **value** is an exception as its function is that of a text string.

➤ Save **form.html** and reload (see Comments above).

**3.2** Fill in your amended form. You will notice that only one of the radio buttons can be selected at any time, just like on an old wireless. This is because the name **IhaveChosen** is attributed to both, thus creating a group of buttons (as opposed to differently named, independent buttons).

➤ Submit the form and examine the results.

**3.3** Unlike radio buttons, checkboxes allow multiple selection, even when grouped together. Add the following, shown in bold italic, to your form code:

```
Radio button 2 <input type="radio" name="IHaveChosen"
value="the second button" />
</p>
<p>
Checkbox 1 <input type="checkbox" name="CheckboxesSelected"
value="box one" /><br />
Checkbox 2 <input type="checkbox" name="CheckboxesSelected"
value="box two" /><br />
Checkbox 3 <input type="checkbox" name="CheckboxesSelected"
value="box three" />
</p>
<input type="submit" value="Submit form" />.
```

**3.4** Save, reload and fill in your form, selecting two or more checkboxes.

➤ Submit the form and examine the results.

## Task 4 Text input

**Objectives** To modify and add text fields to a form.

**Comments** Text elements can be customised within forms, eg, to limit input to a certain number of characters, or to expand entry boxes to accept more text without scrolling.

---

**4.1** Returning to Notepad, add the attribute, shown in bold italic, to the following line:  
`<input type="text" name="MyNamels" size="40" />`.

- Reload your page, noting the larger text box at the top of your form. If you type beyond the length of the box, text will scroll.

**4.2** Add the `maxlength` attribute, shown in bold italic, to limit the number of characters the box will accept:

```
<input type="text" name="MyNamels" size="40" maxlength="5" />
```

- Reload your page and try to enter more than five characters into the name box. This attribute is useful if the data has a specific format, for example, a date or password.

**4.3** For larger amounts of text, a `textarea` element can be useful. Unlike most form tags, the `textarea` has a `</textarea>` end tag. The height and width is set using the `rows` and `cols` attributes.

Whatever appears between the `<textarea>` and `</textarea>` container tags will appear as default text in the box

- Add the following bold italic lines below your checkboxes:

```
Checkbox 3 <input type="checkbox"
name="CheckboxesSelected" value="box three" />.
</p>
```

```
<p>
<textarea rows="10" cols="60" name="comments">Any
comments?</textarea>
</p>
```

```
<p>
<input type="submit" value="Submit form" />.
```

- Save and reload your page.

**4.4** Type an address into your `textarea`.

Although pressing the `<Return>` key will start a new line, this does not guarantee the results will be formatted accordingly. Such a function is performed by the CGI script.

- Submit your form and examine the results.

## Task 5 Creating menus

**Objectives** To create selection menus within HTML forms.

**Comments** Menus provide an alternative selection method to checkboxes and radio buttons. Used in conjunction with JavaScript, they are frequently used to provide a jump menu. Jump menus perform a similar role to standard hyperlinks, but are not covered in this module.

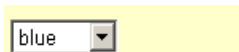
**5.1** A select menu is coded in a similar way to a standard HTML list. The **<select>** tag defines the menu and the **<option>**, and **</option>** tags list each item. Finally, the closing **</select>** tag closes the menu.

- Add the following bold italic lines to your form:

```
<textarea rows="10" cols="60" name="comments">Any comments?</textarea>
</p>
<p>
<select name="FavouriteColour">
<option selected="selected" value="I like blue">blue</option>
<option value="I like red">red</option>
<option value="I like green">green</option>
<option value="I like yellow">yellow</option>
</select>
</p>
<p>
```

- Save your file and reload your page. Notice that the menu only displays one visible item. This is the default **selected** option, designated by the **selected="selected"** attribute.

Clicking on the side scroll arrow will expand the menu and allow one selection to be made. A menu which expands in this way is often known as a dropdown menu.



**Figure 3 - a dropdown menu**

As with other form elements, the **value** attribute of the **option** tag is only visible within the form results.

**5.2** In order to allow a multiple selection, an attribute must be added to the form.

- Add the following bold italic text to your **<select>** tag:

```
<select name="FavouriteColour" multiple="multiple">
```

- Save and reload your form. You will notice that the menu is now permanently expanded in order to allow multiple selection (hint: when selecting, hold down the **<Ctrl>** key to selected individual items and the **<Shift>** key to include all items in between).

**Note** Adding the **size="n"** attribute (where n equals the number of list items) to the select tag will also expand the list, but not allow multiple selection by default.

## Task 6 Making forms more accessible

**Objectives** To add HTML elements and attributes to increase accessibility and usability of forms

**Comments** This module looks at labelling boxes and buttons, logical grouping of sections of a form and controlling tab order, for those using a keyboard to navigate the form.

### 6.1 Create sections to group related elements of your form.

- Add the following code, shown in bold italic, to your form:

```
<fieldset>
```

```
<p>
```

```
Enter name: <input type="text" name="MyName1" size="40" maxlength="5" />
```

```
</p>
```

```
</fieldset>
```

```
<fieldset>
```

```
<p>
```

```
Radio button 1 <input type="radio" name="IHaveChosen" value="the first  
button" /><br />
```

```
...
```

```
<option value="I like yellow">yellow</option>
```

```
</select>
```

```
</p>
```

```
</fieldset>
```

```
<p>
```

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

### 6.2 Save your file and refresh in your browser. Your form should be split into two sections, as in figure 4.

Enter name:

Radio button 1

Radio button 2

Checkbox 1

Checkbox 2

Checkbox 3

Any comments?

blue  
red  
green  
yellow

Submit form Start again

**Figure 4 – form divided into fieldsets**

### 6.3 Add a descriptive **legend** to each fieldset:

- Add the following code, shown in bold italic, just below the first **<fieldset>** tag:

```
<fieldset>  
<legend>Personal Details</legend>
```

- Repeat for the second fieldset, using appropriate text (for example, *My choices*)

#### 6.4 Add a label to your name box to associate it with its accompanying text:

- Add the **<label>** tag shown in bold italic, closing it at the end of the associated text:

```
<label>Enter name:</label> <input type="text" name="MyName1" />
```

- Add **for** and **id** attributes to associate the text with the input box:

```
<label for="yourname">Enter name:</label> <input id="yourname" type="text"  
name="MyName1" />
```

- Save your file.

**Note** Labels should be added to each element with accompanying text. In the case of menus, the **id** attribute should be added to the **<select>** tag. **Submit** and **reset** have labels automatically associated with them.

#### 6.5 Refresh your page in your browser. You should notice no difference. However, users of assistive technology (a screen reader for example) will be informed of relationships between text and input elements.

**Note** The **tabindex** attribute can also be used to restructure the order in which input elements are selected by those using a keyboard or other assistive technology to tab through a form, for example:

```
<input type="checkbox" name="shoesize" tabindex="4" ...
```

If input elements already run in the correct order, the **tabindex** should **not** be used

## Appendix A How do I put a form on the web?

The Common Gateway Interface, or CGI, is a standard for external programs to interface with information servers such as HTTP servers.

A general form-to-email gateway is provided for local use. The form HTML tag must be:

```
<form action="http://www.bristol.ac.uk/cgi-bin/mailform.pl" method="post">
```

All such forms must be registered with the gateway before any completed forms can be forwarded. To register your form, simply fill it in with sample input and click the submit button. You will be prompted for the email address to which completed forms should be sent. This information is then forwarded to the webmaster Team who will register the form and send a notification to the supplied email address. The process will take a few days. Note that if you change the URL of the page containing the form then it will need to be reregistered.

The following special field names are recognised by the form-to-email gateway:

- **mailform\_receipt**  
Allows you to specify the URL of an alternative 'thank you' page, other than the rather plain default one. Typically this page will be presented in the same 'house style' as the rest of your site, and will include a message of acknowledgement and a link back to an appropriate page of your choosing. The input tag should look similar to this:  

```
<input type="hidden" name="mailform_receipt"  
value="http://www.bristol.ac.uk/fictitious/thankyou.html">
```
- **mailform\_http\_referer**  
Allows you to specify the URL of the form's web page. This information is normally automatically provided by the browser and is used by the form-to-email gateway to identify the form and its corresponding email address. However we are increasingly seeing browsers withholding this information for personal privacy reasons. In these circumstances, if the 'http\_referer' field is present, its value will be used instead. So it is good practice to include within your form a hidden input tag similar to the following example:  

```
<input type="hidden" name="mailform_http_referer"  
value="http://www.bristol.ac.uk/fictitious/feedback.html">
```
- **mailform\_subject**  
Allows you to specify additional text for the Subject: field of the email message. For example:  

```
<input type="hidden" name="mailform_subject" value="Room booking">
```

This would result in an email with the subject "Form email return Room booking".
- **mailform\_format**  
Allows you to specify alternative formats for the email message. Accepted values are "raw" (the default), "text" and "html". The "text" option inserts additional line breaks in the body of the message to improve readability. The "html" option also inserts some basic HTML tags to improve further the readability of the message when viewed using an HTML aware mail client. For example:  

```
<input type="hidden" name="mailform_format" value="text">
```

### To summarise:

- Create a form web page with a form tag as above.
- Fill in the form with sample fields and click the submit button.
- When prompted, enter the email address to which future completed forms should be sent.

- Wait a few days for notification that the form has been registered.
- Publicise the form page.

Source: University of Bristol Webmaster Team - Frequently Asked Questions:  
<http://www.bris.ac.uk/is/faqs/websupport.html>