



.net

WEB CENTRE: HANDS-ON HTML

WEB CENTRE

Essential advice for creating your own Web site.

THIS MONTH:

Forms, frames and how CGI scripts work.

You may have noticed that our series of HTML tutorials has concentrated on creating Web pages that go mostly in one direction – from you to the reader. This is all fine and dandy if all you want to do is get your point across, but what if you also want some feedback? What if you desire specific information from your readers? Or what if you are the sort of person who is just craving that response and attention that you know your beautifully designed Web pages rightly deserve?

Don't despair. It is possible to add two-way communication to your Web pages and encourage people to get in touch with you and – even better – it's easy. After all, communication is what the Internet is all about and by providing areas where people can interact, you are increasing the chances that your pages will be used. It also shows that you are genuinely interested in what somebody else has to say for a change.

Keeping this in mind, we have prepared simple and easy-to-follow details for creating pages which people can use to send feedback to you. By the end of this installment, you should be able to build forms which will put the Department of Social Security to shame. And on top of this, we also have a myriad of other coding gems which will help make your Web site shine out above all others.

HANDS-ON HTML *part four*

.net WEB CENTRE

Tip of the month

Are you having problems navigating frames?

There are some problems using frames with Netscape Navigator. If you use the back button, you will be taken back to the previously loaded URL, and not the previous page, which can be a bit annoying. This happens because the back button calls up the last complete URL from the history list, which is usually a different site.

You can get around this on a PC by clicking the right mouse button within the frame and selecting 'back'. If you are using a Macintosh, hold down the mouse button and select 'back'. Also, if you want to print or bookmark a particular frame, you must select the frame by clicking in the desired window first.

Feedback from your Web pages will be abundant once you've read the fourth part of our six-part tutorial. Our resident HTML guru David Sikk explains the procedure for two-way communication.

Forms

One of the best ways to encourage two-way communication in your Web pages is by using forms. A normal mailto box is one way to give your reader room to type whatever they want, but forms are infinitely more powerful. They enable a user to enter information into pages containing buttons, selection lists, text entry boxes, checkboxes and all other forms of snazzy interaction. This is great if you are really inquisitive about who is looking at your pages.

Constructing a form is far less intimidating than it first looks. If you've already set up Web pages using HTML, then you'll find setting up forms familiar, because they are also written using HTML tags

embedded right into the page. Your Web pages can include multiple forms, however you cannot create a form within a form. Instead, you must end the current form with `</FORM>` before you can start a new one.

Here's the source HTML for a form which asks you to enter your name:

```
< FORM METHOD=POST ACTION=
"mailto:netmag@futurenet.co.uk" >
Enter your name: < INPUT NAME=your_name >
< INPUT TYPE=submit VALUE="Test this form" >
< /FORM >
```

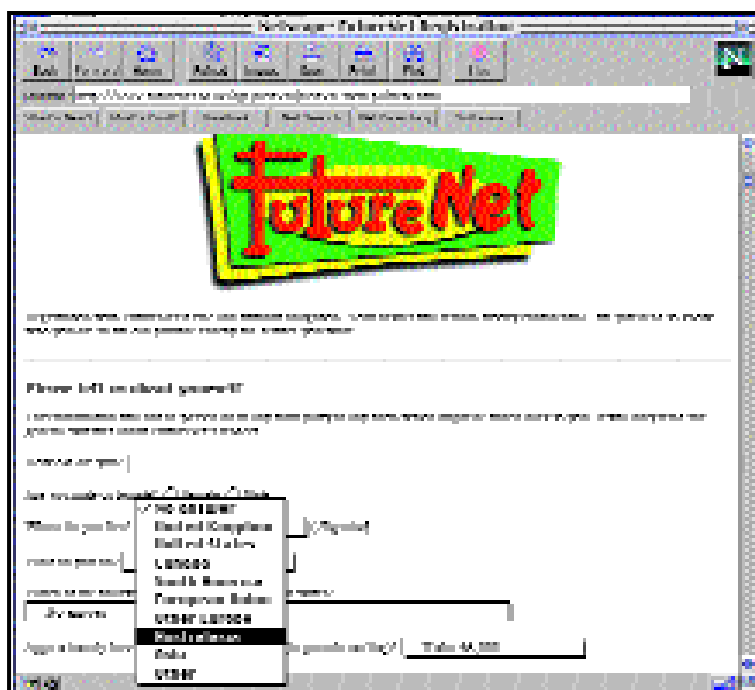
It looks a little daunting at first but it's a doddle really. The FORM tags start and end the form (all input fields of the form are placed between these two tags).

METHOD specifies how the Web server will pass the form data to the program which processes it (always set it to POST), and ACTION tells the server exactly which program that is. If POST is not capitalised, the method defaults to "GET".

Most, if not all of your forms have `/cgi-bin/form` as part of the address. This is a program which reads a configuration file you set up in your directory in which you define how you want the form data to be formatted, routed and what to display to the user after they submit the form.

Enter your name: `<INPUT NAME=your_name>`

This part of the code creates the text input box for the user's name. Most form fields begin with `<INPUT`



Set up a form to get feedback from your readers. Visitors to FutureNet can respond directly to netmag@futurenet.co.uk

Design tips

Form types

There are several form elements you can use, each of which is suited to a different job. Here are a few to get you going:

Drop-down list box

This presents choices to the user. The basic element is a box with a down arrow to the right of it. When the user selects the arrow, a list of choices unfurls. The code looks like this:

```
<SELECT NAME="Search feature">
<OPTION>Top ten search engines</OPTION>
<OPTION>Five funny things</OPTION>
<OPTION>Off-line search programs</OPTION>
```

Text fields

A text field gathers a single line of text from the user and is one of the most common fields used. The code is very simple:

```
<INPUT NAME="text in here">
```

Multiple selection list boxes

This is a bit like the drop-down list box but enables the user to select more than one of the items on a list. It can be created like this:

```
<SELECT NAME="type" SIZE="2"
MULTIPLE="MULTIPLE">
<OPTION VALUE="1">Top ten search engines</OPTION>
<OPTION VALUE="2">Five funny things</OPTION>
<OPTION VALUE="3">Off-line search programs</OPTION>
```

The SIZE indicates how many elements are to be expected in the list box. The MULTIPLE attribute says that more than one item in the list may be selected at a time.

Check boxes and radio buttons

These are an alternative way to collect one or more choices from a list of options. Check boxes enable users to select none, any or all of the choices. With radio buttons, only one of the choices can be selected. The code for checkboxes looks like this:

```
.net magazine: <INPUT
TYPE="CHECKBOX"
NAME="MAGAZINE">
```

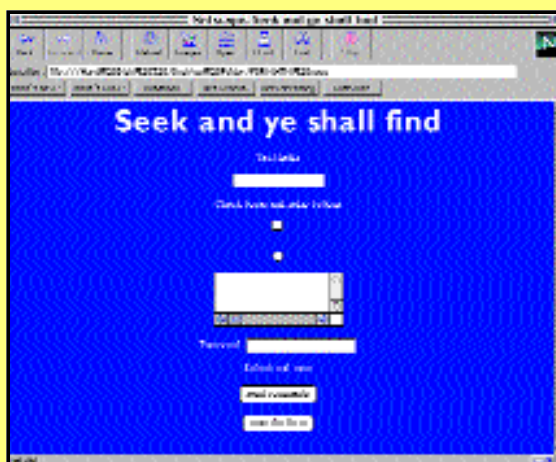
The code for radio buttons look like this: how do you rate .net magazine?

```
<BR>
Great<INPUT TYPE="RADIO"
NAME="rating">
Brilliant<INPUT TYPE="RADIO"
NAME="rating">
Outstanding<INPUT TYPE="RADIO"
NAME="rating">
where the NAME attribute is the name of this particular group of push-buttons.
```

Text areas

Text areas are large areas designed for free text entry that exceeds a single line. You can choose the size of the area by specifying the columns (COLS) and rows (ROWS) attributes. The HTML code looks like this:

```
<TEXTAREA NAME="text" ROWS="3"
COLS="40"></TEXTAREA>
```



Forms come in all shapes and sizes, choose one to suit your specific needs.

Password fields

Use a password field for text input areas that should be protected from "shoulder snoopers". On the Web browsers screen, asterisks appear instead of the text. Here is the code:

```
Password: <INPUT TYPE="PASSWORD"
SIZE="20" MAXLENGTH="16">
```

Submit and reset

Last but not least, the submit button is the crucial element that, when pressed, transmits the contents of the form. The reset button has the effect of clearing the fields on the form. The code for these looks like so:

```
<INPUT TYPE="SUBMIT" VALUE="send comments">
<INPUT TYPE="RESET" VALUE="reset the form">
```

A good place to put these on the page is at the bottom of the form. This ensures that the user scrolls all the way to the end of the form before sending it sending it off.



NAME="fieldname" and end with >
NAME defines the name of the data for the field and is how the program which processes the form references the data from this field.

You can pick any name for your fields as long as it is not one of the pre-defined forms parameters, because these perform certain specialised tasks. For instance, one of the most common errors in creating forms is to have an input field ask for a user's e-mail address and using the variable e-mail to store the address in. You also have to be careful to use only letters, numbers, and underscores in your field names. The form processor will not recognise such field names as e-mail addresses (it's also good practice to stick with letters, numbers and underscores for readability). TYPE defines the various types of input fields such as text, radio, checkbox, password, or submit. You can find out more about these in the Design Tips box on the left.

The text input type creates what is called a single line edit – an input field which enables the user to enter and/or edit a single line of text. Other things you can do with a text input field are to define an initial value for the field and change its size:

Enter your name: <INPUT TYPE=text NAME=your_name
SIZE=50 VALUE="Bruce">

Each form must have exactly one field of type submit. This creates the button the user pushes when they're done filling out a form, and should come after all the other input fields.

```
<INPUT TYPE=submit VALUE="Test this form">
```

If a form doesn't have a submit button, it would be a bit silly because once the user finishes typing in the data they would not have any way to send it off to the server.

When the user clicks the submit button, the browser collects the values of each of the input fields (text typed by the user and so on). They are then sent to the Web server identified in the ACTION keyword of the FORM open tag which in turn passes that data to the program identified in the ACTION, using the METHOD specified. In the picture on page 117, the reader's message would be sent winging its way to netmag@futurenet.co.uk [net](mailto:netmag@futurenet.co.uk)



Handy hint

Appealing forms

A form can contain normal HTML mark-up like headings, rulers, text and so on. Anything, in fact, except another form. This means they can be treated like normal pages and made to look just as good. Worth remembering if you want people to take the time to fill them in.



You can catch up with the first three parts of the Hands-on HTML tutorial by installing the PDF files from the Web Centre section of the CD-ROM. You can then read them on screen using the Adobe Acrobat reader, which is also on the CD. There are three HTML editors on the CD for you to try – Aardvaark Pro for Windows 95 and Windows 3.1, Arachnophilia for Windows 95 and HTML Editor 1.1.4 for Mac.



You can use forms to find out about your readers.

Top advice sites

There are loads of pages out there which are bursting with innovation and ideas. The only trouble is that searching for them is sometimes like finding a needle in a haystack. So here is this month's short-list to make things easier.

Forms
<http://fptoday.com/tables.htm>

A handy site which has details for making tables and forms. Check out the source code in the step-by-step tutorial to get all the details.

Frames
<http://itc.cit.cornell.edu/workshops/courseMaterials/frames/frames.html>
 This site has a course which will teach you what frames are, show you both good and bad examples of using frames on the Web, brainstorm about effective uses of frames and let you build a basic page using frames while learning the syntax.

Tesco
<http://shop.tesco.co.uk/prd/i/pgen/tesco/OL/tesco.html>
 The Tesco Internet Superstore enables you to perform your



Visit the Tesco Web site for some inspiration on how to use frames effectively.

weekly shopping on-line from your local Tesco branch provided it is equipped for home shopping delivery. Have a look around this virtual supermarket to pick up some hints from their innovative use of forms and frames.

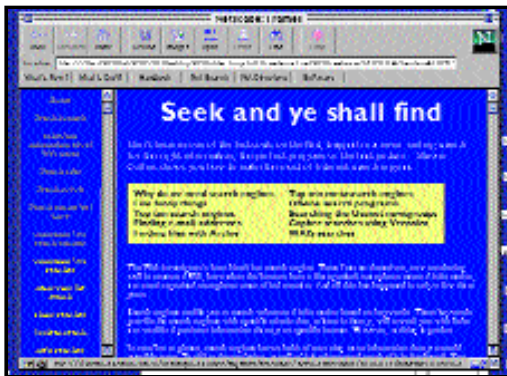
NEXT MONTH:

Other technologies and what they do: Java, Shockwave, Real Audio, Futuresplash, VRML.



Frames

Frames divide Web pages into multiple, scrollable areas enabling you to show information in a more flexible and useful way. You can create separate regions within the same document which can act completely independently of each other and can even have separate URLs. They can be set up so that clicking a link in one frame loads a page into another section and vice-versa.



Log on to FutureNet to see what David has been making with this tutorial.

Frames come into their own when they are used to show elements that the user should always see, such as control bars, copyright notices and title graphics which can be placed in a static, individual frame. As the user navigates the site in "live" frames, the static frame's contents remain fixed, even though the adjoining frames redraw.

Frame-based tables of contents are also more functional. One frame can contain links that, when clicked, display results in an adjoining frame. Other uses are with frames that sit side-by-side enabling queries to be posed and answered on the same page, with one frame holding the query form, and the other presenting the results. Frames provide endless opportunities for fun and here's how they work:

```
<HEAD>
<TITLE>A page with
frames</TITLE>
```

```
</HEAD>
<FRAMESET COLS="20%, 80%">
<FRAME SRC="CONTENTS.HTM">
<FRAME SRC="PAGE1.HTM" NAME="MAIN">
</FRAMESET>
</HTML>
```

This sets up two frames – the first a column over 20 per cent of the Web browser window width. The second is to the right of it and covers 80 per cent of the window width. The two FRAME tags tell the browser what to load into these frames. The page called CONTENTS.HTM goes into the left one and the right one gets PAGE1.HTM which you would have prepared earlier and placed into the same folder. This is a rather snazzy way to create a contents list using the left-hand frame which could list and load a number of different pages into the right-hand frame. **next**

Readers' Web sites

You get free Web space when you subscribe to **.net** and sign up with most service providers. Here's what you've been doing with it.



Heartlink
<http://www.ibmpcug.co.uk:80/~rwall/>



Packed with information about hearts and heart-related illnesses, this home page for the Heartlink charity combines easy navigation with quirky graphics. It doesn't use any fancy HTML coding or JavaScript, but it gets the information across swiftly and effectively.

Honey's Canine Friends
<http://subnet.virtual-pc.com/~an545337/>



Honey is a Cavalier King Charles spaniel bitch (not to be confused with one of King Charles' cavalier bitches) and she's here to tell you all about her owners and provide a few links to doggy Web sites. Decent use of tables is evident, but there's not a whole lot of tail-shakin' going on.



Gospel Music Archive
<http://subnet.virtual-pc.com/da567664/>



A sizeable collection of Gospel songs arranged for guitar. Each song includes the guitar chords, words and sometimes the tablature (if they could feel be bothered) so you can strum along with the best of 'em. A promising graphical start belies the text-based meat of the gospel matter, but it's functional.

Malcolm Hulme's Home Page
<http://www.btinternet.com/~eirridia/>



A good-looking well-designed site let down by some dodgy content. Malcolm provides some interesting information on his job, pastimes and life in the Shetland Islands, though living that far north seems a bit too fast-paced for us. "There's an otter at the bottom of my garden," says our Malc, "...and rabbits and sheep. And if anything interesting turns up I'll make a note of it here."



Paul's Dodgy Home Page
<http://www.netcomuk.co.uk/~jarvis/index.html>



Not, as you might expect, an on-line shrine dedicated to the chubby popsters of 'Good enough' fame, but a self-indulgent shoe-gazing site packed with pointless icons, broken links, garish backgrounds and (YES!) a photo album (under construction, obviously). Send him some abusive e-mail and tell him to get his lazy-student arse into gear.

If you'd like us to take a look at your Web site, e-mail its address and a one-line description to swright@futurenet.co.uk