

CREATING WEB FORMS AND FRAMES Using HTML



Creating Web Forms and Frames

1. What is a Web Form
2. What is a CGI Script File
3. Initiating the HTML File
4. Composing the Form Tags
 - Get and Post
 - Input Text
 - Input Password
 - Input Checkbox
 - Input Hidden
 - Input File
 - Input Radio
 - Input Option Select
 - Input Submit
 - Input Reset
5. Saving Your HTML Form
6. Testing Your Web File
7. Uploading the HTML File
8. What is a Frame
9. Frame tags
 - Frameset
 - Frame
 - NoFrames

Revised 9/01 PAR

What is a Form

Forms are a simple way that a person viewing your Web page can send you data. This is done by the user filling in various parts or input lines within the form designated by you.

Forms are supported by almost every browser and make it possible to create documents that collect and process user input and formulate personalized replies. Once a user fills out a form, it is submitted to a server or e-mailed to a specific address. If sent to a server, that server passes that information to a supporting program or application that processes the data.

What Software is Needed

The software you will need for generating the web form includes:

- ◆ Web browser to view a web page, such as Netscape or Internet Explorer.
- ◆ Text editor to create the HTML file; such as WordPad, or Notepad.
- ◆ FTP (File Transfer Protocol) program to upload a page. There are several available for a Mac or a PC.

CGI

CGI is the supporting program to process the data entered in the form. Server-side data-processing aspects of forms are not part of the HTML standard. They are defined by the server's software. The CGI behind this form creates an E-mail message by copying each form field's contents to a separate line. It mails the information to the specified E-mail address.

There is a generic CGI script available for use at Ohio University. Information on this script is available at:

<http://www.ohiou.edu/pagemasters/memo85/append5.html>

Mailto:

Mailto forms allow the programmer to set up a form such that all data collected from the form will be e-mailed to you or another e-mail address in simple text format. The data is not processed at all. This is a simpler method than the CGI format but can be more problematic because the user filling out the form must have their browser correctly set up to send e-mail. Therefore it is recommended that if your site is housed on an OU server you use the CGI method, but if it is a site at another server it is easier to use the Mailto method.

Initiating the HTML File

Click the **Start** button, trace to **Programs** option, then trace to **Accessories**, and click on **Notepad** to open it. Once in the Notepad program, enter the following commands to start an HTML document :

Create your HTML file so that it looks something like this. The HTML tags do not have to be in uppercase. The spacing between commands is up to you. However, the spacing within the brackets is important.

```
<HTML>
<HEAD>
<TITLE> Web Form </TITLE>
</HEAD>
<BODY>
There will be a form here soon.
</BODY>
</HTML>
```

Now we need to save the file as an HTML file. To do this click on the **File** menu and then choose **Save As**. Choose where you want to save the file. In this class we will be saving to the Public folder for convenience. Then choose a File Name to save it as. The name must end in **.htm or .html**. Next select the down arrow next to the Save as Type. Choose **All Files**. Finally, click on the **Save** button.

Now we will see what our HTML file will look like in a web browser. First we'll open Internet Explorer by clicking on the blue "E" on the Desktop, or under **Programs** in the **Start** menu choose **Internet Explorer**.

Once in **Internet Explorer** open your file. Click on **File** in the top left corner, then on **Open**, and then on **Browse**. Navigate to the Public file folder by double clicking it or highlighting it and choose your file. Click **Open**. You will see this:

"There will be a form here soon."

Throughout this class we will be making a series of changes to our HTML document, saving that document, then looking at it in the browser to see if those changes yielded the desired results. This is an important part of the web desing: learning how to make changes and saving those changes, then checking to see what the new changes look like, then making more changes.

Composing the Form

We'll compose our form assuming it is for an O.U. web page, using the generic CGI. The command `<FORM>` initiates a form section of a web pages and `</FORM>` ends that section. To begin a form we must tell the browser to send the information that a user enters into to a CGI script file and tell where that file is. We do that with the `ACTION` attribute:

```
<FORM action="http://ouvaxa.cats.ohiou.edu/htbin/genform.com">
```

Another attribute of the `FORM` tag is the `METHOD`, which is how the form input will be sent to the gateway. The method can be either "get" or "post". Post means to send the form entry results as a e-mail document. This is the most common method. Get is usually used with search engines. However, this is the method used by the OU CGI script. So the `FORM` command would look like this:

```
<FORM action="http://ouvaxa.cats.ohiou.edu/htbin/genform.com"
  method="get">
```

Next we must input the code to send the information from the form to the appropriate e-mail address. This is done with:

```
<INPUT type="hidden" name="mailto" value="youremail@ohiou.edu">
```

The Type="hidden" attribute hides this from the user. The name="mailto" attribute names this Input tag. The value="youremail@ohiou.edu" attribute tell the CGI program where to send the information from the form. So now our HTML Form commands should look like:

```
<BODY>
```

```
<FORM action="http://ouvaxa.cats.ohiou.edu/htbin/genform.com"
  method="get">
<INPUT type="hidden" name="mailto" value="youremail@ohiou.edu">
</FORM>
```

```
</BODY>
```

Input Commands

The CGI script is a generic script program provided by Ohio University. This means that it is set up to e-mail you the data from the form. Since it is a generic script, there are some limitations. See:

www.cats.ohiou.edu/pagemasters/memo85/append5.html

Input Limitations

- Text fields are limited to single lines. If you attempt to use multi-line text fields, any time a user enters **more than 255 characters, none of them** will be transcribed into the E-mail you receive. You can work around this limitation by also including a standard mailto link.
- Every field is limited to a maximum length of 80 characters!
- The Web page containing the form must be on an Ohio University server (contains ".ohiou.edu"). To test, upload the HTML file to your OU server, but use a different name or folder.

- The E-mail destination of the information must be one user of an OU system (contains ".ohiou.edu").
- The total number of fields defaults to 20. If you want to change this, you must include a hidden field specifying the number of fields your form uses.
- The names of the visible fields for data entry must be a lower case letter "f" followed by two or three digits from 01 to 999, inclusive.
- If you include fields numbered beyond 20, you must include a hidden form-field named "maxlines" with value equal to the highest field number used.
- You must include a valid Ohio University E-mail address in the hidden form-field named "mailto"!
- You may choose to provide an absolute URL in the hidden form-field named "nexturl" and appropriate link text in the hidden form-field named "nextname". If you do, they will be used to construct the return link.
- The visible fields can be text, radio-button, check-box, or pop-up-selection. Value for these fields must be at most 80 characters.
- The hidden form-field named "subject" is optional but strongly recommended; it identifies the Web page where the E-mail originates.
- If you do not use a particular field, or if the user leaves a field blank, a blank line will be included at that place in the message.
- You do not have to number the fields sequentially. You should organize the form in a way that will be logical and convenient for the user and number the fields so that the resulting e-mail will be easiest for the recipient to use.

Input Text

To create a single line of text for the user to enter we use the Input command, with the Type attribute equal to Text. For example:

```
<INPUT type="text">
```

That will create a box for text input. Using the OU CGI we have to name the input command appropriately:

```
<INPUT type="text" name="f01">
```

Then when the form is mailed to us we will get a line that say f01= whatever they entered in that text box. If we wanted something in the text box for the user to overwrite we would use the Value attribute. For example:

```
<INPUT type="text" name="f01" value="(###) ###-####">
```

(###) ###-####

The size of the text box can also be changed. To do this we use the Size attribute. The default size is 20. Here are some options, but remember that using OU's CGI script we can not go above size=80.

```
<INPUT type="text" name="f01" value="(###) ###-####" size="14">
```

```
<INPUT type="text" name="f01" size="30">
```

Now that we've seen a few text line options, let's make some changes to our HTML document and see what it looks like.

```
<FORM action="http://ouvaxa.cats.ohiou.edu/htbin/genform.com"
  method="get">
```

```
<INPUT type="hidden" name="mailto" value="youremail@ohiou.edu">
```

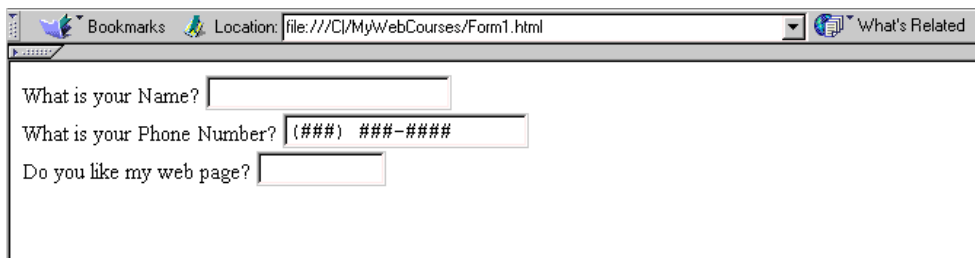
```
What is your Name? <INPUT type="text" name="f01" size="20"> <BR>
```

```
What is your Phone Number? <INPUT type="text" name="f02"
value="(###) ###-####"> <BR>
```

```
Do you like my web page? <INPUT type="text" name="f03" size="10">
```

```
</FORM>
```

The new command above is
 for Break. It is like hitting the enter key. The different text lines in our form will now be on separate lines.



The screenshot shows a web browser window with the address bar displaying "file:///C:/MyWebCourses/Form1.html". The browser has "Bookmarks" and "What's Related" buttons. The form content is as follows:

What is your Name?

What is your Phone Number?

Do you like my web page?

Checkboxes

Checkboxes are an option on a form that allows users to select a line of text in an on/off yes/no method. On screen they appear as a small box that either has a check in it or does not. The command for a checkbox is as follows.

```
<INPUT type="checkbox" name="f11" value="checkbox a">
```

Checkboxes can be checked by default with the addition of the keyword CHECKED in the INPUT tag. Here is an example.

```
<INPUT type="checkbox" CHECKED name="f12" value="checkbox b">
```

Now we put the two tags together and add some text afterward like this:

```
<INPUT type="checkbox" name="f11" value="checkbox a">This is a  
checkbox.
```

```
<P>
```

```
<INPUT type="checkbox" CHECKED name="f12" value="checkbox b">  
This is a checkbox that is automatically checked.
```

Option Select Lists

These lists are drop-down windows in which a user selects a choice from a list of options selected by the programmer. The code for an option select list with three choices follows.

```
<SELECT type="text" name="f05" size=1>  
<OPTION value="first">Your first choice  
<OPTION value="second">Your second choice  
<OPTION value="third">Your final answer  
</SELECT>
```

| |
|--|
| Your final answer |
| Your first choice Your second choice Your final answer |

If you want one of the choices to be automatically selected, add the code SELECTED after the OPTION command like this.

```
<OPTION value="first" SELECTED>
```

Radio Buttons

Radio buttons are similar to checkboxes. However, they have some limitations that checkboxes do not. This limitation is that it forces the user to choose one and only one of the choices. One may be selected by default.

Below is the code for two radio buttons with text following them. The second of these buttons has been selected by default the same way a checkbox was selected with the SELECTED feature inserted after the INPUT command.

```
<INPUT type="radio" name="f07" value="radio1"> This is a radio button.
```

```
<P>
```

```
<INPUT type="radio" CHECKED name="f07" value="button2"> This one  
has been selected by default.
```

Hidden

A Hidden input is a name/value pair that is returned to you but does not show up anywhere on the web page. The text for a Hidden input is simple.

```
<INPUT type="hidden" name="Location" value="USA Form">
```

When the form is returned to us we would get:

```
Location=USA Form
```

When using the OU CGI form, the Hidden type is needed by the CGI program stating where to send the data from the form. For example,

```
<INPUT type="hidden" name="mailto" value="youremail@ohiou.edu">
```


Password

This feature allows the user to enter a password that does not appear on screen but will be sent to you. The command for the password is similar to a text line and can be edited the same way.

Please enter your password.

```
<INPUT type="password" name="UserPass">
```

Textarea

This command allows you to generate a text box on your form for user input, not just a line of wrapping text. The basic command for this is:

```
<TEXTAREA name="anyname">  
</TEXTAREA>
```

We can edit this by adding more attributes within the tag. Columns and rows can be described. Also text added between the starting and ending Textarea tags appear within the text box. This text is formatted exactly as typed including tabs, spacing and returns. Below is an example of this.

Please add any comments you may have about this form class here

```
<P>
```

```
<TEXTAREA name="Comments" rows="9" cols="44">
```

Constructive criticism carries more clout
than negative does.

 Tabs and
returns work within TEXTAREAS.

```
</TEXTAREA>
```

Constructive criticism carries more clout
than negative does.

 Tabs and

returns work within TEXTAREAS.

Submit Image

Earlier we learned how to create a submit button on the form. We also saw how to edit the text within that submit button. Now we will learn how to use an image instead of a button to send the form. Note, you can only make a Submit button, not a Reset image button.

```
<INPUT type="image" src="submit.gif">
```



Saving

Every time changes have been made to your simple text documents they should have been saved. The method to save as an HTML file has been discussed earlier. To save you would go to the **File** menu and choose **Save** or **Save As**. Make sure that the file extension (ending) is **.htm** or **.html**.

Testing and Uploading

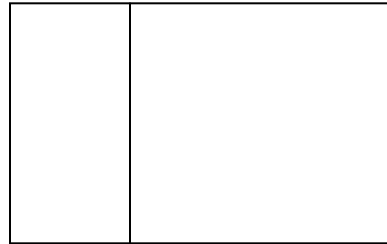
Initial stages of testing are done after changes are made to see the effects of text and HTML changes on the view of the page. To fully test the form it needs to be uploaded to a server. We will be using our OAK accounts provided by the university. File uploading is done using an FTP program and is covered in more detail in Part One "Creating Web Pages with HTML"

Once the HTML file containing the form has been uploaded it can be accessed on the Internet and further tested by you, the programmer, filling it out and then sending the results to yourself.

Frames

Frames are an easy way to make any page look more professional. The frame commands tells the browser to split the screen into two or more parts, each with a separate web page. To do this we need to create a web page with the commands to tell the browser how to split the screen. You begin with the FRAMESET section which is in place of the BODY section.

```
<HTML>
<HEAD>
<TITLE>Sample </TITLE>
</HEAD>
  <FRAMESET COLS="25%,*">
    <FRAME SRC="pageone.html">
    <FRAME SRC="pagetwo.html">
  </FRAMESET>
</HTML>
```



In this FRAMESET tag we tell the browser to split the window vertically with the attribute COLS. There will be two pages shown. The first one taking up 25% of the screen, and the second one taking up the remainder of the screen. To split the window horizontally, use the ROWS attribute.

```
<FRAMESET ROWS="75%,*">
<FRAME SRC="pageone.html">
<FRAME SCR="pagetwo.html">
</FRAMESET>
```



We also need to tell the browser the names (URLs) of the web pages to go into the separate windows. The FRAME command tells the browser the location of the page to be viewed in a particular frame. The first page source will be displayed on the left column or the top row and the remaining pages will follow.

Frames can be used as a navigational tool for the users to browse through our pages. To do this we need to create several web pages. One page to hold the navigational links on the left, one as the default page and a couple to practice navigating. First, to generate the page that tells how to set up the frames, enter the following. Save this file as "**Frame.html**".

```
<HTML>
<HEAD>
<TITLE>Start Frames </TITLE>
</HEAD>
  <FRAMESET COLS="20%,*">
  <FRAME SRC="Contents.html">
  <FRAME SRC="Default.html" name="main">
  </FRAMESET>
</HTML>
```

This will create the main page that will hold our other pages in its frames. Naming the second, or right hand window frame "main" will allow us to switch this frame for others depending on which link our users click on.

Now we need to create a page that will be held in the larger right window as the default before the user selects which page to view. Enter the following text. Save this file as "**Default.html**".

```
<HTML>
<HEAD>
<TITLE> My Default Page </TITLE>
</HEAD>
  <BODY bgcolor="white">
  <FONT size=6 color="blue">
  <CENTER>
  This is my main page.
  <P> From here you will be able to navigate to
  my different sites by choosing my links on the left.
  </CENTER>
  </BODY>
</HTML>
```

Now we need to create the Contents page. This page will be seen in the first or left hand window. Enter the following into Notepad and save as "**Contents.html**".

```
<HTML>
<HEAD>
<TITLE> My Contents Page </TITLE>
</HEAD>
  <BODY BGCOLOR="yellow">
    <B> Pick An Animal </B>
    <P><A HREF="dog.html" TARGET=main>DOG</A>
    <P><A HREF="cat.html" TARGET=main>CAT</A>
    <P><A HREF="bird.html" TARGET=main>BIRD</A>
    <P>
  </BODY>
</HTML>
```

Within each anchor tag notice the "TARGET=main" addition. This tells the browser to put the page signified by the link address into the Frame named "Main". You may choose any name you wish, but it must match. For time saving purposes, the pages dog.html, cat.html, and bird.html are already on your lab computer. Please copy them and the images with the same names, to wherever your html file is located. Once all this is done, you should get the following:

