

FORMS

Introduction

Forms are a way to gather information from people who visit your web site. Forms allow you to ask visitors for specific information or give them an opportunity to send feedback, questions, or requests to you. Common uses of forms include polls, e-commerce, feedbacks, and registration. A form contains fields in which users enter information. These fields can be text fields, radio button, checkboxes, menus, or lists.

There are two aspects to creating Forms: creating the Form Objects (Text Fields, Checkboxes, Submit buttons, etc.) and making the Forms function properly. This lesson will focus more on the creation of Forms Objects, not on the programming required to make Forms transmit data to and from your server. We will be using a mail cgi (common gateway interface) that has been written for use on polylog1. Any more than that is beyond the scope of Dreamweaver and this class. In my opinion, Forms (and by association, the programming that makes them work) is at the heart of what makes the Web different from paper and publishing mediums of the past.

In this lesson you will:

- Create a form on a web page
- Add single-line text fields
- Add multi-line text fields
- Add checkboxes
- Add radio buttons
- Add list/menu items
- Add buttons
- Validate a form
- Create a jump menu
- Link the form to a Cal Poly mail program
- Test a form

Form Basics

A form begins and ends with the HTML `<form>` tag. The opening `<form>` tag indicates the beginning of the form and sets its properties; the closing `</form>` tag, of course, marks the form's end.

In between these tags, different objects provide the basic user interface elements – the design – of the form. Radio button, text fields, and pull-down menus (also called pop-up menus) are just a few of the ways you can gather input. It's perfectly OK to include other HTML elements inside a form, too, in fact you site's visitors would be lost if you couldn't also add (and format) text that explains the purpose of each form object. And without using a table to control the layout of a form, it can quickly become an unreadable mess.

Every form element, whether it's a text field or a checkbox, has a **name** and a **value**. The name indicates what information the element is intended to collect; for example, if you want your visitors to type their email addresses into a text field, the name of that field might be **email**. It is the user's input – what your visitors type into the text field, for example, or the selections they make from a pull-down menu.

After your visitors fill out the form and click the Submit button to transmit their responses, each form element is transmitted as a name/value pair like this: `email=bob@bobville.com`. Submitting both pieces of information lets the program that processes the form figure out what the input

means. After all, without a name, a value doesn't mean much. The name/value pair (age=39) provides context for a user's input.

The Code Backstage

Creating a form is just the first step in collecting information from you Web site's visitors. You also need to connect the form to a computer program that actually **does** something with the information. The program may simply take the data from the form and email it to you; however, it could do something as complex as contacting a bank, processing a credit card payment, creating an invoice, and notifying a shipping department to deliver the latest Stephen King novel to someone in Nova Scotia. A form is a petty useless thing without a program like this running on the web server.

These information-crunching programs can be written in a variety of languages – Perl, C, VBScript, JavaScript, Java – and may be part of a dedicated application server such as Cold Fusion Server or Microsoft's Active Server Page technology. The point is the Dreamweaver, by itself, can only create the front end of your web site forms; it can't actually collect or process the data that your web visitors type into it.

Writing the necessary behind-the-scenes processing software can be complex, but the concepts behind forms are straightforward:

1. Someone fills out a form on your web page and clicks the submit button.
2. The form data is transmitted over the Internet to a program on your web server.
3. The program collects the data and does something with it – whatever you and the programmer decide it should do.
4. Finally, the web server returns a page to the user. It may be a standard web page, with a message like – “Thanks for the Info!” or a page dynamically generated by the program itself – a detailed invoice page, for example.

So, how do you create the processing half of the forms equation if you not a programmer? If you're part of a Web development team in a company, you may already have programmers in-house who can help you create the processing program.

And, many web hosting companies offer free forms-processing programs as part of their services. Here at Cal Poly, we have in place one such program – it is call **sendmail.cgi** and we will use it to email the results of our forms to ourselves.

Form Objects in Dreamweaver

FORMS OBJECTS IN DREAMWEAVER	
Name	Function
Form	This is the very first step in creating a Form. This inserts the <Form> tag into your document. If you do not place all your Objects inside the <Form> tag, your form will not work.
Text Field	Inserts a Text Field Object on your Form. They can be set to contain single or multiple lines of data.
Button	Inserts a Submit button Object on your form.
Checkbox	Inserts a Checkbox Object on your form. These checkboxes are used to toggle between selecting a single option on a form.
Radio Button	Radio buttons are used to select one item out of a list of available options.
File List/Menu	These two objects (list or menu) allow you to make single or multiple selections in a small area of space
File Field	Inserts a text box and button that lets the end user browse to a file on their hard drive, for uploading.
Image Field	Inserts an image into a Form, that can be used to make graphic-based buttons.
Hidden Field	Stores information that does not need to be displayed but is necessary for processing the Form on the server.

Creating a Form

Before adding individual fields, you must place a form on the page. The form will contain fields into which users enter information, and it will specify what should be done with the data.

1. Open the **myform.html** document from the **forms** folder. Position the insertion point below the text **“Enter our drawing for a free adventure trip”** and choose **Insert > Form**.

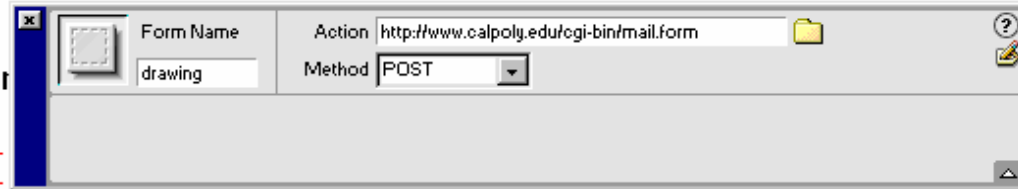
Red dotted lines visually define the form area in the document window; that area is located between the **<form>** and **</form>** tags in the code. These red lines are invisible elements that will only be visible in Dreamweaver. The red lines are not draggable. The size of the form area depends on what you place inside the form, and it will expand as much as necessary to accommodate the contents.

2. Select the form by clicking the red dotted line. The property inspector changes to display form properties. There are several options on the property inspector.
 - a. **Form Name** lets you give the form a name if you want to control it with a script application like JavaScript. Name it **drawing**.
 - b. In the **Action script** type:

```
http://www.calpoly.edu/cgi-bin/mail.form
```

it tells the browser what to do with the form data, generally a URL where the CGI program resides.

- c. Leave the **Method at Post**. Method defines how the form data is handled: GET, POST, or Default. Data sent by a form is a continuous string of text from the information typed in by the user. GET appends form contents to the URL specified in the Action text box. GET is not a secure method of transferring data, so it should not be used for sensitive information. Usually you'll use the POST option. POST send the form value in the body of the message.

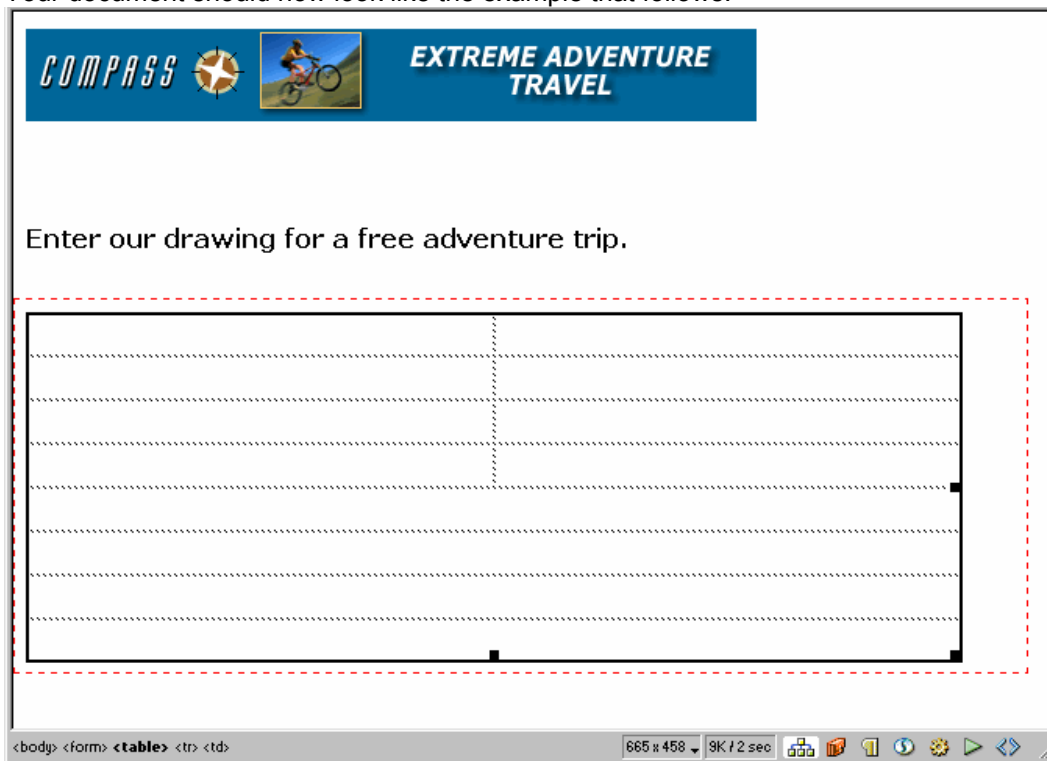


- Place the insertion point in the area between the red dotted lines. Click the **Insert Table** from the Common category of the Objects panel. In the Insert Table dialog box, make the table **8 rows, 2 columns**, and **600 pixels wide**. Set the **border to 0**, the **cell padding to 5**, and the **cell spacing to 0**. Click OK.

The table will improve the layout of the form and make it easy to align text or images with the form fields to label them. You can place a table inside a form or you can place a form in a table, but the table must be completely contained or be contained by the form.

- Select the top four cells in the left column. From the **Hroz pop-up menu** of the property inspector, choose **Right** to Right justify all four cells. These four cells will contain the text that label the form fields. You will add the labels to the table as you add the form fields.
- Merge both cells in row 5. Repeat this step for rows 6, 7, and 8. The first four rows each have two cells. The last four rows now have a single cell each.
- Save the document.

Your document should now look like the example that follows.



Adding Single-Line Text Fields

Text fields are for gathering information the user types at a keyboard. Typical single-line text fields collect name, address, and e-mail information from users.

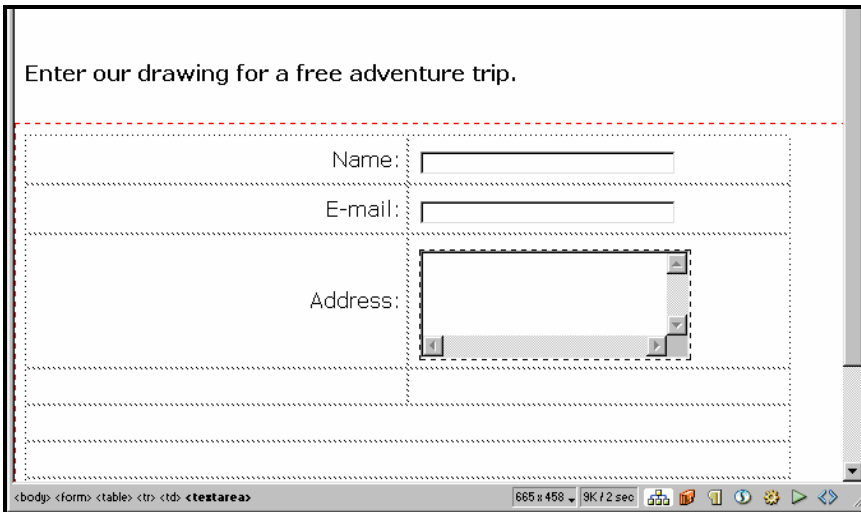
You must place all form fields and buttons within the red dotted lines; otherwise, they will not be part of the form. If you try to insert form fields outside of the red lines, Dreamweaver will display an alert box with Yes or No options asking if you want to add a form tag. If you choose No, the field or buttons will not function as a part of any form.

1. Type **Name**: in the left cell of row 1 and press **Tab**. The insertion point moves to the next table cell.
2. Choose **Insert > Form Objects >Text Field**. (or choose the **Text Field icon** in the Form Objects panel). A single line text field is placed in the form. The property inspector displays Text Field properties whenever a text field is inserted or when you click on a text field in the document window. The default Type option is a Single line.
3. In the **Name text box** on the property inspector, replace **textfield** with **name**. When the form is submitted, the name of the text field identifies the information that was entered into the field. In this case, "name" signifies that the information entered into this field is the visitor's name. Names are required for all fields. Do not use any spaces or special characters in the name, and remember that names are case-sensitive.
4. In the **Char wide text box** on the property inspector, type **40**. The initial width of the text field is about 24, even though the text box is blank. The actual size of the text field in the browser will vary because it is dependent upon the size to which the user has set their browser's default text. The height of the text box is also determined by the browser's default text size.
5. In the **Max Chars** text box on the property inspector, type **50**. Max Chars limits the total number of characters a user can enter. Initially this text box is blank, and the number of characters a user can enter is unlimited.
6. Type **E-mail** in the left cell of row 2 and press **tab**. Add a single-line text field in the right cell and name the field **email**. Set the **Char Width** to **40** and the **Max Char** to **70**. This field will accept the user's email address.
7. Save and preview the file in a browser.

Adding Multiple-Line Text Fields

Multiple-line text fields are used to collect larger amounts of information from a user. Typical multiple-line text fields collect comments and feedback from users.

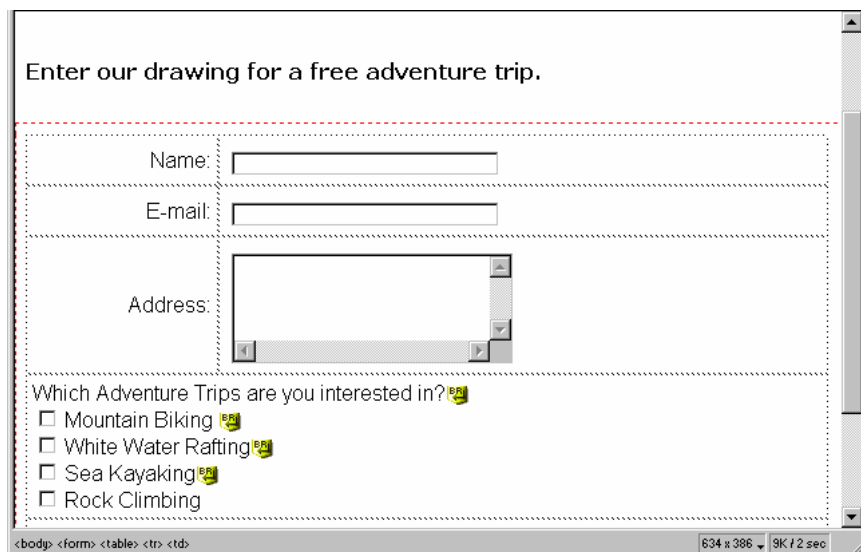
1. Type **Address** in the left cell or row 3 and press **Tab**. The multiple-line text field will be inserted in the same column as the single-line text boxes from above.
2. Choose **>Insert > Form Objects > Text field**. A single line text field is placed in the form and the property inspector shows Text Field properties.
3. On the property inspector, select **Multi line** for the **Type Action**.
4. In the **Name** text box on the property inspector, replace **textfield** with **address**.
5. In the **Char Width** text box, type **40**. You will see an additional option on the property inspector. The **wrap drop-down menu** is only available for multiple-line text fields. Wrap specifies how text that is typed into a multiple line field is displayed if there is more text than will fit in the visible area. Choose **default**.
6. In the **Num lines** text box on the property inspector, type **4**. This option dictates how many lines will appear in the scrollable area. It does not limit the number of lines a user can enter.
7. Save the file and preview it in a browser.
Your document should now look similar to this:



Adding Checkboxes

Checkboxes allow users to choose one or more options in a group of related items. Checkboxes are typically used when you want the user to choose as many of the listed options as desired. If you want your user to choose only one selection, then you should use a radio button.

1. In the **myform.html** document, type **"Which adventure trips would you be interested in?"** in row **5** of the table. Add a line break after the text by pressing Shift+Return.
2. Choose **Insert > Form Objects > Check Box**. A checkbox is inserted into the form, and the property inspector displays checkbox properties.
3. With the checkbox still selected, replace the word checkbox with the word **biking** into the **Name** text box on the property inspector.
4. Type the words **Mountain Biking** next to the check box.
5. In the **checked value** text box, type **Yes**. When the visitor to the page checks the biking checkbox, the name and value will be passed to the CGI to indicate that the checkbox has been selected.
6. Position the insertion point below **Mountain Biking** using a line break. Repeat steps 2 through 5, adding checkboxes for **White Rafting**, **Sea Kayaking**, and **Rock Climbing**, each on a separate line.
7. In the **name text box**, type **rafting**, **kayaking**, and **climbing**, respectively. Type **yes** in each **Checked Box Value text box**. Your document should now look similar to this.
8. Save the file and preview it in a browser.



Adding Radio Buttons

Radio Buttons are a group of options in which selecting one option automatically deselects all other options. Typical uses for radio buttons are credit card selections and yes/no answers.

1. Type **Have you ever taken an adventure trip before?** In row 6 of the table.
2. On the same line as the text, choose the **Insert Radio Button** from the Objects panel. A radio button is inserted into the form.
3. Type **Yes** to the right of the radio button you just inserted.
4. Click on the radio button to select it. The property inspector displays the radio button properties. In the **Name text box**, replace **radiobutton** with **trip**.
5. In the **Checked Value text box**, type **yes**. When the form is submitted, this value will be sent to the script that processes the form on the server.
6. Repeat steps 2 through 5 for the **"No"** answer. Place it to the right of the **Yes**.
7. This time, type **No** in for the Checked Value, but use the **same name (trip)** in the **Name text box**.

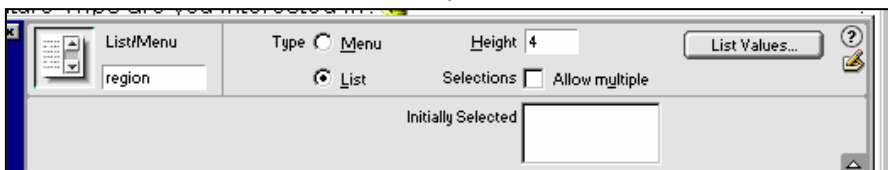
When using radio buttons, you must use the same name for each group of items. Radio buttons are meant to allow only one selection. Using the same name for several radio buttons indicates that those buttons are part of the same group. If the names are not the same, the radio buttons will be treated as different groups and negate the purpose of using radio buttons.

8. Save the file

Adding List/Menu Items

A list/menu enables users to pick options from a scrolling list or menu. A scrolling list gives you the option to allow users to make multiple contiguous or noncontiguous selections. A drop-down menu restricts users to one selection. In both types, items chosen by the user will be highlighted.

1. In the document, type **Which regions are you interested in exploring?** in row 7 of the table and insert a line break.
2. Choose **Insert > Form Objects > List/Menu**. On the property inspector, select **List** for the **Type** option and change the **Height** to **4**. A small menu is inserted into the form, and the property inspector displays Menu properties. Dreamweaver inserts a drop-down menu by default. You've changed the format to a scrolling list. Check the **Allow Multiple** checkbox for the **Selection** option.



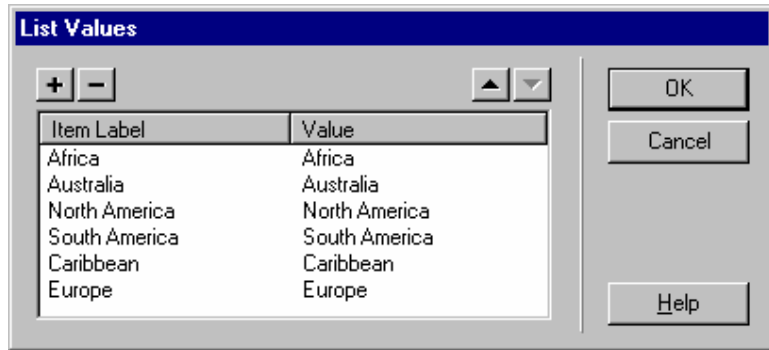
3. In the **Name text box**, replace **select** with **region**, and click **List Values...** The

4. In the **Item Label** field, type **Africa**, then press Tab. The longest item on the list determines the width of the list/menu.
5. In the **Value** field, again type **Africa**. This text is sent to the CGI or server to indicate that the option has been selected.
6. Press Tab or click the **plus (+)** sign in the dialog box to add another option to the menu. Use the minus (-) sign to delete items from the list values box.
7. Repeat steps 4 through 6, adding **Australia, North America, South America, Caribbean, and Europe** to the list.
8. Change the **Value** field to match the name of each region. It should look like the following:

- Click OK to close the dialog box. The list shows the regions you just added.
- Save the file and preview it in a browser.

An addition option of the property inspect for list/menu items is the Initially Selected

Box. You can choose to have any one of the items in the list be selected when the page loads. This might not be desirable for scrolling lists, but for menu items it is helpful for the drop-down menu to have a sample choice or instruction appear on the first line.



Adding Buttons

Forms usually have two buttons, one to sent the form data (Submit) and one to clear the form (Reset). The Submit button tells the browser to send the data according to the Action and Method Specified. The Reset button clears all information from the fields on the page.

- Position the insertion point in the last row of the table, and click the **Insert Button** from the **Forms Object panel**. A submit button is placed in the form, and the property inspector displays Button properties. Since a submit button is the default, you do not need to change any of the options for this button.
- Position the insertion point on the same line just after the submit button and add another button to the page. The only difference is the name - this button is called **submit2** because no two button can have the same name. The only form objects that can have the same name are radio buttons, which can be grouped with other radio buttons.
- Choose **Reset form** as the **Action** on the property inspector. The text label will

automatically change to Reset. The button name, however, remains the same. This action will cause all text fields, checkboxes, and radio buttons to clear and revert to their original state when the page was first loaded in the browser.

- In the **Label box**, type **Clear Form**; in the Button Name text box, type **Reset**.
- Save the file and preview it in a browser.

Validating Forms

The Validate Form behavior checks the contents of text fields in a form to ensure the user has entered the proper information. Whether accidentally or intentionally, users sometimes enter the wrong information or skip a field entirely. You can check data as the user enters it or just before the form has been sent to the server.

In this exercise, you will add a behavior that will check the information typed into a form to be sure all required text fields have been filled out and that the information is the right type of data.

1. Click the **email field** that is in the second row of the table. Choose **Windows > Behaviors**. The Behaviors panel opens.
2. Choose **Validate Form** from the **Actions drop-down menu** in the Behaviors panel. The Validate Form dialog box opens.
3. Select the **email text field** from the **Named Fields** list on the dialog box. Check the **Required box** for the **Value Option**.

This specifies that it is necessary for the e-mail field to contain data. The Validate Form action is added to the Behaviors panel with the onBlur event. Now an error message will be displayed if the user tabs to the next field without filling in this one. The onBlur event is activated when the user leaves the text field. In this example, if the user tabs to the next field in the form, leaving the e-mail text field blank, an error message will be displayed. If the user clicks into specific fields (not tabbing through the form) this initial validation check is ignored.

4. Choose **Email address** from the **Accept** options. Click OK. This option will check whether or not the text field contains an @ symbol.
5. Select the **Submit button** on the document. Click the **plus (+)** button on the Behaviors panel to add an action and choose **Validate Form** from the drop-down menu. The Validate Form action is added to the Behaviors panel with the **onSubmit** event. Now when the user clicks the Submit button, the checks will be made.
6. Select the **Name text field** from the **Named Fields** list on the dialog box. Check the **Required box** for the **Value Option**.
7. Select **Anything** from the **Accept** options. Anything specifies that the field is required, but it does not need to have certain type of data.
8. Click OK.
9. Save the file and test it in a browser. Click in the e-mail field and then press Tab without entering any data. You will see an error message generated by the Validate Form behavior.

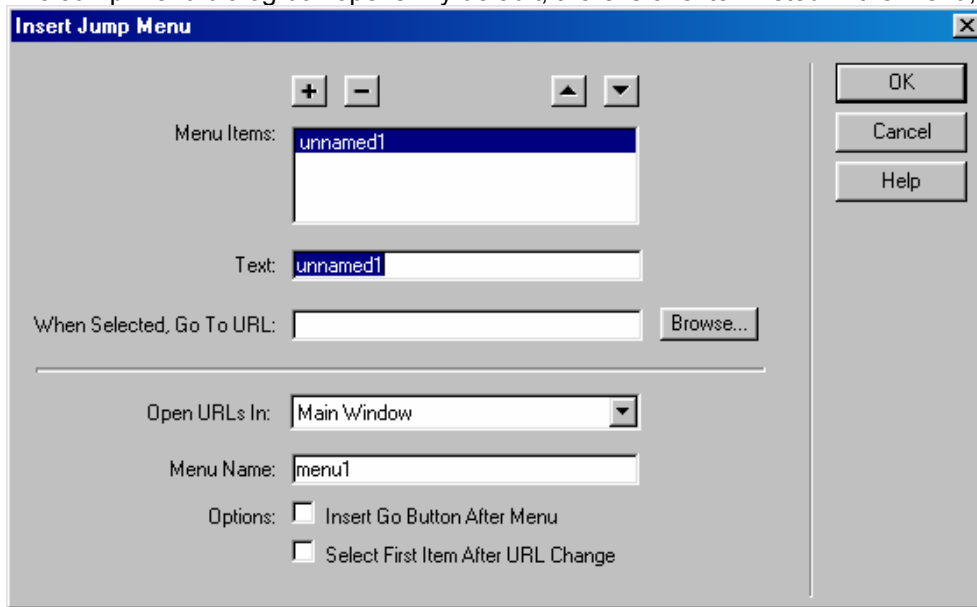
NOTE: To add the e-mail field to the checks that occur when the Submit button is clicked, open the Behaviors panel. Select the e-mail field and the desired option. Click OK when you are done.

Creating Jump Menus

A jump menu is a drop-down menu that contains links to other pages in your site or to other web sites. Like regular links, the jump menu can link to any type of file, including graphics or PDF files. The jump menu provides an easy-to-use interface for linking to pages in your site. A jump menu is embedded in a form and looks like a menu list in the browser.

1. Place the insertion point at the top of the page, on a blank line between the banner and the text “**Enter our drawing for a free adventure trip.**” Choose **Insert > Form Object > Jump Menu**.

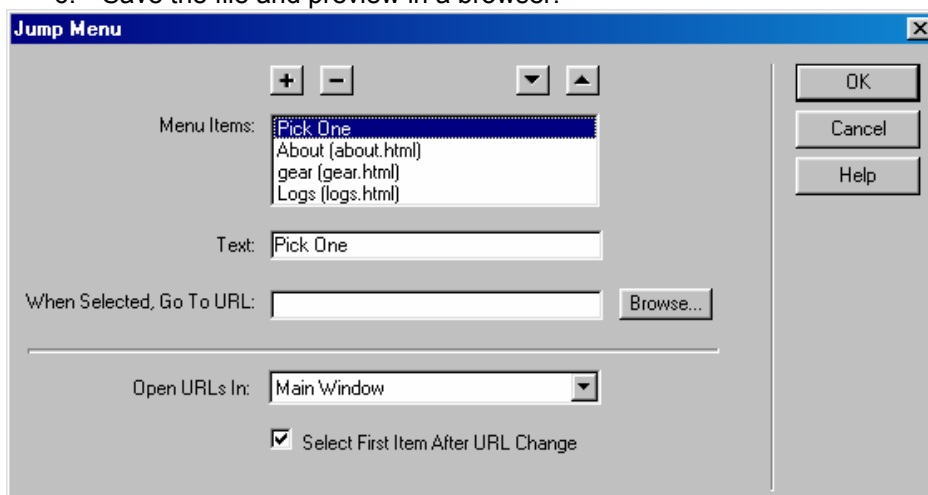
The Jump menu dialog box opens. By default, there is one item listed in the menu, **unnamed1**.



2. In the **Text** box of the **Jump Menu** dialog box, type **Pick One**. Leave the **When Selected, Go to URL** text box blank.

The first item in the menu list will appear in the first line of the menu. Since the user will see this initially in the menu list, the first line should be a short description of the list or a short instruction to let the user know that this is a jump menu.

3. In the Options area, choose **Select First Item after URL change**. This forces the menu list to display the first menu item in the list when the user returns to this page; otherwise, the list displays the most recent option chosen.
4. Click the **plus (+) button** to add a new menu item. Type **about** in the Text field, press Tab, and type **about.html** in the **When Selected, Go to URL** text box.
5. Repeat this step, entering **Gear, gear.html** and **Logs, logs.html**. Click OK when you are done. When these items are selected in the browser window, they will link to their appropriate pages. A link will be activated when the user selects the corresponding item.
6. Save the file and preview in a browser.



If you want to add a GO button to your list, select the **Insert Go Button After Menu** checkbox. In most cases, you won't need a Go button, because choosing an item from the jump menu normally takes you to the appropriate URL.

MAILING A FORM THROUGH THE CAL POLY WWW SERVER

In order for forms input to be returned, there has to be a program available to the server to handle the input. Such a program has been created here at Cal Poly. It is called a 'mail.form. It accepts the input from your form, formats it for mail delivery, and mails it to you. Following are the instructions for doing this. They are also available at <http://www.calpoly.edu/~www/mail.form.html>

1. Your form needs to include certain lines which reference the mail.form program. To begin, let's assume your Central Unix user id is "jswanson".
2. To do this with our form that we have created in Dreamweaver, we must go to the html. To do this, click on **F10**.
3. Find the line in the HTML that begins with <form method=POST action=""> We need to add the following information between the two quote marks:

```
http://www.calpoly.edu/cgi-bin/mail.form
```

This line tells the server that a form is starting, and to use the mail form program to process the form.

4. Hit the enter key to go down one line.
5. Next type the following line:

```
<input name="Location:" type=hidden  
value="http://www.calpoly.edu/~jswanson/forms/pol  
icy.html">
```

The value above, is the URL you want the user to see next, after the form has been submitted. If you leave this out, the user will be sent to a generic acknowledgement page indicating the form has been submitted.

6. Hit the enter key to go down one line.
7. Type in the following line:

```
<input name="To:" type=hidden value="jswanson">
```

The value above is the address that the form input will be mailed to, in this case, jswanson. This is a required field. **This form ONLY WORKS WHEN SENT TO CALPOLY ACCOUNTS, SO IF YOU WERE TO PUT IN SOMETHING LIKE: jswanson@thegrid.net. IT WOULD NOT WORK.**

8. Hit the enter key to go down one line.
9. Type in the following line:

```
<input name="From:" type=hidden value="The User  
of My Form">
```

This line is whatever you want the mail to look like it came from.

10. Hit the enter key to go down one line.
11. Type in the following line:

```
<input name="Subject:" type=hidden value="Online  
Registration">
```

This line is whatever you want the subject of the mail message you receive to be. If you leave it out, the line defaults to "WWW Form Values".

12. Hit the enter key to go down one line.
13. Type in the following line:

```
<input name="X-comment:" type=hidden value="Judy  
Swanson,  
http://www.calpoly.edu/~jswanson/forms/form.html"  
>
```

THIS LINE IS REQUIRED: Put your name and the web address of the form you are creating here for tracking or trouble-shooting (if you use more than one form, you will want to readily identify the location of the form).

14. A form also need a few other lines, like the </form> tag but Dreamweaver already added this for you.
15. Save the file.

What the whole deal looks like:

```
<form method="post" action="http://www.calpoly.edu/cgi-  
bin/mail.form">  
  <input name="Location:" type=hidden  
value="http://www.calpoly.edu/~jswanson/forms/policy.html">  
  <input name="To:" type=hidden value="jswanson">  
  <i  
Swanson">  
  <input name="Subject:" type=hidden value="Online  
Registration">  
  
Swanson, http://www.calpoly.edu/~jswanson/forms/form.html">
```