

ITEC447

Web Projects

CHAPTER 9 – FORMS

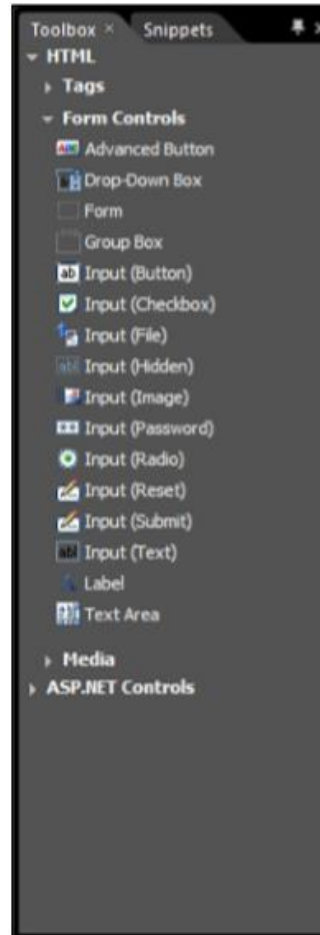
Getting Interactive with Forms

- The last few years have seen the emergence of the interactive web or “Web 2.0,” as people like to call it.
- The interactive web is an evolution from one-way communication to two-way communication, where the content becomes a conversation rather than an information session.
- At the core of this evolution lies a simple group of tools introduced shortly after the World Wide Web came into existence.
- These tools are Hypertext Markup Language (HTML) forms, and they give the visitor the ability to input information and communicate with the site rather than just ingest the information on it.

Creating Forms in Expression Web 4

- HTML defines all the different form elements and form controls.
- As a result, when you place these elements inside `<form>` tags in an HTML page, the browser automatically knows what they are and how they are supposed to behave; all you need to do is tell the browser what to do with the information gathered.
- To make the creation of forms as easy as possible, Expression Web 4 provides all the available form controls in one convenient location: the *Toolbox* panel.
- From here, you can drag and drop any of the form controls directly into either *Code* view or *Design* view without writing a single line of code.

Creating Forms in Expression Web 4 Cont.



When you hover the mouse over each form control, Expression Web 4 provides a ScreenTip with a short explanation of what each control does.

Creating Forms in Expression Web 4

Cont.



- **Advanced Button** creates a button whose actions are defined by the designer by embedding HTML code.



- **Drop-Down Box** creates a drop-down box where you can define the different options.



- **Form** is the outer tag that defines the form as a whole. The group of all the elements contained within the `<form>` tags makes up the form. A page can have multiple independent forms.









- **Group Box** creates a separate subgroup or box within the form. You can use this control to visually separate different sections of a form and still preserve the form's integrity by keeping it together. This is a purely visual tool without any actual function.



- **Input (Button)** creates a standard HTML button with an *onclick* event that can trigger JavaScript. This function is most often used to create *Submit*, *Cancel*, and *Reset* buttons but can be used for any other purpose as well.

Creating Forms in Expression Web 4

Cont.

-  **Input (Check Box)** creates a check box. The check box lets the visitor make yes-or-no choices, such as “Send Copy of the Information to Your Own Email address?”
-  **Input (File)** creates a text box with a Browse button attached that lets the user input a location or browse for a file on her computer to submit along with the rest of the information in the form.
-  **Input (Hidden)** creates a hidden text box that is invisible in the browser window but present in the code. This function is often used to insert extra content into a form without giving the visitor a chance to change it.
-  **Input (Image)** makes an image “click sensitive,” meaning you can use it as a button. The function also collects the x and y values of where on the image you clicked. This information can be used for added interactivity.
-  **Input (Password)** creates a text field where you can input a password. When the visitor enters text into the field, each character is replaced by an * symbol.
-  **Input (Radio)** creates a radio button. These buttons are similar to check boxes, but rather than being standalone yes/no units, they are connected and work as multiple choice.

Creating Forms in Expression Web 4

Cont.



- **Input (Reset)** creates a Reset button that, when clicked, sets the value of each element in the form to its default setting.



- **Input (Submit)** creates a Submit button that submits the form.



- **Input (Text)** creates a single-line text box that can collect text, such as a name or an address.



- **Label** associates a label with a form control, meaning that you can create a line of text, image, or other element connected to and working as a description of a specific form control. The Label can also set keyboard shortcuts for specific form controls.



- **Text Area** creates a multiline text box for longer segments of text. You can define how many lines of text the box allows.

- Because form controls are a bit cryptic in their description, it is easier to understand how they work by seeing them in action. The following exercise uses some of the form controls to make a simple email form.

Create an Email Form

- One of the most basic and most useful applications of an HTML form is to create an email form for your website.
- In addition to giving the user the ability to send emails to you directly from your website, an email form can also help reduce the amount of spam you receive every day.
- If you leave a *mailto:* hyperlink in your page, spam bots (computers that automatically sift through the Web looking for email addresses and then inundate them with spam) will find the address and use it.
- If you place an email form on the page instead, the spam bot will have a much harder time finding your address and you will most likely receive less spam.

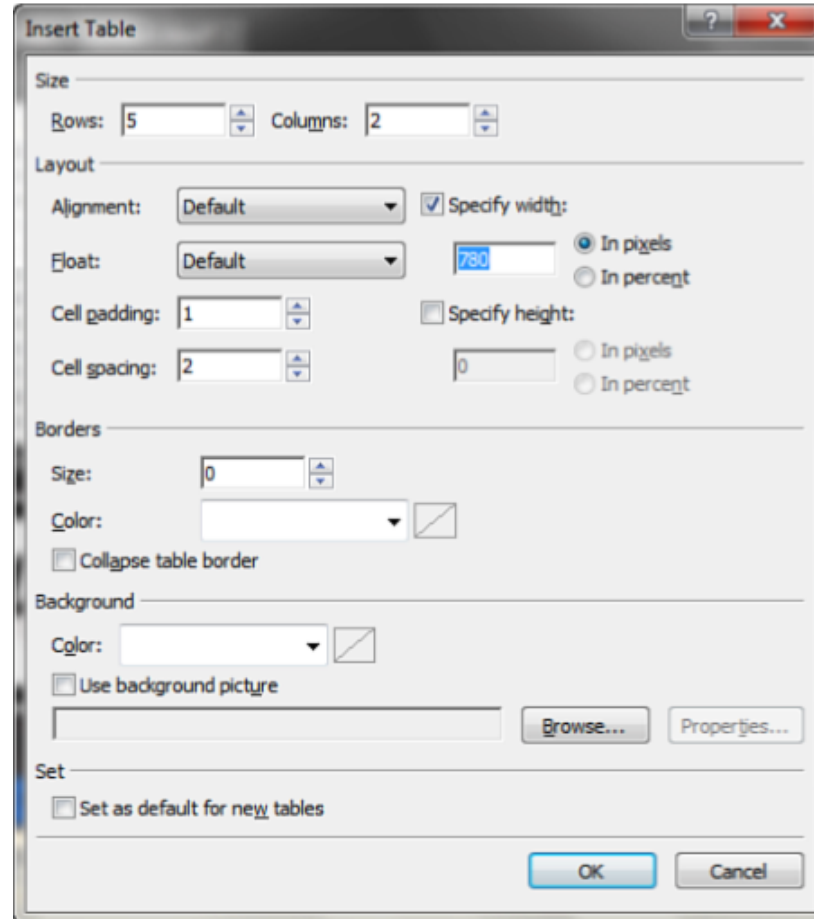
Create an Email Form Cont.

- Create a new page from the DWT you created in Hour 19, “Dynamic Web Templates,” by selecting *New, Create from Dynamic Web Template* on the menu bar and then save it as *contact.html*.
- Change the *Heading* to *Contact Me* and place the cursor in the contents editable region.
- In *Split* view, remove the `<p>` and `</p>` tags in *Code* view.
- Go to the *Form Controls* in the *Toolbox* panel and find the *Form* control.
- Anything contained within this control is considered part of the same form.
- Double-click the *Form* control to place it in the editable region of the page.

Create an Email Form Cont.

- Place the cursor inside the new form area in *Design* view and select *Table, Insert Table* on the menu bar to open the *Insert Table* dialog.
- Set *Rows* to 5 and *Columns* to 2.
- Under *Specify Width*, check the *In Pixels* radio button and set the *width* to 780.
- Click *OK* to insert the table.

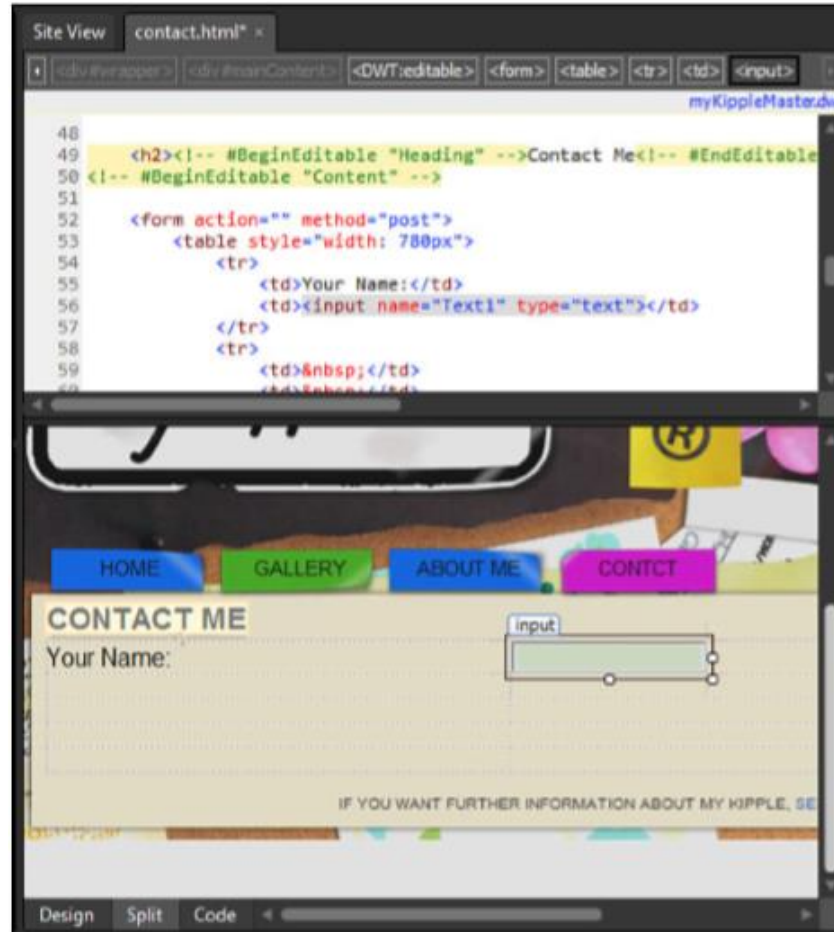
Create an Email Form Cont.



Create an Email Form Cont.

- With the cursor placed in the first cell of the table, type *Your Name:.*
- Press the *Tab* key to move to the next cell.
- Go to the *Form Controls* in the *Toolbox* panel and find the *Input (Text)* control.
- Place it in the cell either by dragging and dropping it or by double-clicking it in the *Toolbox*.

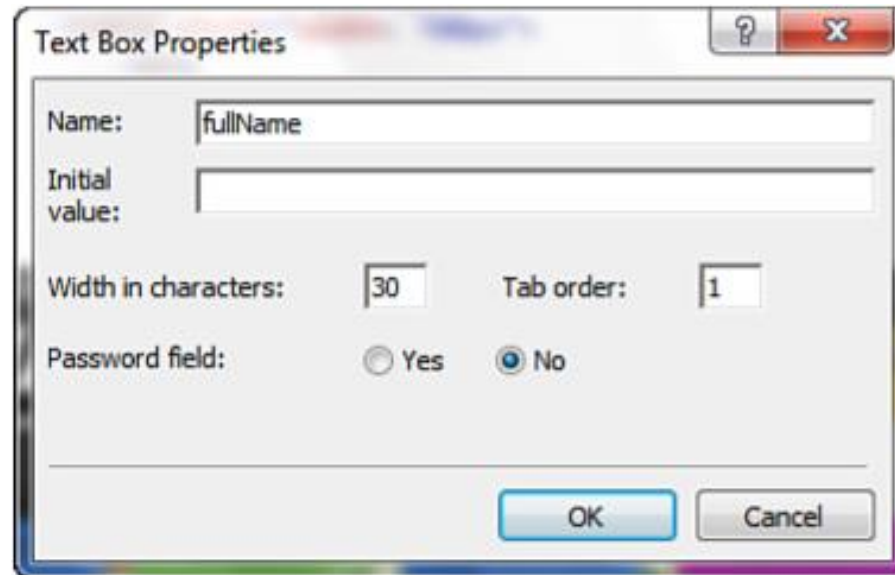
Create an Email Form Cont.



Create an Email Form Cont.

- In *Design* view, right-click the new form control and select *Form Field Properties* from the context menu.
- This opens the *Form Field Properties* dialog for this form control.
- Set the *Name* to *fullName*, the *Width in Characters* to *30*, and the *Tab Order* to *1*.
- Click *OK* to apply the changes.
- Back in *Design* view, place the cursor in the second left cell and type *Your E-mail Address:.*
- Press the *Tab* key to move to the next cell and insert another *Input (Text)* control there.
- Open the *Form Field Properties* dialog for the new form control and set the *Name* to *email*, the *Width in Characters* to *30*, and the *Tab Order* to *2* to match the first field.

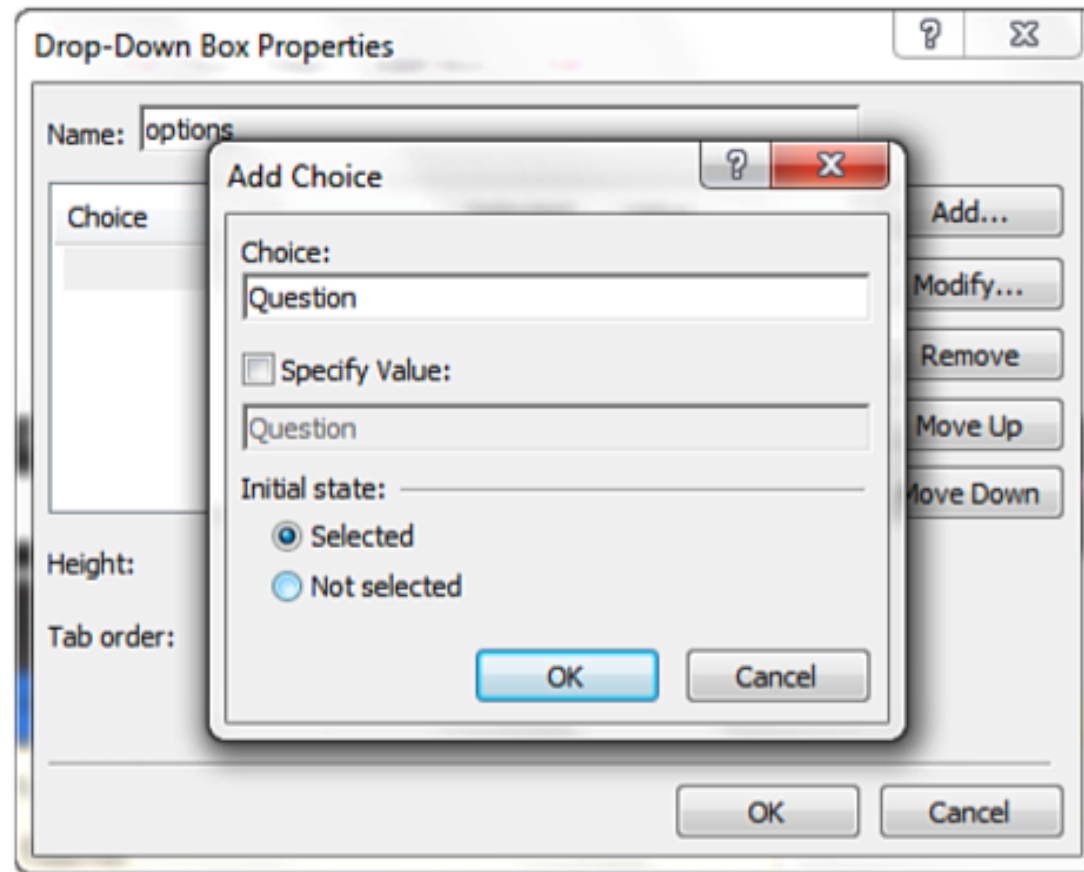
Create an Email Form Cont.



Create an Email Form Cont.

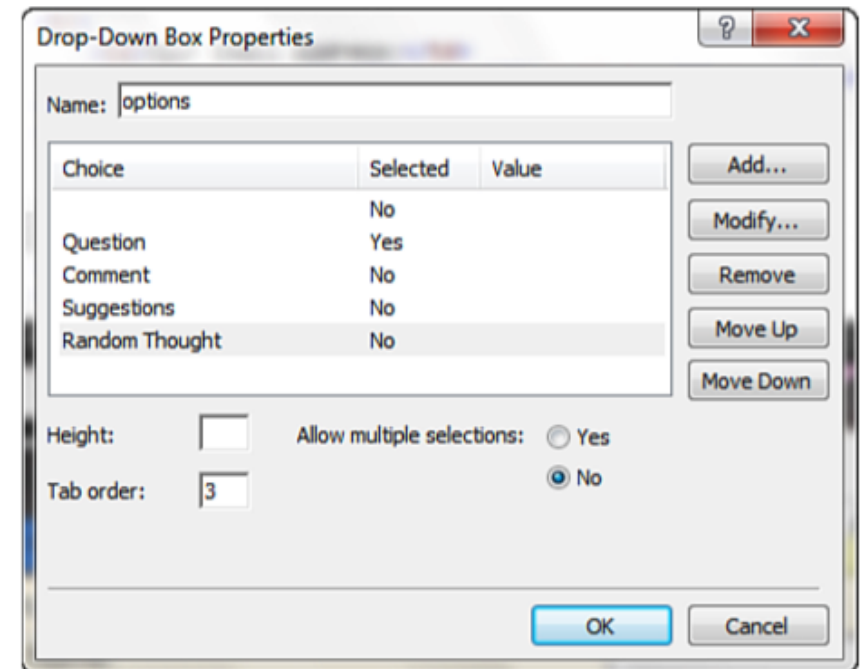
- Place the cursor in the third left cell and type *Type of Inquiry:*.
- Press the *Tab* key to go to the next cell and insert a *Drop-Down Box* control.
- Right-click the *Drop-Down Box* and select *Form Field Properties* from the context menu to open the *Form Field Properties* dialog.
- In the *Drop-Down Box Properties* dialog, set *Name* to *options*.
- Click the *Add* button to create a new choice.
- In the *Add Choice* dialog, type *Question* as the choice and click the *Selected* radio button under *Initial State*.
- Click *OK* to add the new choice.

Create an Email Form Cont.



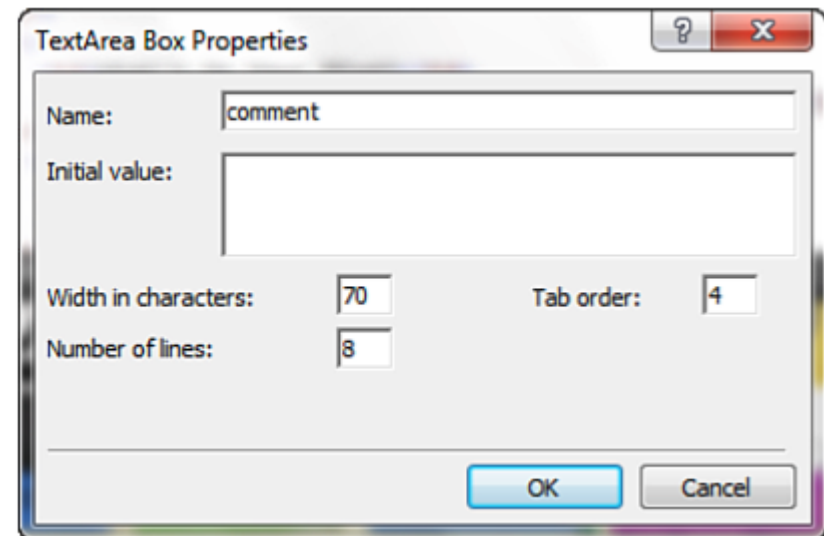
Create an Email Form Cont.

- Use the *Add* button to create three more choices: *Comment*, *Suggestion*, and *Random Thought*.
- Set the *initial state* for all of them to *Not Selected*.
- When all the choices are created, set the *Tab Order* to 3 and click *OK*.
- The *Drop-Down Box* automatically resizes itself to fit the longest choice.



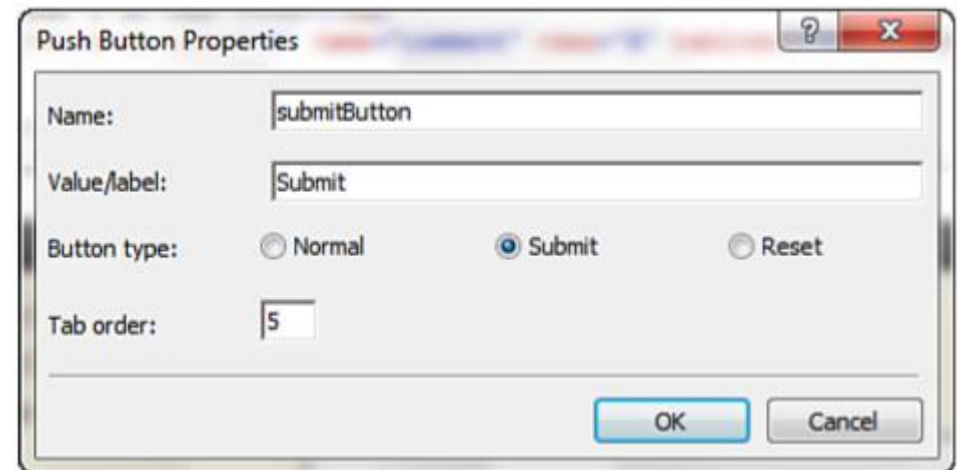
Create an Email Form Cont.

- In the fourth left cell, type *What's On Your Mind?*.
- Press the *Tab* key to move to the next cell and insert a *Text Area* form control.
- Open the *Form Field Properties* for the *Text Area*.
- Set the *Name* to *comment*, *Width in Characters* to *70*, *Number of Lines* to *8*, and *Tab Order* to *4*.
- Click *OK* to apply the changes.



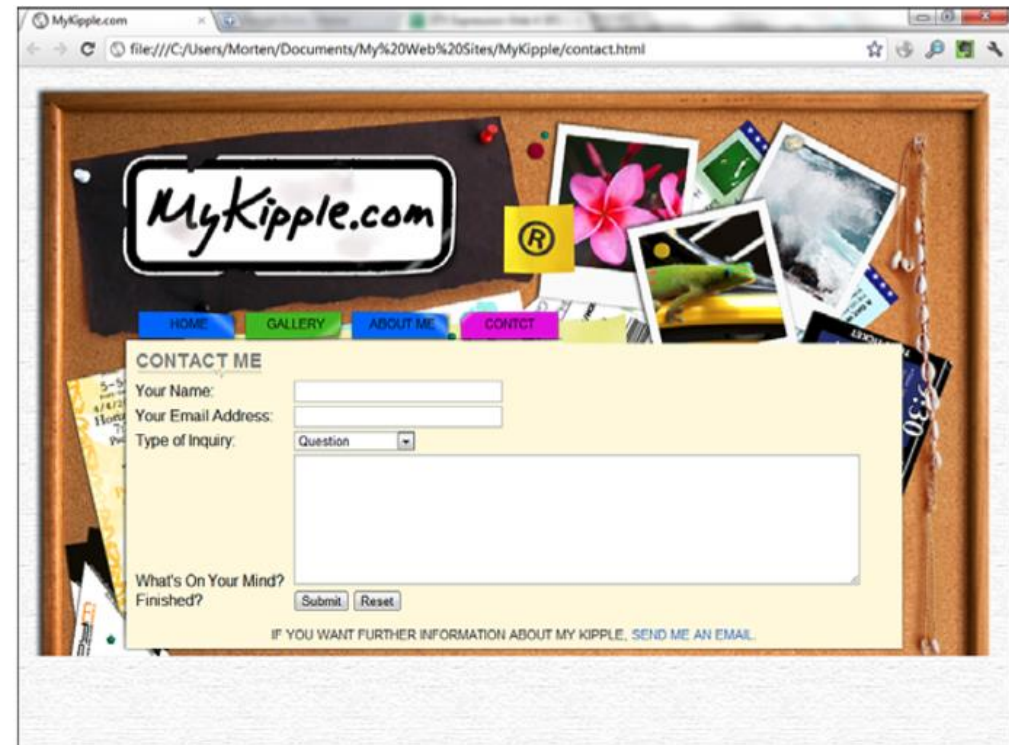
Create an Email Form Cont.

- Place the cursor in the bottom-left cell and type *Finished?*.
- Press the *Tab* key to move to the lower-right cell and insert an *Input (Submit)* and an *Input (Reset)* form control.
- This creates two buttons that read *Submit* and *Reset*, respectively.
- Open the *Form Field Properties* for the *Submit* button. (step 17)
- In the *Push Button Properties*, change the *Name* to *submitButton*, change the *Value/Label* to *Submit* (with an uppercase S), leave the *Button Type* set to *Submit*, and set the *Tab Order* to 5.
- Click *OK* to apply the changes.



Create an Email Form Cont.

- Repeat step 17 for the *Reset* button, but change the *Name* to *resetButton*, the *Value/Label* to *Reset*, and set the *Tab Order* to 6.
- Now all the basic components for the email form are in place.
- Save the page and test it in your browser.



Send Form Results

- Now that you have a basic form built in to your page, you need to connect it to the functions that make it work.
- Provided your server has FrontPage Server extensions installed, Expression Web 4 has built-in functions that make it easy to create forms to send results directly to a predefined email address, file, or database.

Send Form Results Cont.

- Right-click anywhere inside the email form and select *Form Properties* from the context menu.
- This opens the *Form Properties* dialog.
- Because you want the form to generate an email, select the *Send To* radio button at the top of the dialog.
- In the *Send To* area, you have two options.
- You can enter a filename and you can enter an email address.
- If you enter a filename, the results of the form will be sent to a file stored on the server.
- The two options can be used separately or together.
- Saving the results to a file and sending it to an email address is an easy way to make sure the message doesn't get lost.

Send Form Results Cont.

Form Properties

Where to store results

Send to (Requires FrontPage Server Extensions)

File name: Browse...


E-mail address:

Send to database (Requires FrontPage Server Extensions)

Send to other

Form properties

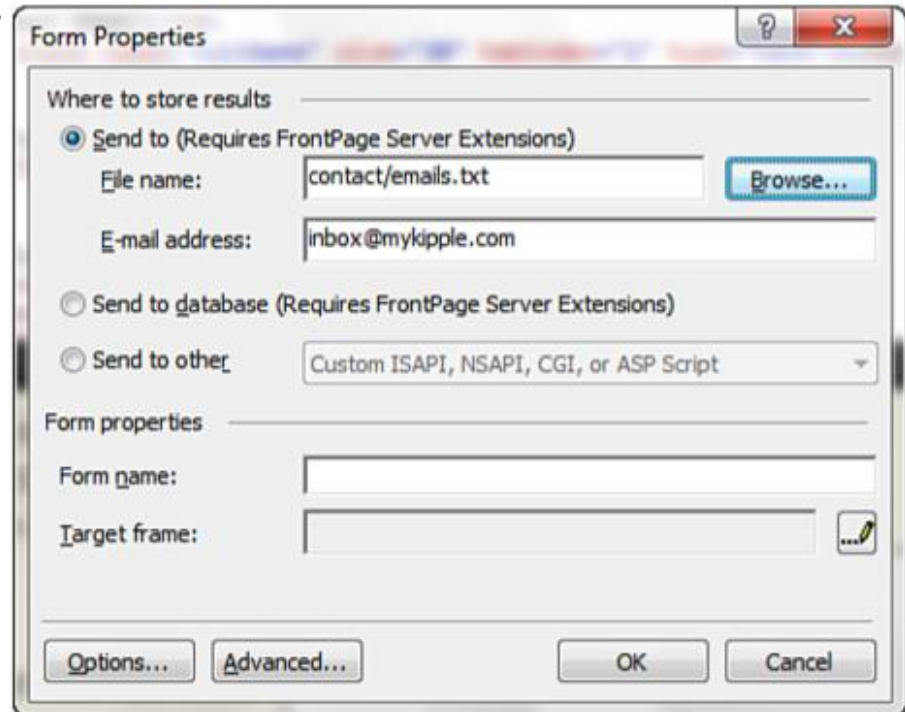
Form name:

Target frame: 

Options... Advanced... OK Cancel

Send Form Results Cont.

- Click the *Browse* button next to the *File Name* area to open the browser and create a new folder called *Contact*.
- Click *Cancel* and enter *Contact/emails.txt* in the *File Name* area.
- In the *E-mail Address* area, enter your own email address.



The screenshot shows the 'Form Properties' dialog box. The 'Where to store results' section is active, with the 'Send to (Requires FrontPage Server Extensions)' radio button selected. The 'File name' field contains 'contact/emails.txt' and has a 'Browse...' button next to it. The 'E-mail address' field contains 'inbox@mykipple.com'. Below this, there are two other radio button options: 'Send to database (Requires FrontPage Server Extensions)' and 'Send to other' with a dropdown menu showing 'Custom ISAPI, NSAPI, CGI, or ASP Script'. The 'Form properties' section below has empty fields for 'Form name' and 'Target frame'. At the bottom, there are buttons for 'Options...', 'Advanced...', 'OK', and 'Cancel'.

Send Form Results Cont.

- To further configure the output generated by the form, click the *Options* button.
- This opens the *Saving Results* dialog.
- The first tab of the *Saving Results* dialog deals with the file to which the content is saved.
- From here, you can change the location of the file and the file format (how the content is displayed within the file).
- For most purposes, *Formatted Text* is the best option, but there are many alternatives, including *HTML and XML* and *text database* variants.
- Below the *File Format* option, you can check whether you want to include the form field names and whether you want to generate an optional second file.
- Leave everything as it is in this tab.

Send Form Results Cont.

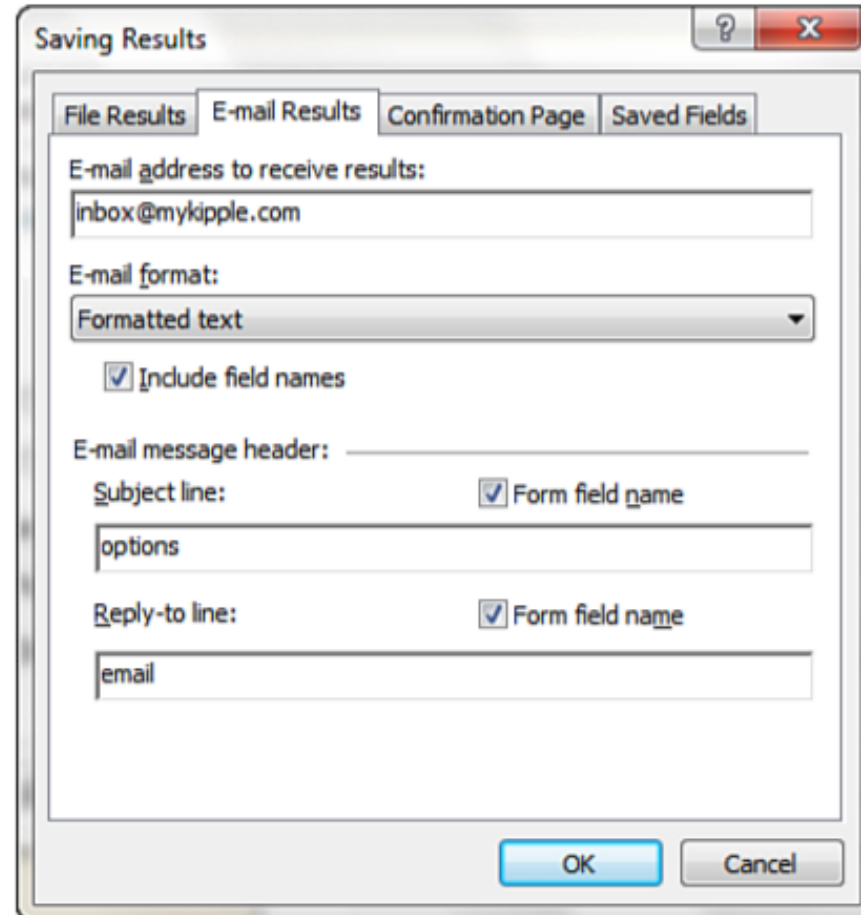
The screenshot shows a 'Saving Results' dialog box with the following elements:

- Tabbed interface with 'File Results' selected.
- 'File name:' field containing 'contact/emails.txt' and a 'Browse...' button.
- 'File format:' dropdown menu set to 'Formatted text'.
- Checkboxes for 'Include field names' and 'Latest results at end', both checked.
- 'Optional second file:' section with an empty 'File name:' field and a 'Browse...' button.
- 'File format:' dropdown menu set to 'HTML'.
- Checkboxes for 'Include field names' and 'Latest results at end', both checked.
- 'OK' and 'Cancel' buttons at the bottom.

Send Form Results Cont.

- Under the *E-mail Results* tab, you can configure what the email sent by the form will look like.
- The email address bar is self-explanatory, and the *E-mail Format* menu provides the same options as the *File Format* menu in the prior tab.
- Under *E-mail Message Header*, you can set the *Subject line* and the *Reply-to line* for the email generated.
- Each of these can be set to a standard line of text or to a form field name by checking the *Form Field Name* box.
- Check the *Form Field Name* box for the *Subject line* and insert the field name *options* (the drop-down menu).
- For the *Reply-to line*, check the *Form Field Name* box and enter the field name *email*.
- That way, the *Reply-To* address matches the address the visitor entered in the form.

Send Form Results Cont.



The screenshot shows a 'Saving Results' dialog box with the following configuration:

- File Results** | **E-mail Results** | Confirmation Page | Saved Fields
- E-mail address to receive results:
- E-mail format:
- Include field names
- E-mail message header: _____
- Subject line: Form field name
- Reply-to line: Form field name
- Buttons:

Send Form Results Cont.

- The *Confirmation Page* tab provides you with the option of sending the visitor to a specific page in your site after the file has been stored, email has been sent, or both. (step 8)
- This can be any page on your site, but it is normal to create a page with a message telling the visitor the process was successful.
- If you do not insert a link, the server generates a page for you automatically. (However, this page does not match the rest of your site.)
- Even though the page does not exist yet, set the *URL of Confirmation Page* to *Contact/confirmed.html*.
- If you have inserted validation scripts in your form (such as scripts to ensure that certain fields are filled out), you can also create a custom page to redirect the visitor to if the form is not properly filled out.

Send Form Results Cont.

Saving Results

File Results | E-mail Results | Confirmation Page | Saved Fields

URL of confirmation page (optional):

Page displayed when the user successfully submits the form.
A default page is provided.

URL of validation failure page (optional):

Page displayed when the user's input fails the validation scripts.
A default page is provided.

Send Form Results Cont.

- The last tab in the *Saving Results* dialog is *Saved Fields*.
- From here, you can decide what fields should be saved and submitted from the form.
- By default, Expression Web 4 inserts all the fields, including the *Input (Submit)* button.
- In almost every case, you want to remove this last instance because it is unnecessary. (The fact that the file has been saved or the email sent means that the Submit button was clicked.)
- To remove it, simply highlight the *submitButton* instance in the window and delete it.
- The *Saved Fields* tab also allows you to save the date and time of the form submission in various different formats that you can select from the drop-down menus.
- Finally, you can choose to save additional information about the user, including the remote computer name, username, and browser type. (Note that the username will be saved only if the user is required to log in before submitting the form.)

Send Form Results Cont.

The image shows a Windows-style dialog box titled "Saving Results". It has a standard title bar with a question mark icon and a close button (X). The dialog contains several sections:

- File Results | E-mail Results | Confirmation Page | Saved Fields**: A set of tabs at the top, with "Saved Fields" currently selected.
- Form fields to save:**: A text area containing the following fields:
 - fullName
 - email
 - options
 - comment
- Save All**: A button located below the text area.
- Date and time:**: A section with two dropdown menus:
 - Date format:** currently set to "(none)".
 - Time format:** currently set to "(none)".
- Additional information to save:**: A section with three checkboxes:
 - Remote computer name
 - Username
 - Browser type
- OK** and **Cancel**: Two buttons at the bottom of the dialog.

Send Form Results Cont.

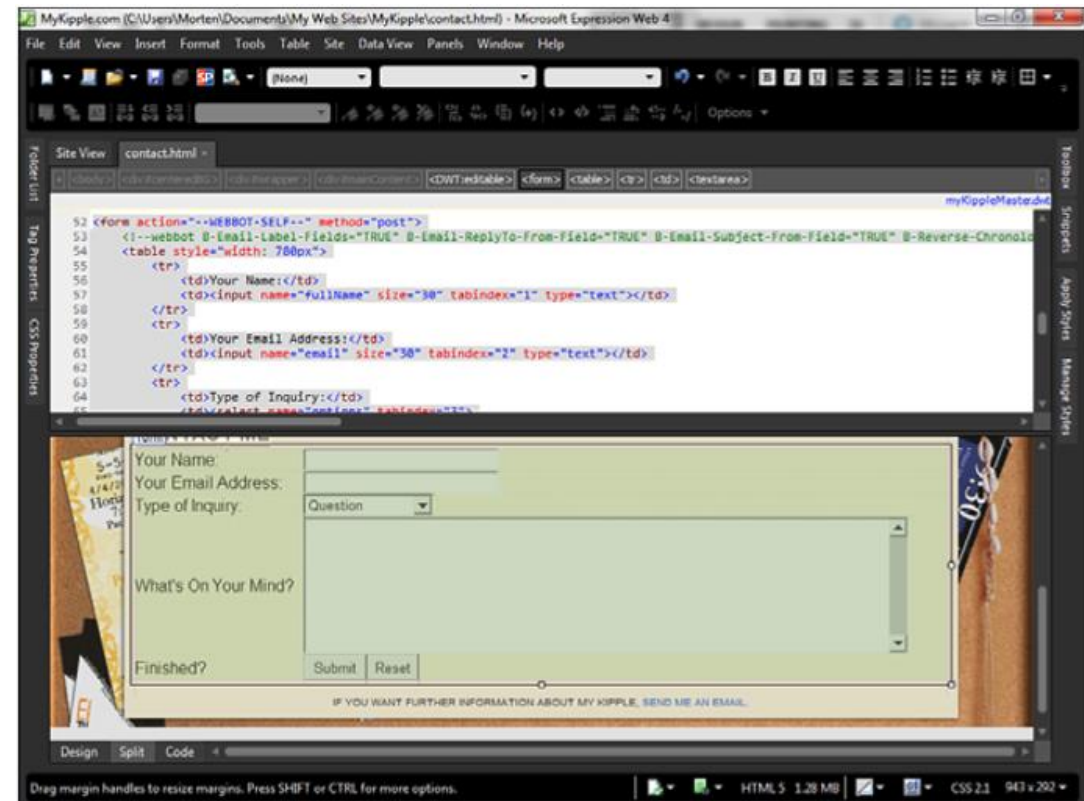
- When you click *OK* to save the changes, a warning appears and tells you that the file you are trying to link to does not exist and asks you whether you want to save the link anyway.
- This is because, in step 8, you entered a hyperlink to a page you have not created yet.
- Click *Yes*.
- When you click *OK* in the *Form Properties* dialog, another warning will likely appear, telling you that this form cannot be configured to send emails because it is at a disk-based location or on a server that has not been configured to send emails.
- This is because the email functionality you have configured requires server-side scripting in the form of FrontPage Server extensions.
- Your local computer will not have these capabilities installed.
- The warning asks whether you want to remove the email recipient.
- Click *No* to keep the email address in the script.
- After the page is uploaded to a server with FrontPage Server extensions, it will work properly.
- The email form with all the necessary functionality is now complete.

Other Uses for Form Results

- Creating an email form using forms is just the tip of the gigantic iceberg of possibilities that are available. Forms can be used for a wide range of functions that go far beyond simple text communication.
- If you open the *Form Properties* dialog again, notice that in addition to the *Send To* option, you can also select *Send to Database* or *Send to Other Scripts*.
- These functions let you use forms to communicate with and make alterations to databases.
- The databases can, in turn, manage content on a website or an online store, or even keep track of warehouse stocks for a company.
- The possibilities are virtually endless.
- Setting up a form with a database or script connection requires both a server that supports the chosen script language and a deeper understanding of the use of databases on the Web.

Forms in Code View

- Click anywhere inside the form while in *Split* view and look at the *Quick Tag Selector*.
- Depending on where you placed the cursor, the last few tags differ, but the main `<form>` tag should be the first one in the series.



Forms in Code View Cont.

- The `<form>` tag was the first element you inserted into the page, and all the form controls are contained within this tag.
- By looking at the form code in *Code* view, you can see that Expression Web 4 has organized it in such a way that it is easy to read.
- A closer look at each form control shows you that they are all based on a basic code formula.
- Click the first text box as an example.
- The code highlighted in *Code* view is simple to read:

```
<input name="fullName" size="30" tabindex="1" type="text" />
```

- As you can see, the form function is no different from any other HTML tag, and you insert the different attributes in the same way as any other tag.
- This also means that you can use the *Tag Properties* panel to make changes to the form functions.

Forms in Code View Cont.

- More than that, it also means you can apply CSS styles to the form controls just as you have done with other tags throughout this book.
- For example, by making a new style with the *selector* name *form*, you can set the *font family*, *font size*, *background color*, *box*, or any number of other style attributes for the forms.
- Alternately, you can set individual styles for the different form elements separately by applying classes to them.
- When you put them next to one another, you can get a clear idea of how exactly the code for each of the form controls works.

Forms in Code View Cont.

- Text box:

```
<input name="fullName" size="30" tabindex="1" type="text" />
```

- Drop-down box:

```
<select name="options" tabindex="3" >  
    <option selected="selected">Question</option>  
    <option>Comment</option>  
    <option>Suggestion</option>  
    <option>Random Thought</option>  
</select>
```

Forms in Code View Cont.

- Text area:

```
<textarea cols="70" name="comment" rows="8" tabindex="4"></textarea>
```

- Input (Submit):

```
<input name="submitButton" tabindex="5" type="submit" value="Submit" />
```

- Input (Reset):

```
<input name="resetButton" tabindex="6" type="reset" value="Reset" />
```