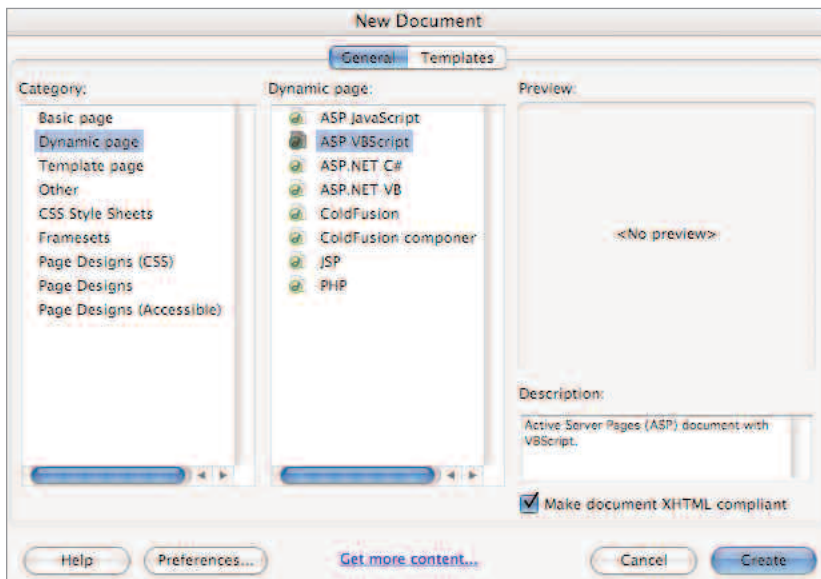


CS 631 • Advanced Web Design • Week 1 & 2: ASP Forms

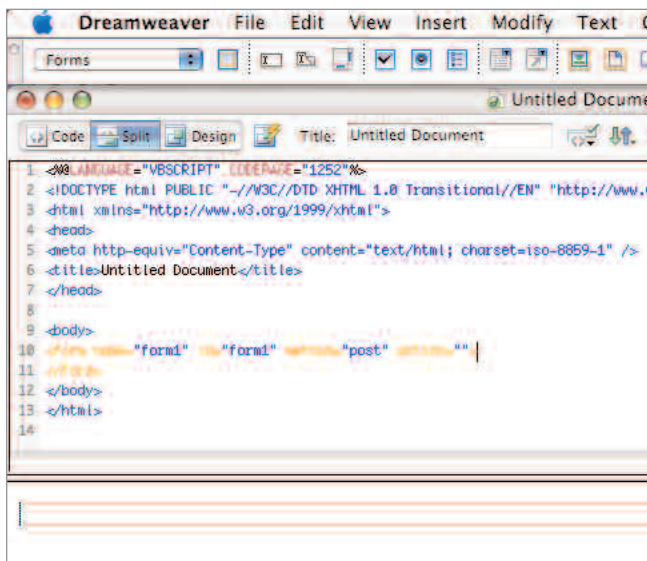
G. Chelak • students@gchelak.com

This exercise will utilize Active Server Pages (ASP) technology to execute a user-completed form. Using ASP means we will be creating a dynamic application (in this case a form) that will be processed on Testing Server. The results will be posted to the user's browser window. Data may be passed between pages many different ways & via different technologies; each with their own benefits & shortcomings. While each method may require a slightly different procedure, the concepts are the same.

1. File > New: choose Dynamic Page in the Category list, and ASP VBScript as the page type. Make sure to check the "Make document XHTML compliant" box in the lower right - this lets us do our 'state-of-the-web' CSS design while also supporting the older HTML 4.1 tags.

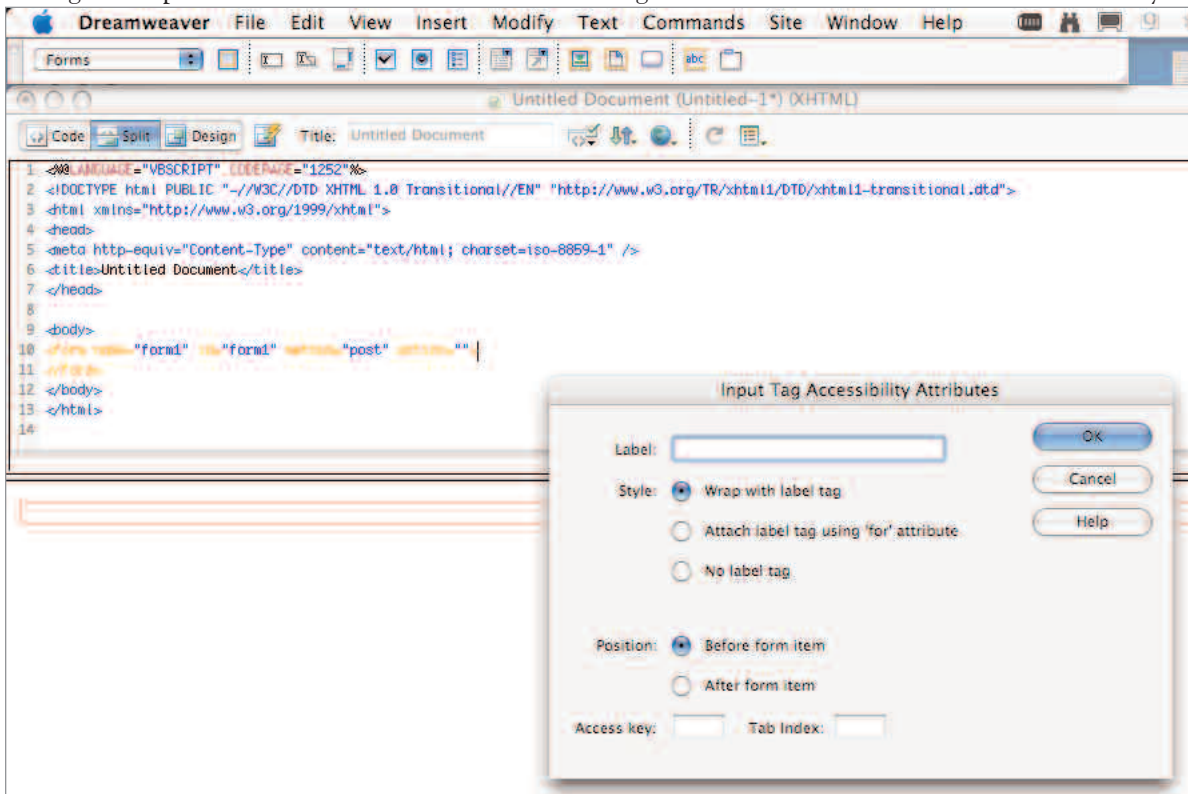


2. Place your cursor in the Design View. Select the Forms Category from the Insert Bar at the top of the screen. Click the Form icon (first from the left) to add a form to your page. This will appear as a dotted outline.

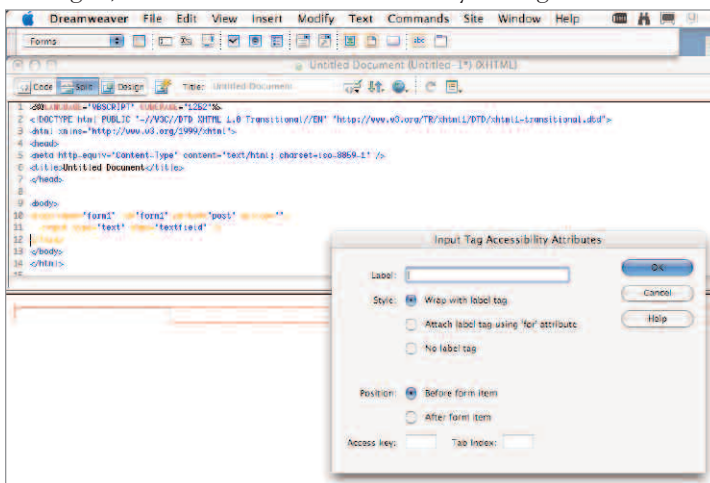


3. In the Design View, place the cursor inside the outlined form. Click the Text Field icon (Insert Bar > second from the left) to add a text field.

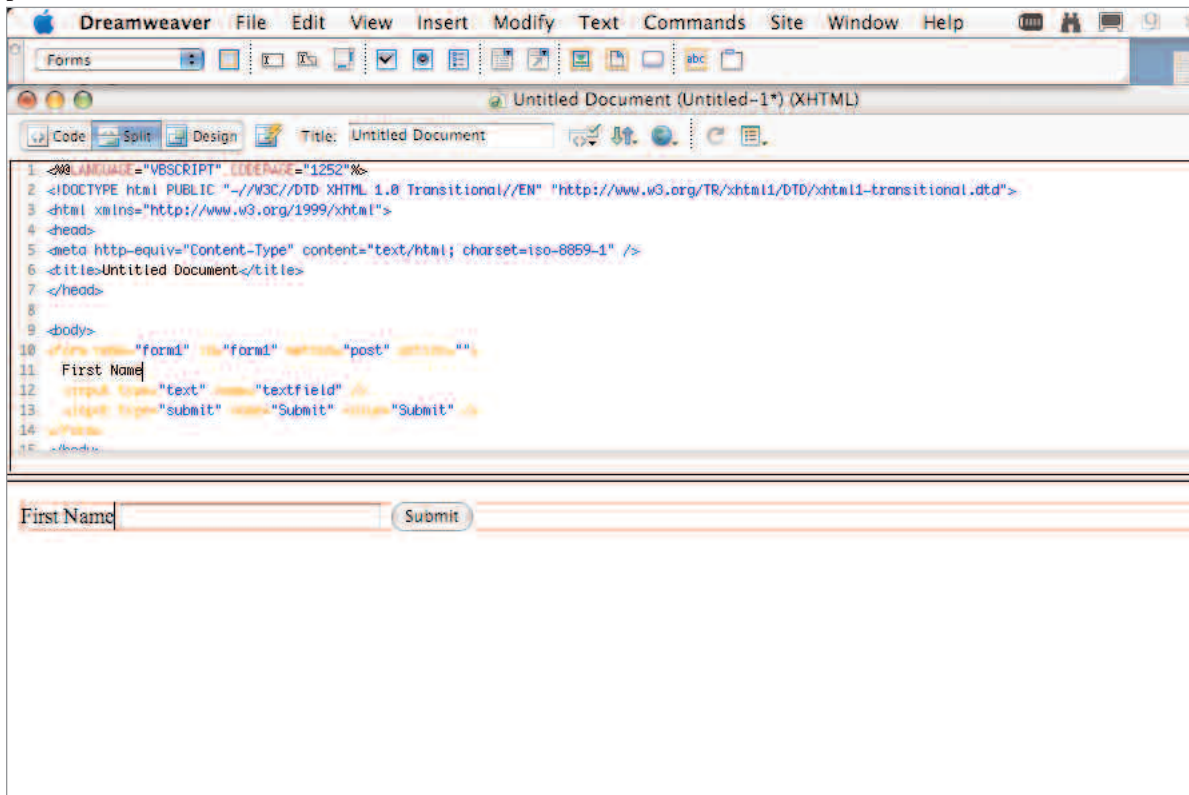
*Note: If your Accessibility Options are enabled in Dreamweaver's Preferences, you will also get the dialogue box shown below. In that case, add the label of "First Name" and choose "After form item" if you would like to label this form element. This labels the Text Field for the user. Choosing "OK" without filling in anything will work, though you will need to go back and create this label by hand as labels are necessary for screen readers to read web pages to blind or sight-impaired users. Though rarely (if ever) enforced, there is already legislation in place to require this type of accessibility from websites. From my industry experience, this seems more important to larger companies than smaller ones. It is something that should not be ruled out automatically.



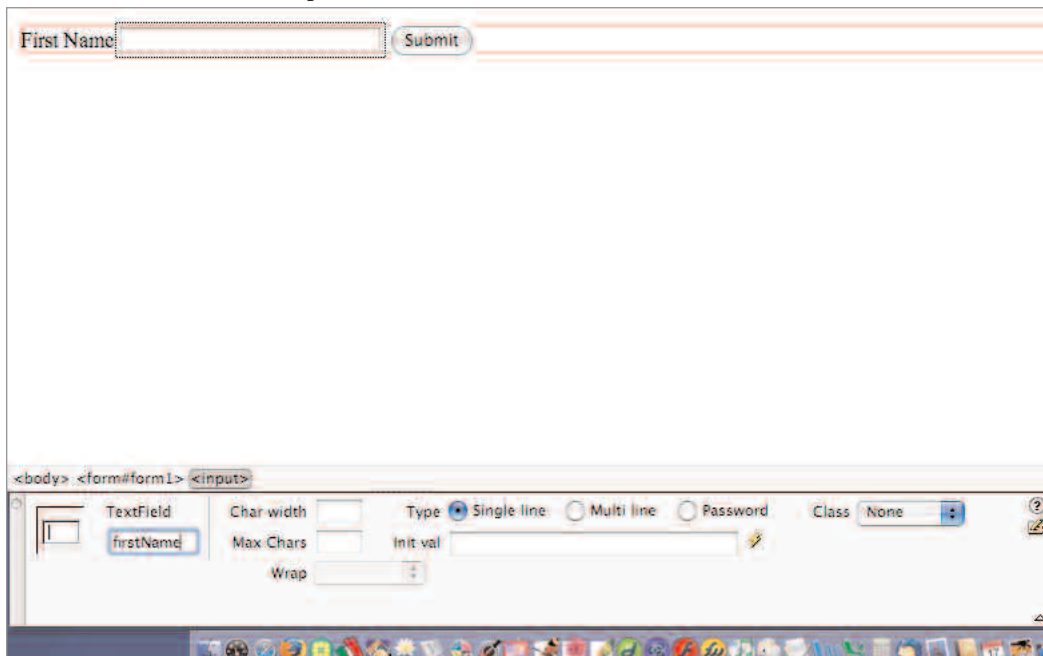
4. In the Design View, place the cursor after the Text Field and click the Button icon (Insert Bar > third from the right). Note the same Accessibility dialogue box.



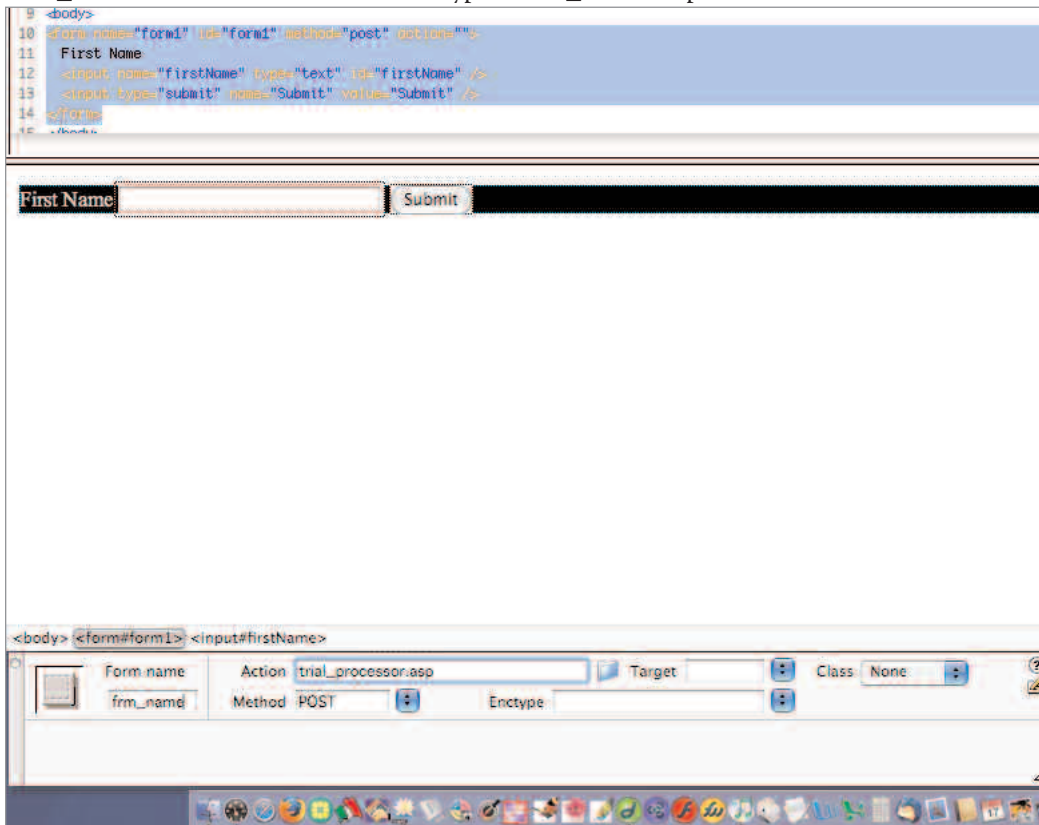
4.5. This is the result of adding the two prior form elements. If you did not get the Accessibility dialogues in the prior two steps, you must now enter the field label for at least the Text Field so users know what information to put in the field.



5. Click the Text Field in the Design View. The Properties Inspector now changes to show the available options. In the left side of this Inspector, name the text field “firstName” as shown.

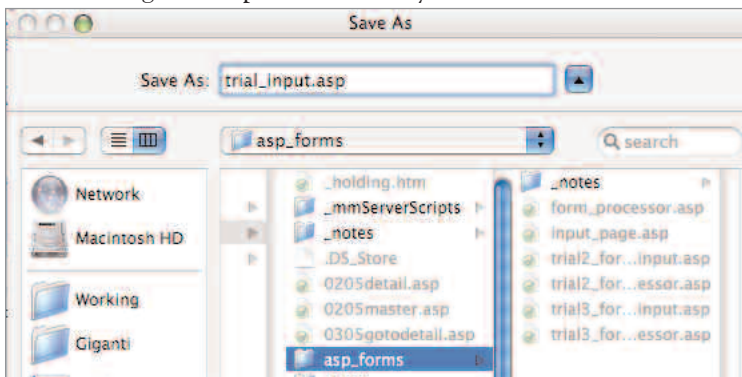


6. In the Tag Selector at the bottom edge of your document window, click on “<form#form1>” to select the entire form. Again, the Properties Inspector changes to reflect the available options. Name the form “frm_name” and in the Action field type “form_action.asp”. Make sure the Method is set to “Post”.



7. File > Save As... and name the file “form_input.asp”. Save this file inside your website’s local root folder.

*Note: I have chosen to create a subdirectory for this assignment inside my local root folder. When considering the vast scope of designing & developing large sites, being more organized like this in the structure of the local root folder can provide you huge time savings. An important point: in my experience, many sites that started out small have become much larger over time. A good structure ensured I was able to handle every one of these unforeseen growth spurts efficiently. This has translated directly to more money in my pocket.



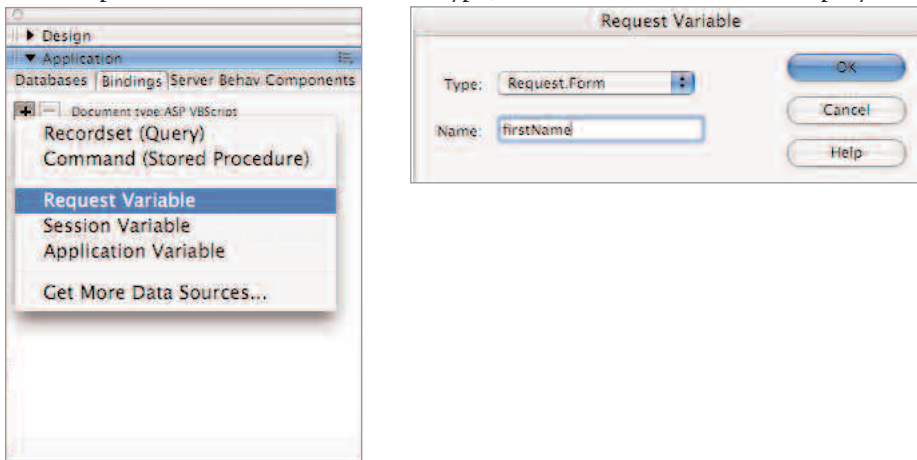
8. Create a new blank ASP VBScript page as in step #1.

9. Save this file as “form_action.asp”.

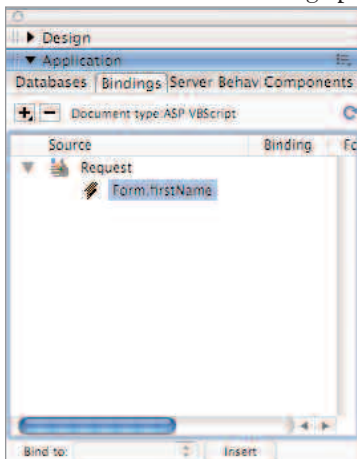
10. In the Design View, type “Thank you, , for filling out my form.” Drag-select this line of type. In the Properties Inspector, on the left side, choose the format of Paragraph to wrap this line of type inside paragraph tags. This will provide a tag set to place our ASP code within, isolating it from the rest of the page content.

11. Open the Application Palette, and choose the Bindings panel. Bindings are ways to get data sources. Click the “+” button and choose “Request Variable” from the available options.

Fill out the subsequent dialogue box as shown. This tells the server what type of data to look for, and in the case of multiple data sources of the same type, the name of the one to display.



12. The Bindings panel now displays the completed data source request. In the Design View, place the cursor immediately before the second comma. In the Bindings panel, click to select the “Form.firstName” request. At the bottom of the Bindings panel, click the “Insert” button.



13. Dreamweaver now displays the ASP code in the Code View, and Macromedia’s PsuedoCode in the Design View. Save the current file.

14. Upload both files to a Testing Server that supports ASP. Preview the “form_input.asp” page & fill in your first name in the Text Field. Click the Submit button.

15. You are taken to the “form_action.asp” page, which now displays your first name just before the second comma instead of the ASP code or the PsuedoCode. Congratulations! You have created a web application that has been executed on the Testing Server.

16. View the source on the current page. There is no ASP code. In fact, the line thanking you for filling out the form appears to have been created with your name in it. The first page sent the information to the Testing Server, which passed the information to the second page and executed the ASP code on that page. The server combined this result with the content already on the second page, and sent both to your computer to render in your web browser. This is how Dynamic Data Applications work.