TUTORIAL CHAPTER 9

**Exercise**

Your task is to design the form elements used to create the form shown in Figure 1.

**Figure 1. A simple form web page**



In addition to looking like the form shown in Figure 1, your form must include the constructs listed in the following instructions. When creating the code for your form, you must use the HTML5 tags that are appropriate to replicate the form and fulfill all the specifications listed.

1. Code the form with autocomplete active.
2. Given the image shown in Figure 1, it is easy to see that two field sets are used to create the main structure of the form. Your task is to create the field sets, including the names *Customer Info* and *Books.* Don't worry about the content fields for the moment.
3. The **Name** field you create should have autofocus, placeholder text, and be required. Don't forget to select the appropriate type for this field as well as all the fields that follow.
4. The **Telephone** field should have placeholder text, a pattern to restrict entry, and be required.
5. The **Email address** field should have placeholder text and allow multiple entries. This field should also be required.
6. The **Books** field should have a data list. You can select the content you would like to list.
7. The **Quantity (Maximum 5)** field should have a minimum value of 1 and a maximum value of 5.

**Website Case Study**

**JavaJam Coffee House**

**Task 1:** **Configure the CSS.** Modify the external style sheet (javajam.css) as indicated below:

1. Create a label element selector to float to the left with block display. Set the text alignment to right, assign a width of 8em, and set an appropriate amount of right padding.
2. Configure the input element and textarea element selectors with block display and 1em of bottom margin.
3. Configure a selector for an id named mySubmit that sets the left margin to 12em.
4. Save the javajam.css file.

**Task 2: Create the Jobs Page.** Use the Menu page as the starting point for the Jobs page. Launch a text editor and open menu.html. Save the file as jobs.html. Modify your jobs.html file to look similar to the Jobs page as follows:

1. Change the page title to an appropriate phrase.
2. The Jobs page will contain a paragraph and s form in the main element. Delete the table in the main element.
3. Add a paragraph that contains the following text: “Want to work at JavaJam? Fill out the form below to start your application.”
4. Prepare to code the HTML for the form area. Begin with a form element that uses the post method and the action attribute to invoke server-side processing. Configure the action attribute to send the form data to <http://webdevbasics.net/scripts/javajam.php>.
5. Configure the form control for the Name information. Crete a label element that contains the text “Name”. Create a text box named myName. Use the for attribute to associate the label element with the form control.
6. Configure the form control for the E-mail information. Create a label element that contains the text “E-mail:”. Create a text box named myEmail. Use the for attribute to associate the label element with the form control.
7. Configure the Experience area on the form. Create a label element that contains the text ‘Experience”. Create a textarea element named myExperience with rows set 2 and cols set to 20. Use the for attribute to associate the label element with the form control.
8. Configure the submit button. Code an input element with type=”submit” and value=”Apply now”. Assign the input element to an id named mySubmit.
9. Code an ending </form> tag on a blank line after the submit button.

If you are connected to the Internet, submit the form. This will send your form information to the server-side script configured in the form tag. A confirmation page that lists the form information and their corresponding names will be displayed.

**Task 4: Configure the HTML5 controls.** Modify the jobs.html file in a text editor.

1. Add the following sentence to the paragraph above the form: “Required fields are marked with an asterisk (\*).”
2. Use the required attribute to require the name, e-mail, and experience form controls to be entered. Add an asterisk at the beginning of each label text.
3. Configure the input element for the e-mail address with type=”email”.

Submit your form with missing information or only a partial e-mail address. Depending on the browser’s level of HTML5 support, the browser may perform form validation and display an error message.

