

Hands-On Exercises #3 – JavaScript

Source: Contents were adapted from Ch 9 & 14 of Felke-Morris, T. (2019). Web Development and Design Foundations with HTML5. 9th ed. USA: Pearson

Exercise 3-1 Adding a Form to JavaJam Coffee House

Continue from previous week, in this exercise you will create new Jobs page that contains a form. You have the following tasks:



Figure 3.1 JavaJam Jobs page

Task 1: The Website Folder.

Create a folder called javajam3_1. Copy all of the files from the “javajam_this_week” folder into the javajam3_1.

Task 2: Configure the CSS.

Modify the external style sheet (javajam.css). Open javajam.css in a text editor. Review Figure 3.1 and the wireframe in Figure 3.2. Notice how the text labels for the form controls are on the left side of the content area but contain right-aligned text. Notice the empty vertical space between each form control. Configure CSS as indicated below:

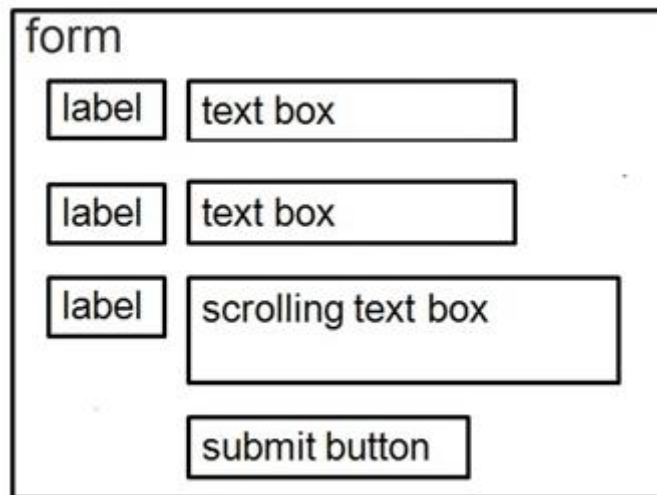


Figure 3.2 Wireframe for the form

1. Create a form element selector with a style declaration that sets 2em of padding. Configure 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 1em 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 9.5em.
4. Configure a selector for an id named herojobs with 250px height. Set the background image to herojobs.jpg. Also configure background-size: 100% 100%.

Save the javajam.css file

Task 3: 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 (shown in Figure 3.1) as follows:

1. Change the page title to an appropriate phrase.
2. The Jobs page will contain an h2, a paragraph, and a form in the main element. Assign the div to an id named herojobs. Delete the table in the main element.
3. Edit the text within the h2 element to say "Jobs at JavaJam". Replace the text in the paragraph with the following: "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/javajam8.php>.
5. Configure the form control for the Name information. Create 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 to 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 tag on a blank line after the submit button. Save your file and test your web page in a browser. It should look similar to the page shown in Figure 3.1. 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 HTML5 Form Controls.

Get more practice with the new HTML5 elements by modifying the form on the Jobs page to use HTML5 attributes and values. 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"`.

Save your file and display your web page in a browser. Submit the 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. Figure 3.3 shows the Jobs page rendered in the Firefox browser with an incorrectly formatted email address.

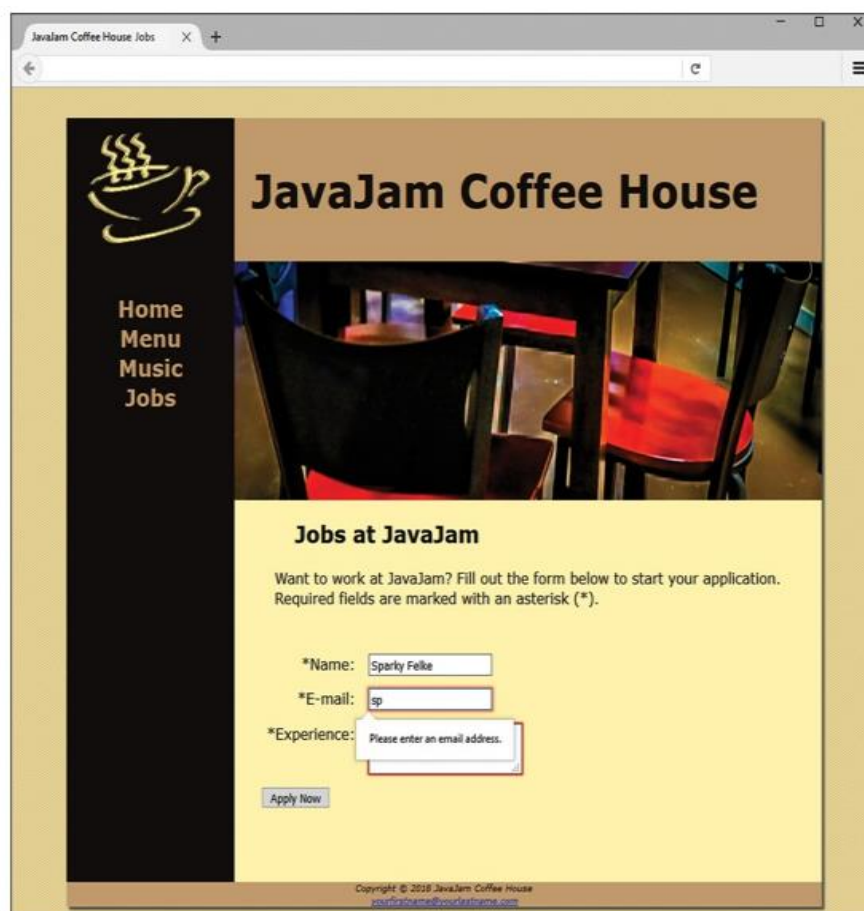


Figure 3.3 The Jobs page with HTML5 form controls

Exercise 3-2 Adding JavaScript to JavaJam Coffee House

Use the web pages indicated in this exercise from the Chapter 9 javajam9 folder. You have three tasks in this case study:

Task 1: The Website Folder.

Create a folder called javajam3_2. Copy all the files from the javajam3_1 folder into the javajam3_2 folder.

Task 2: Add a Date to the Home Page.

Launch a text editor and open the index.html file. You will add the date that the document was last modified to the bottom of the index.html page.

1. Using JavaScript. Modify the page as follows:
 - a. In the footer area after the e-mail link, add a script block contained within a div element that will write the following message to the document:

“This page was last modified on: date”

- b. Use the document.lastModified property to print the date.

Save the index.html page and test it in the browser. You should see the new information in the footer area below the e-mail address.

Task 3: Display an Alert Message on the Menu Page.

Launch a text editor and open the menu.html file. You will add JavaScript to the menu.html page so that an alert message will pop up when the user places the mouse over the phrase “Mug Club”. The alert message will indicate “JavaJam Mug Club Members get a 10% discount on each cup of coffee!”

1. Using Javascript. Modify the page as follows:
 - a. Add a hypertext link to the first paragraph with an onmouseover event handler as follows:

```
<a href="#" onmouseover="alert('JavaJam Mug Club Members get a  
10% discount on each cup of coffee!');">Mug Club</a>
```

Save the menu.html page and test it in the browser. Figure 3.4 shows the alert message when users place their mouse pointer over the hyperlinked phrase.



Figure 3.4 JavaJam Menu page with the mouseover alert