PRACTICE

Practice the skills you learned in the tutorial using the same case scenario.

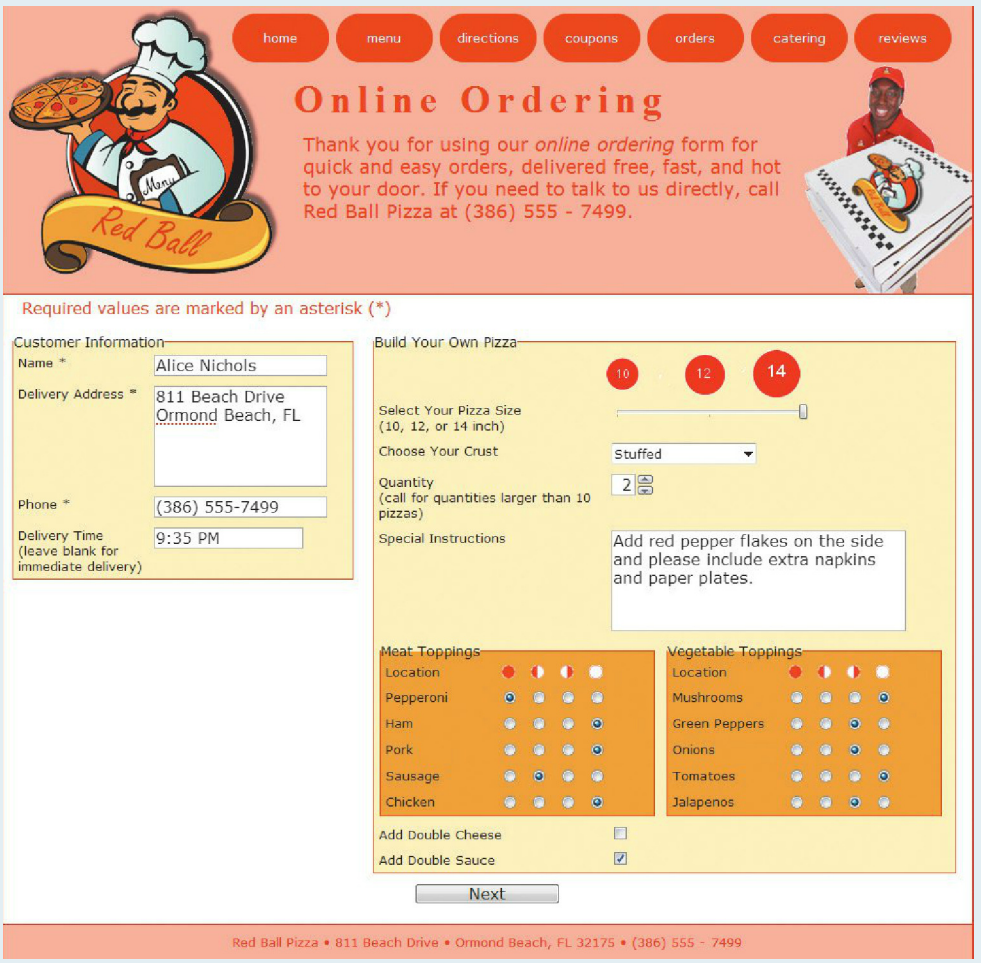
Review Assignments

Data Files needed for the Review Assignments: delivery.png, formsubmit.js, full.png, left.png, modernizr-1.5.js, none.png, okay.png, ordertxt.htm, pizzatxt.css, redball.css, redball.png, regex.txt, right.png, sizes.png, warning.png

Alice wants you to start working on an order form for customers who want to place their orders online using the Red Ball Pizza Web site. She suggests that you create a prototype page in which customers can enter their contact information for delivery and provide the ingredients for the pizza they want Red Ball Pizza to make for them. Alice wants you to validate the Web form as much as possible before it is sent to the Web server. She has created the file *regex.txt*, which contains regular expression patterns for validating the customer's phone number and preferred time of delivery.

A preview of the form you'll create is shown in Figure 6-77 as it appears in the Opera browser.

Figure 6-77 Build Your Own Pizza form



Timea/Shutterstock.com
Stephen Coburn/Shutterstock.com

Complete the following:

1. Use your text editor to open the **ordertxt.htm** and **pizzatxt.css** files from the tutorial.06\review folder included with your Data Files. Enter ***your name*** and ***the date*** within the comment section of each file, and then save them as **order.htm** and **pizza.css**, respectively.
2. Go to the **order.htm** file in your text editor. Link the file to the **pizza.css** style sheet file.
3. Directly below the paragraph in the `section` element, insert a `form` element with the name and id `pizza` that has the action *http://www.redballpizza.com/cgi-bin/buildpizza* and uses the `post` method.
4. Create two field sets with the ids `custInfo` and `buildPizza`, and with the legend text **Customer Information** and **Build Your Own Pizza**, respectively.
5. Within the `custInfo` field set, create a label with the text **Name** * along with a text input box for the `custname` field. Add the placeholder text **First and Last Name** and make the field required.
6. Create a text area box for the `address` field along with the label **Delivery Address** *. Make the `address` field required.
7. Create a label containing the text **Phone** * and text input box for the `phone` field. Add the placeholder text **(nnn) nnn-nnnn** and make the field required. The text of the input box should follow the regular expression pattern `^\d{10}$|^(\(\d{3}\)\s*)?\d{3}[\s-]?\d{4}$`.
8. Create a text input box for the `delTime` field with the label text **Delivery Time (leave blank for immediate delivery)**, with the parenthetical text placed on a new line. Add the placeholder text **hh:mm AM/PM** and specify the regular expression pattern `^([0-9]|1[012]):[0-5][0-9]\s?((a|p)m|(A|P)M)$`.
9. Within the `buildPizza` field set, add the *sizes.png* file as an inline image. Add the id `sizeimage` to the inline image.
10. Create a range slider for the `size` field along with the label text **Select Your Pizza Size (10, 12, or 14 inch)**, placing the parenthetical text on a new line. Set the default value to 12, the minimum value to 10, the maximum to 14, and the step value to 2.
11. Create a selection list for the `crust` field along with the label text **Choose Your Crust**. The selection list should display the option text **Thin**, **Thick**, **Stuffed**, and **Pan** with option values equal to the option text.
12. Create a number spinner for the `quantity` field with the label text **Quantity (call for quantities larger than 10 pizzas)**, with the parenthetical text on a new line. The field has a default value of 1 and ranges from 1 to 10 increments of 1 unit.
13. Create a text area for the `instructions` field with the label text **Special Instructions**.
14. Within the `buildPizza` field set, create two field sets with the ids `meat` and `vegetables`, and with the legend text **Meat Toppings** and **Vegetable Toppings**, respectively.
15. Within the `meat` field set, add a label with the text **Location** followed by four inline images for the *full.png*, *left.png*, *right.png*, and *none.png* files with the alternate text **full**, **left**, **right**, and **none**, respectively.
16. Create a group of four option buttons for the `pepperoni` field with the field values **full**, **left**, **right**, and **none**, respectively. Make the *none* option checked by default. Nest the four option buttons within a `fieldset` element belonging to the *optionGroup* class.
17. Repeat the previous step for the remaining meat toppings, Ham, Pork, Sausage, and Chicken, naming the fields `ham`, `pork`, `sausage`, and `chicken`, respectively.
18. Repeat Steps 15 through 17 for the vegetable toppings within the `vegetables` field set, creating option groups for Mushrooms, Green Peppers, Onions, Tomatoes, and Jalapenos, and naming the fields `mushrooms`, `green peppers`, `onions`, `tomatoes`, and `jalapenos`, respectively.

19. Directly after the `vegetables` field set, create a check box for the `doubleCheese` field with the label text **Add Double Cheese**. Create another check box for the `doubleSauce` field with the label text **Add Double Sauce**.
20. Directly after the `buildPizza` field set, create a submit button with the button text **Next**.
21. Save your changes to the file, and then go to the **pizza.css** file in your text editor and create the styles described in the following steps.
22. Display all field sets with a background color value of (255, 246, 205) and with a solid 1-pixel border with the color value (233, 69, 0). Float the field sets on the left with a 1% margin. Set the width of the `custInfo` field set to 35%, the width of the `buildPizza` field set to 60%, and the widths of the `meat` and `vegetables` field sets to 47% each. Set the background color of the `meat` and `vegetables` field sets to the (237, 178, 74) color value.
23. Set the font size of the field set legends to 0.9 em.
24. Display all labels as blocks with a font size of 0.8 em. Float the labels on the left only when the left margin is clear. Set the label width to 40% of the containing element. Set the top and bottom margins to 5 pixels and set the size of the left padding space to 5 pixels.
25. Display all `input` elements and `textarea` elements as blocks floated on the left. Set the width to 50% with top and bottom margins of 5 pixels. Set the height of the `textarea` elements to 100 pixels.
26. Display all inline images nested within a form as blocks floated on the left with top and bottom margins of 5 pixels.
27. Set the width of the `delTime` input box to 150 pixels.
28. Set the left margin of the `sizeimage` inline image to 40%. Set the width of the `size` field range slider to 200 pixels. Make the background of the `size` field range slider transparent.
29. Float the selection list for the `crust` field on the left with a font size of 0.8 em. Set the top and bottom margins to 5 pixels and the width to 150 pixels.
30. Set the width of the spinner control for the `quantity` field to 40 pixels.
31. Set the width of `fieldset` elements that belong to the *`optionGroup`* class to 50%. Remove the border from the field set and make the background transparent.
32. Set the width of radio buttons to 30 pixels and the width of check boxes to 20 pixels.
33. Set the width of the submit button to 150 pixels, set the `float` property to `none`, and set the top and bottom margins to 0 pixels and the left and right margins to `auto`.
34. If an `input` element, `select` element, or `textarea` element receives the focus, set the background color to the value (220, 255, 220).
35. If an `input` element receives the focus and is valid, set the background color to the value (220, 255, 220) displaying the background image *okay.png* at the bottom-right corner with no tiling. Size the background image so that it's contained within the input box.
36. Repeat Step 35 for `input` elements that receive the focus and are invalid, setting the background color to the value (255, 232, 233) and the background image to the file *warning.png*.
37. Save your changes to the file and then load the **order.htm** file in your Web browser, preferably a browser that has good support for HTML5 forms such as Opera or Google Chrome. Test the form by confirming that it shows warnings for all invalid data values and for required fields that have no values.
38. Submit your completed files to your instructor, in either printed or electronic form, as requested.