

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

PRACTICE

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

The screenshot shows a web page titled "Build Your Own Pizza form". At the top, there's a navigation bar with links: home, menu, directions, coupons, orders, catering, and reviews. To the left, there's a cartoon illustration of a chef holding a pizza, with a banner below him that says "Red Ball". To the right, there's another illustration of a delivery person carrying a pizza box. The main content area is titled "Online Ordering" and includes a message: "Thank you for using our *online ordering* form for quick and easy orders, delivered free, fast, and hot to your door. If you need to talk to us directly, call Red Ball Pizza at (386) 555 - 7499." Below this, there's a note: "Required values are marked by an asterisk (*)".

Customer Information

Name *	Alice Nichols
Delivery Address *	811 Beach Drive, Ormond Beach, FL
Phone *	(386) 555-7499
Delivery Time (leave blank for immediate delivery)	9:35 PM

Build Your Own Pizza

Select Your Pizza Size (10, 12, or 14 inch): 10 12 14

Choose Your Crust:

Quantity (call for quantities larger than 10 pizzas):

Special Instructions: Add red pepper flakes on the side and please include extra napkins and paper plates.

Meat Toppings

Location	Pepperoni	Ham	Pork	Sausage	Chicken
Location	●	●	●	●	●
Pepperoni	●	●	●	●	●
Ham	●	●	●	●	●
Pork	●	●	●	●	●
Sausage	●	●	●	●	●
Chicken	●	●	●	●	●

Add Double Cheese

Add Double Sauce

Vegetable Toppings

Location	Mushrooms	Green Peppers	Onions	Tomatoes	Jalapenos
Location	●	●	●	●	●
Mushrooms	●	●	●	●	●
Green Peppers	●	●	●	●	●
Onions	●	●	●	●	●
Tomatoes	●	●	●	●	●
Jalapenos	●	●	●	●	●

Next

Red Ball Pizza • 811 Beach Drive • Ormond Beach, FL 32175 • (386) 555 - 7499

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.

Apply your knowledge of Web forms and CSS to create a comments form for a sports blogging page.

APPLY**Case Problem 1**

Data Files needed for this Case Problem: **blog.css**, **commtxt.css**, **formsubmit.js**, **go.png**, **modernizr-1.5.js**, **sblogo.jpg**, **sblogtxt.htm**, **stop.png**

Sblogger Sports enthusiast Steve Lavent is working on a Web site called *Sblogger* that will contain articles and commentary on the world of sports. He's asked for your help in designing the page on which users can submit their own comments on the articles that Steve posts to his blog. Your job will be to create a prototype for the commentary form, and to include HTML code and CSS styles to help validate user input. A preview of the Web form you'll create is shown in Figure 6-78.

Figure 6-78 Sblogger comments form

The screenshot shows a web browser displaying a blog post titled "Jenkins on Ice". The post discusses NBA star Dennis Jenkins signing a contract for "Long Sleep" to be revived after death. It includes a sidebar with recent articles and a comment form for users to post their own thoughts.

Recent Articles:

- Basebrawl (Comments: 10)
- Say that Pent! (Comments: 7)
- The Wages of Dim (Comments: 15)
- Worst Play Evah! (Comments: 12)
- Give me my Hall Pass (Comments: 23)
- Go to the Back of the Bloodline (Comments: 19)
- You ain't da Man (Comments: 24)
- Yips and Yaps (Comments: 21)
- Monument in Gouda (Comments: 18)
- Next Time Rent It! (Comments: 33)
- That's not a Sport (Comments: 25)

Post Your Comments

Username: Spider3014
E-mail: spider3014@spdx.mail.com
Password: [REDACTED]
Comment (500 character limit): I remember when Jenkins told an interviewer at halftime after going 0 for 5 in the first half, that all he needed was to hit his next 6 shots to get his shooting percentage above 100% for the entire game.

Submit Comment

Complete the following:

- In your text editor, open the **sblogtxt.htm** and **commtxt.css** files from the tutorial.06\case1 folder included with your Data Files. Enter **your name** and **the date** in the comment section of the file. Save the files as **sblogger.htm** and **comments.css**, respectively.
- Go to the **sblogger.htm** file in your text editor and review the contents and structure of the document. Link the file to the **comments.css** style sheet.
- Scroll down to the **h2** heading title **Post Your Comments**, and directly below this heading insert a **form** element with the action **www.sblogger.cgi-bin/subcomments** using the **post** method.
- Within the **form** element, create a field set with the name and id **commentFS**.
- Create a **label** element with the text **Username**. Within the **label** element, insert an input box for the **username** field. Make the field required and add the title **Supply your username**.

 EXPLORE

6. Create a `label` element with the text **E-mail**. Within the `label` element, insert an input box for the `email` field. Set the data type to `email` and make the field required. Add the title **Supply a valid e-mail address**.
7. Create a `label` element with the text **Password** containing an input box for the `password` field. Set the data type to `password` and make the field required. Add the title **You must provide your password**.
8. Create a `label` element with the text **Comment (500 character limit)**, placing the parenthetical text on a new line, and within the `label` element insert a text area box for the `commentBox` field. Limit the number of characters that can be typed into the text area box to 500 characters.
9. Add a submit button with the text **Submit Comment**.
10. Save your changes to the file and then go to the `comments.css` file in your text editor. Create style rules for the styles described in the following steps.
11. Display the field set with a background color of (245, 245, 255). Set the top and bottom margins to 15 pixels and the left and right margins to `auto`. Add 5 pixels of padding and set the width of the field set to 90%.
12. Display labels as blocks floated on the left once the left margin is clear of elements. Set the font size to 0.9 em and the width to 100%. Set the top and bottom margins to 5 pixels.
13. Display input boxes and text area boxes as blocks floated on the right margin. Set the font size to 0.9 em and the width to 55%. Set the left and right margins to 10 pixels. In addition, set the height of text area boxes to 150 pixels.
14. Display the submit button as a block once both margins are clear. Do not float the submit button. Set the dimensions of the button to 200 pixels wide by 30 pixels high. Set the top and bottom margins to 5 pixels, and the left and right margins to `auto`.
15. When input boxes and text area boxes receive the focus, change the background color to the value (225, 225, 240).
16. When input boxes receive the focus and are valid, change the background color to (225, 240, 225) and display the background image `go.png` in the bottom-right corner of the box with no tiling. Resize the background image so that it is contained within the input box.
17. When input boxes receive the focus and are invalid, change the background color to (240, 225, 225) and display the background image `stop.png` in the bottom-right corner of the box with no tiling. Resize the background image so that it's contained within the input box.
18. Close the style sheet file, saving your changes.
19. Open `sblogger.htm` in your Web browser and test the form. Verify that you cannot leave the `username`, `email`, and `password` fields blank and still submit the form. Also verify that the form is rejected if an invalid e-mail address is submitted in the `email` field.
20. If you have access to the Google Chrome browser, test the Web page in that browser and verify that the title text you entered for the input boxes in Steps 5 through 7 is included in the error box supplied by the browser when those input boxes fail the validation test.
21. Submit your completed files to your instructor, in either printed or electronic form, as requested.

 EXPLORE

Apply your knowledge of Web forms and CSS to create a payment form for a salt and spice online grocery store.

APPLY**Case Problem 2**

Data Files needed for this Case Problem: `diners.png`, `discover.png`, `formsubmit.js`, `go.png`, `master.png`, `modernizr-1.5.js`, `paytxt.css`, `regex.txt`, `sb.css`, `sback.png`, `sblogo.png`, `spicetxt.htm`, `state.txt`, `stop.png`, `visa.png`

The Spice Bowl Rita Sato is the manager of the Web development team for The Spice Bowl, a new online grocery store specializing in gourmet spices. She's asked you to work on the Web forms for the site. The first form you'll create is a payment form in which customers enter their billing address and credit card data. She wants you to include validation tests for customers' postal codes, phone numbers, e-mail addresses, and credit card numbers. A preview of the form you'll create is shown in Figure 6-79.

Figure 6-79 The Spice Bowl payment form

The screenshot shows a web page titled "The Spice Bowl". On the left is a vertical navigation menu with links like Home, Keyword Search, Spices, Seasonings, Blends, Salts & Peppers, Popcorn Seasonings, Dip Mixes, Bulk Spices, Extracts, Flavorings, Spice Jars, Spice Jar Labels, Spice Sets, Mortar & Pestles, Cookbooks, Online Recipes, and Forums. Below these are links for My Account, Checkout, Order History, Shipping Info, Tech Support, Hours, and Contact Us. The main content area has a header "Payment Form". It contains three sections: "Billing Information (required)" with fields for First Name (Rita), Last Name (Sato), Street Address (871 Park St.), Street Address (2), City (East Point), State (GA), ZIP/Postal Code (30344), Country (United States), and Phone ((404) 555-8711); "Credit Card (required)" with fields for Credit Card Number (5112345678901235), Expiration Date (January 01), CSC (001), and logos for American Express, Discover, MasterCard, and Visa; and "Additional Information" with fields for Contact Email (rita.sato@thespicewbowlcompany.com) and Special Notes (Please box the spices and the salts separately). At the bottom is a "Submit Order" button. To the right of the form, there are two sidebar sections: "The Spice Bowl Recommends" with a recipe for Salt-Roasted Pecans (2 cups pecans, 3 Tbs. butter, melted, 1 1/4 tsp. fine sea salt) and instructions to preheat oven to 325°, toss pecans and butter together, add salt and toss again, spread in a single layer on a baking sheet, bake about 15 minutes, cool on baking sheet; and "Dried Beef Sticks" with a list of ingredients (5 lbs. ground chuck, 5 Tbs. quick salt, 3 tsp. mustard seed, 2 tsp. granulated garlic, 2 tsp. cracked pepper, 1 Tbs. Hickory smoked salt, 1 Tbs. liquid smoke) and instructions to mix beef and season, place in a large covered container for three days, make sure beef is thoroughly mixed each day. On third day, shape beef into sticks, and bake at 150° for 8 hours (turn half-way through).

Complete the following:

- In your text editor, open the `spicetxt.htm` and `paytxt.css` files from the `tutorial.06\case2` folder included with your Data Files. Enter **your name** and **the date** in the comment section of each file. Save the files as `spice.htm` and `payment.css`, respectively.
- Go to the `spice.htm` file in your text editor and review the contents and structure of the document. Link the file to the `payment.css` style sheet.
- Scroll down to the `h1` heading *Payment Form*, and below it insert a `form` element with the action `http://www.thespicewbowlcorp.com/cgi-bin/payment` using the `post` method.

4. Within the form, create three field sets with the ids **billing**, **creditcard**, and **info**, and with the legend text **Billing Information (required)**, **Credit Card (required)**, and **Additional Information**, respectively.
5. Within the **billing** field set, create input boxes for the customer's first name, last name, street address, second line of the street address, and city with the field names **fName**, **lName**, **street**, **street2**, and **city**, respectively. Add field labels for these input boxes as shown in Figure 6-79.
6. Create a selection list for the **state** field containing two-letter abbreviations of the state names. You can find a list of state abbreviations in the *state.txt* file. Add the label text **State**.
7. Create an input box for the **zip** field along with the label text **ZIP/Postal Code**. The pattern of the field value should match the regular expression for postal codes. You can use the regular expression patterns listed in the *regex.txt* data file.
8. Create an input box for the **country** field along with the label text **Country**. Make the default value of the field equal to the text string **United States**.
9. Create an input box for the **phone** field along with the label text **Phone**. The pattern of the field value should match the regular expression for phone numbers.
10. All of the fields within the **billing** field set should be marked as required fields with the exception of the **street2** field.
11. Within the **creditcard** field set, create a field set belonging to the class **optionGroup**. Within this field set, create four label elements. Within each label, create an option button belonging to the **ccard** field followed by an inline image belonging to one of four credit card companies. The field values associated with the four option buttons are *diners*, *discover*, *master*, and *visa*, and the inline images are *diners.png*, *discover.png*, *master.png*, and *visa.png*, respectively.
12. Directly below the **optionGroup** field set, create an input box for the **ccardnumber** field along with the label text **Credit Card Number**. The text of the field value should match the regular expression for credit card numbers found in the *regex.txt* file.
13. Add a label containing the text **Expiration Date** followed by a selection list for the **ccardmonth** field that contains the first entry **--Month--** followed by the text for each individual month in chronological order from **January (01)** through **December (12)**. Also add a selection list for the **ccardyear** field with the first entry **--Year--** followed by year values for **2014** through **2018**.
14. Add a label containing the text **CSC** followed by an input box for the **csc** field. The **csc** field should be limited to three characters, follow the regular expression pattern **^\d{3}\$** and display the placeholder text **nnn**.
15. Make all of the fields within the **creditcard** field set required fields.
16. Within the **info** field set, create a label containing the text **Contact Email** followed by an input box for the **email** field that uses the **email** data type.
17. Add a label containing the text **Special Notes** followed by a text area box for the **notes** field.
18. At the bottom of the form, directly above the closing **</form>** tag, insert a submit button with the button text **Submit Order**.
19. Save your changes to the file, and then go to the **payment.css** file and add styles rules for the different form elements as described in the following steps.
20. Display all field sets without borders and with a 5-pixel margin. Display all field set legends with a background color of (239, 198, 145) and a text color of black. Set the top and bottom margins to 10 pixels and indent the legend text 20 pixels. Set the width of the legend to 100% of the width of the field set. Finally, add rounded borders to the legend, 10 pixels in radius.

21. Display all labels as blocks, floated on the left margin once the left margin is clear. Set the font size to 0.8 em and align the label text to the right. Set the top and bottom margins to 2 pixels, and set the left and right margins to 5 pixels. Set the width of the labels to 25%.
22. Display all `input` elements as blocks, floated on the left with a font size of 0.9 em and a width of 60%. Set the top and bottom margins to 2 pixels, and set the left and right margins to 0 pixels.
23. Display all `select` elements as blocks floated on the left with a font size of 0.9 em and a margin of 2 pixels.
24. Display all `textarea` elements as blocks floated on the left with a font size of 0.9 em, a height of 75 pixels, and a width of 60%.
25. Set the left margin of the field set belonging to the `optionGroup` class to 25%. For `label` elements nested within the `optionGroup` field set: a) display the labels inline; b) do not float the labels; and c) set the label width to 100 pixels.
26. Display option buttons inline with no floating. Set the margin of the option buttons to 0 pixels and the width to 20 pixels.
27. Set the width of the input box for the `csc` field to 50 pixels.
28. For the submit button, add the following styles: a) set the background color to the value (239, 198, 145); b) do not float the input box; c) set the dimensions of the button to 150 pixels wide by 30 pixels high; d) set the top and bottom margins to 10 pixels, and set the left and right margins to `auto`; and e) add a rounded border with a 15-pixel radius.
29. When `input`, `select`, and `textarea` elements receive the focus, change the background color to the value (255, 218, 165).
30. When an `input` element receives the focus and is invalid, change the background color to the value (255, 245, 215) and display the `stop.png` background image file in the bottom-right corner of the input box with no tiling.
31. When an `input` element receives the focus and is valid, change the background color to the value (215, 255, 215) and display the `go.png` image in the bottom-right corner of the input box with no tiling.
32. Save your changes to the style sheet.
33. Open the `spice.htm` file in your Web browser. Verify that you cannot submit the form without all required fields filled out, and without phone number, e-mail address, and CSC values entered in the proper format.
-  EXPLORE 34. Verify the validation checks for the credit card number by confirming that the form rejects the following credit card numbers (which are not valid card numbers):
 - 31012345678901
 - 6012123456789019
 - 5912345678901235
 - 8123456789012349
-  EXPLORE 35. Further verify the validation checks for the credit card number by confirming that the form accepts the following credit card numbers (which are valid card numbers):
 - 30312345678901
 - 60111123456789019
 - 5112345678901235
 - 4123456789012349
36. Submit your completed files to your instructor, in either printed or electronic form, as requested.

Apply your knowledge of Web forms to design a form for a demo page displaying the impact of different CSS text styles.

CHALLENGE

Case Problem 3

Data Files needed for this Case Problem: `back.png`, `cssdemotxt.htm`, `cssforms.css`, `modernizr-1.5.js`, `rundemo.js`, `wm.css`, `wmlogo.png`

WidgetMage Anna Lopez is the founder of WidgetMage, a Web site that specializes in designing small demos and applications for Web sites. Anna has asked you to work on creating a CSS demo page in which users can interactively select style values and see their effects on the page. Anna already has the JavaScript code written to make the demo page work and a style sheet for the Web form. She wants you to finish her project by writing the HTML code for the Web form. A preview of the page you'll create is shown in Figure 6-80. In this page, you'll enter sample text into a text area box and then select style values from the form controls in the page. The page shown in Figure 6-80 is based on the Opera browser.

Figure 6-80 WidgetMage CSS demo page in Opera

The screenshot shows the WidgetMage CSS Demo page. On the left, there's a sidebar with a navigation menu. The main content area is titled "CSS Demo". It contains a form for selecting CSS styles. The form includes sections for "Fonts", "Colors", and "Sizes", each with dropdown menus and sliders. A preview box shows the rendered text "CSS demo page from WidgetMage" with applied styles. A text input field on the right says "Enter your sample text below".

Complete the following:

1. In your text editor, open the `cssdemotxt.htm` file from the `tutorial.06\case3` folder included with your Data Files. Enter *your name* and *the date* in the comment section of the file. Save the file as `cssdemo.htm`.
2. Review the contents and structure of the document. Link the file to the `cssforms.css` style sheet.
3. Directly below the paragraph within the `section` element, insert a `form` element. You do not have to specify an action or a method.
4. Create field sets with ids equal to `textstring`, `fonts`, `colors`, and `sizes`. Do not add a legend to the `textstring` field set. For the three remaining field sets, add the legend **F**onts, **C**olors, and **S**izes, respectively.

Note: In the steps that follow, make sure you add both an `id` attribute and a `name` attribute to each input box, selection list, or text area box, giving the same value to both attributes. Also, make sure that you enter the name-id values in lowercase letters. Finally, make sure that every selection list has both option text and option values set to the same text string.

EXPLORE

5. Within the `textstring` field set, insert a label containing the text **E**nter your sample text below, followed by a text area box. The text area box should have the name and id value `samplertext`. Set the text area so that it automatically receives the focus when the page is loaded, and set its tab index value to 1. Set the line wrap within the text area box to hard to preserve any line breaks within the text area box.

EXPLORE

6. Within the `fonts` field set, insert a label containing the text **F**ont **F**amily followed by a selection list for the `fontfamily` field. Add the following options to the selection list: **d**efault, **s**erif, **s**ans-**s**erif, **m**onospace, **c**ursive, **f**antasy, **A**rial, '**B**ook **A**ntiqua', '**C**ourier **N**ew', **G**eneva, **H**elvetica, **I**mpact, **P**alatino, and '**T**imes **N**ew **R**oman'. Set the option values equal to the option text, including the single quotes where required for the font name.

7. For the options created for the `fontfamily` selection list, enclose the generic font names in an option group with the label **g**eneric and the specific fonts within an option group with the label **s**pecific.
8. After the `fontfamily` selection list, insert a label with the text **F**ont **S**tyle. Add a selection list for the `fontstyle` field with the option text and option values equal to **n**ormal, **i**italic, and **o**blique.
9. Add a label with the text **F**ont **W**eight along with a selection list for the `fontweight` field containing the option values and option text **n**ormal and **b**old.

EXPLORE

10. Add a label with the text **T**ext **D**ecoration along with a selection list for the `textdecoration` field containing the option values and option text **n**one, **l**ine-**t**hrough, **o**verline, and **u**nderline.

11. Add a label with the text **T**ext **T**ransform along with a selection list for the `texttransform` field containing the option values and option text **n**one, **c**apitalize, **l**owercase, and **u**ppercase.

EXPLORE

12. Add a label with the text **F**ont **V**ariant along with a selection for the `fontvariant` field containing the option text and values **n**ormal and **s**mall-caps.
13. Within the `colors` field set, add a label with the text **F**ont **C**olor (**h**exadecimal). Add an input box for the `color` field. Set the data type of the input box to `color`. Set the default value to the text string `#000000`. Add the placeholder text `#rrggbb`.
14. Add a label containing the text **B**ackground **C**olor (**h**exadecimal) followed by an input box for the `backgroundcolor` field. Set the data type of the input box to `color`. Set the default value to the text string `#FFFFFF`. Add the placeholder text `#rrggbb`.
15. Within the `sizes` field set, add a label containing the text **F**ont **S**ize (**p**x) followed by a range slider for the `fontsize` field. Set the default value of the `fontsize` field to 14, the minimum to 8, and the maximum to 40. The range slider should move from the minimum to the maximum in steps of 1 unit.

16. Create range sliders for the next four items in the `sizes` field set. Figure 6-81 lists the label text, field names, and ranges of the range sliders in the CSS demo page. Add appropriate range labels before and after the range sliders as indicated in Figure 6-80.

Figure 6-81

Range sliders in the CSS demo page

Field	Label Text	Default	Range
fontsize	Font Size (px)	14	8 - 40 by 1
letterspacing	Letter Spacing (px)	0	0 - 10 by 1
wordspacing	Word Spacing (px)	0	0 - 10 by 1
lineheight	Line Height (em)	1	0 - 4 by 0.2
textindent	Text Indent (px)	0	0 - 10 by 1

 EXPLORE

17. Directly after the `sizes` field set, add a command button containing the text **Remove Styles** and the id `removestyles`.
18. Scroll to the top of the file. Within the `head` element, add a `script` element to connect to the `rundemo.js` file.
19. Save your changes to the file.
20. Open the `cssdemo.htm` file in a Web browser that supports range sliders. Test the form by entering sample text into the text area box near the top of the form. Verify that when you tab out of the text area box, the text appears in the rendering box at the bottom of the page.
21. Change the style of the rendered text by selecting options and values from the style controls on the form.
22. If you have access to Opera or another Web browser that supports the `color` data type, verify that you can change the font and background color using a color picker. If your browser does not support the `color` data type, change the colors by entering hexadecimal values for the font color and background color fields.
23. Submit your completed files to your instructor, in either printed or electronic form, as requested.

 EXPLORE

Test your knowledge of Web forms by creating an order form for an online computer store.

 RESEARCH**Case Problem 4**

Data Files needed for this Case Problem: `formsubmit.js`, `go.png`, `mclogo.jpg`, `modernizr-1.5.js`, `stop.png`

Millennium Computers You are employed at Millennium Computers, a discount mail-order company specializing in computers and computer components. Your supervisor, Sandy Walton, has asked you to create an order form Web page so that customers can purchase products online. Your order form is for computer purchases only. There are several options for customers to consider when purchasing computers from Millennium. Customers can choose from the following:

- Processor Speed: 3.2 GHz, 4 GHz, 5.2 GHz
- Memory: 1 GB, 2 GB, 4 GB, 8 GB
- Monitor Size: 15", 17", 19", 21"
- Hard Drive: 240 GB, 500 GB, 750 GB, 1 TB
- DVD Burner: yes/no
- Tuner Card: yes/no
- Media Card Reader: yes/no

Complete the following:

1. Use your text editor to create an HTML file named **pc.htm** and two style sheets named **mill.css** and **oform.css**. Enter *your name* and *the date* in a comment section in each file. Include any other comments you think will aptly document the purpose and content of the files. Save the files in the tutorial.06\case4 folder included with your Data Files.
2. Design a Web page for Millennium Computers. Insert any styles you create in the *mill.css* style sheet. You are free to use the *mclogo.jpg* file and whatever text or images you wish to complete the look and content of the Web page.
3. Within the **pc.htm** file, insert a Web form containing the following elements:
 - Input boxes for the customer's first name, last name, street address, city, state, zip code, and phone number. The field names should be **fName**, **lName**, **street**, **city**, **state**, **zip**, and **phone**, respectively.
 - Selection lists for the processor speed, memory, monitor size, and hard drive size. The field names should be **pSpeed**, **mem**, **monitor**, and **hd**, respectively. The option values should match the option text.
 - Option buttons for the DVD burner, tuner card, and media card reader options. The field names should be **dvd**, **tuner**, and **mCard**, respectively.
 - A check box for the **warranty** field that asks whether customers want the 24-month extended warranty.
 - A text area box requesting additional information or comments on the order.
 - Three form buttons: a submit button with the text **Send Order**, a reset button with the text **Cancel Order**, and a command button with the text **Contact Me**.
 - Add validation checks marking all of the customer contact fields as required.
 - Use regular expression patterns to ensure that each user enters his or her zip code and phone number in the correct format.
4. Name the form **corder** and add attributes so the form is submitted using the **post** method to the CGI script located at http://www.mill_computers.com/orders/process.cgi.
5. Create a style for your form in the **oform.css** style sheet. The layout and appearance of the form are up to you. It should include style rules to highlight input boxes that receive the focus, and it should employ inline validation for missing or incorrectly entered data.
6. Test your Web site on a variety of browsers to ensure your design works under different conditions.
7. Submit your completed files to your instructor, in either printed or electronic form, as requested.

ENDING DATA FILES

