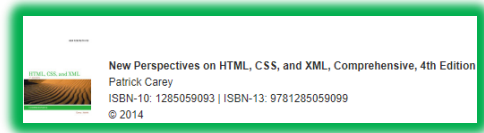


Tutorial Six Assignment Outline: Creating a Web Form

This PDF file consists of Eleven (11) pages

Please note: Students are strongly recommended to read **Tutorial Six** in Carey and Vodnik Textbook (HTML, CSS, and XML, 4th Edition).



Please carefully read the following pertinent information:

- **Due Date:** Sunday, March 07 at 11:30 p.m.
- **Submission Type:** Must be uploaded to the IVC Student Web Server located at (<https://st3w.ivc.edu>)
- **Naming Rule:** When submitting this tutorial project to IVC Student Web Server, students are required to use the naming rule (**t6-Red Ball Pizza**). Failure to do so will result in a 10% deduction for this assignment.
- **Default HTML Files:** For this tutorial project, the author provides you with default html files. These files carry the word “txt” in their name. The main purpose is to introduce **starting point** for developing the tutorial project. Open these files, remove the word “txt” as instructed in the grading rubric for this project, and saved them as HTML files, without the word “txt”. Failure to comply will result in a 10% deduction.

Tutorial – 6 Objectives

Students will have mastered the materials presented in Tutorial Six when they can:

Section 6.1 (Parts of a Web Form) <ul style="list-style-type: none"> • Introducing Web Forms (HTML 396 – 399) • Creating a Web Form (HTML 399 - 402) • Creating a Field Set (HTML 402 – 404) • Creating Input Boxes (HTML 404 – 408) • Adding Field Labels (HTML 408 – 410) • Applying a Style Sheet to a Web Form (HTML 411 - 415) • Defining Default Values and Placeholders (HTML 415 - 418) 	Section 6.2 (Selection Lists and Option Buttons) <ul style="list-style-type: none"> • Creating a Selection List (HTML 422– 429) • Creating Option Buttons (HTML 429 – 434) • Creating a Text Area Box (HTML 434 – 436) • Creating Check Boxes (HTML 437 – 440) Section 6.3 (HTML5 Data Types) <ul style="list-style-type: none"> • Exploring HTML5 Data Types (HTML 444– 453) • Suggesting Options with Data List (HTML 454 – 456) • Creating Form Buttons (HTML 456 – 460) • Validating a Web Form (HTML 460 – 466) • Applying Inline Validation (HTML 466 – 470)
--	---

Grading Rubric- Tutorial Six

IMPORTANT The following reflects an enhanced version of the author’s original grading rubric. The criteria described for each section, plus associated pages, listed below, are the closest proximity as an assessment tool, and may not completely convey the complete tasks that should be performed to obtain full credit for **Tutorial_6 project assignment**. It is of vital importance to closely follow and master the materials presented in the **Tutorial Six Chapter**.

Therefore, in addition to the criteria listed below, it is further expected to measure your complete work, For the assessment of any key components that may quantify and validate the performance of **Tutorial_6 project assignment**, based on;

1. Correct positioning and layout of all elements on the **Red Ball Pizza home page source code** will be inspected for correct spelling for accuracy of the work, presented. *
2. Final design of **Red Ball Pizza home page**, should resemble the **Fig. 6-65 on HTML 459**, as shown at the end of this document *

* Additionally, to achieve your desired results, it is reasonably expected your completed work shall comply with the statements as described in the above two steps. Failure to comply will result in an additional **10 % deduction**.

Description	Pts & Page#	Your Score
<p>Section 6.1 (Parts of a Web Form)</p> <p>Introducing Web Forms This section starts on HTML396 and extends to HTML399.</p> <ul style="list-style-type: none"> ➤ Open surveytxt.htm in text editor. Enter your name/the date in comment section. Save the file as survey.htm. ➤ Review the content and structure of survey.htm while opened as text. ➤ Verify that the initial appearance of the page is as shown in Fig. 6-1, when survey.htm is viewed in a web browser. 	<p>0.5 HTML 396 through HTML 399</p>	
<p>Creating a Web Form This section starts on HTML399 and extends to HTML 402.</p> <ul style="list-style-type: none"> ➤ To add the form element, open survey.htm in a text editor and insert the survey form element, directly above the closing </section> tag as shown in Fig.6-4. Save changes to the file <p style="text-align: center;">* * * *</p> <ul style="list-style-type: none"> ➤ Return to survey.htm and set the form attributes to the form element, as shown in Fig. 6-6. Save changes to the file. <p style="text-align: center;">* * * *</p> <ul style="list-style-type: none"> ➤ Return to survey.htm in a text editor. Go to the head section of the document. Directly below the script element that accesses the modernizert.js, link the formsubmit.js JavaScript program, as shown in Fig.6-7. Save changes to the file. 	<p>0.5 HTML 399 through HTML 402</p>	
<p>Creating a Field Set This section starts on HTML402 and extends to HTML 404.</p> <ul style="list-style-type: none"> ➤ Within the form element in survey.htm, insert the custInfo/experience fieldsets, as shown in Fig. 6-8. Save changes to the file. ➤ Within the first fieldset in the survey.htm file, insert the Customer Information legend element. ➤ In the second fieldset, insert Share Your Experience at Red Ball Pizza Legend, as shown in Fig.6-9. Save changes to the file. ➤ Refresh the survey.htm in a web browser and view the current appearance of the form with the fieldsets and legend, as shown in Fig.6-10. 	<p>0.5 HTML 402 through HTML 404</p>	

<p>Creating Input Boxes This section starts on HTML404 and extends to HTML 408.</p> <ul style="list-style-type: none"> ➤ Return to survey.htm, and within the <code>custInfo fieldset</code>, insert the input boxes to record customer information, as shown in Fig. 6-12. Note that the asterisks (*) next to the Name and Email text entries tell the user that these are required fields. (See “Validating a Web Form” on HTML 460 through HTML 470) Save changes to the file. <p style="text-align: center;">* * * *</p> <ul style="list-style-type: none"> ➤ Within the Share Your Experience at Red Ball Pizza fieldset, insert input boxes to record the customer’s experience, as shown in Fig.6-13. Save your changes to the file. ➤ Refresh the survey.htm in a web browser. Test the controls by typing your first and last name in the Name input box Verify that Fig. 6-14 shows the newly inserted control elements in the form with sample text in the <code>custname</code> field 	<p>0.5 HTML 404 through HTML 408</p>	
<p>Adding Field Labels This section starts on HTML408 and extends to HTML 411.</p> <ul style="list-style-type: none"> ➤ Return to survey.htm, add labels for customer information <code>fieldset</code> and enclose the text string <code>Name*</code> within a <code>label</code> element, associating it with the <code>custname</code> input box ➤ Repeat this process for all remaining descriptive text strings, as shown in Fig. 6-15. Save changes to the file. ➤ Refresh the survey.htm in a web browser. Test the labels by clicking each label and verify that the cursor appears within the corresponding control elements. 	<p>0.5 HTML 408 through HTML 411</p>	
<p>Applying a Style Sheet to a Web Form This section starts on HTML411 and extends to HTML 415.</p> <ul style="list-style-type: none"> ➤ Open formstxt.css. Enter your name/the date in comment section. Save the file as forms.css. Below the comment section, add the style rule, as shown in Fig. 6-16. <p style="text-align: center;">* * * *</p> <ul style="list-style-type: none"> ➤ Directly below the style for the <code>fieldset</code> selector, add the style rule, as shown in Fig. 6-17. Save changes to the file. <p style="text-align: center;">* * * *</p> <ul style="list-style-type: none"> ➤ At the bottom of the forms.css style sheet, add the style rule for the labels, as shown in Fig. 6-18. ➤ Below the style rule you just created, add the style rule for input controls, as shown in Fig. 6-19. Save changes to the file. ➤ Return to survey.htm in a text editor, Below the <code>link</code> element that links the file to the rb.css style sheet, add the element to link to the forms.css style sheet. Save changes to file. ➤ Refresh survey.htm in a web browser, and verify Fig. 6-20 shows the revised appearance of the Web form. <p style="text-align: center;">* * * *</p>	<p>1.5 HTML 411 through HTML 415</p>	

<ul style="list-style-type: none"> ➤ Return to the forms.css in a text editor, and add the style rule at the bottom of the file, as shown in Fig. 6-21. Save changes to the file. ➤ Refresh survey.htm in a web browser, and verify that the width of the input box for the <code>state</code> field has been reduce. 		
<p>Defining Default Values and Placeholders This section starts on HTML415 and extends to HTML 418.</p> <ul style="list-style-type: none"> ➤ Return survey.htm in a text editor. Scroll down to the input element for the <code>city</code> field value, and add the attribute to the <code><input></code> tag, as shown on Fig. 6-22 ➤ Add the attribute for <code>state</code> field value, as shown in Fig. 6-22. Save changes to the file. ➤ Refresh survey.htm in web browser, and verify that input boxes for the <code>city</code> and <code>state</code> fields show the text values Ormond Beach and FL, respectively. <p style="text-align: center;">* * * *</p> <ul style="list-style-type: none"> ➤ Return to survey.htm in a text editor and follow the instructions as provided in the first four steps, and add placeholders for the survey form, as shown in Fig. 6-23. Save changes to the file. ➤ Refresh survey.htm in a web browser, and verify that placeholder text has been added to the <code>custname</code>, <code>zip</code>, <code>phone</code>, and <code>receipt</code> input boxes and notice that placeholder text is distinguished from default text by appearing in the grayed-out font, as shown in Fig.6-24. 	<p>1.5 HTML 415 through HTML 418</p>	
<p>Section 6.2 (Selection Lists and Option Buttons)</p> <p>Creating a Selection List This section starts on HTML422 and extends to HTML 429.</p> <ul style="list-style-type: none"> ➤ Return survey.htm in text editor, scroll down to the bottom of the second field list and follow the instructions provided in the first two steps to create the <code>ordertype</code> selection list, as shown on Fig.6-26. Save changes to the file. ➤ Additionally, you want to set the style of the <code>select</code> element, like the input boxes you have developed in previous session. Next, verify that it floats alongside the label, and its font size and margin space are set to match the layout of the survey form. ➤ Go to the forms.css style sheet in text editor. At the bottom of the file, add the code, as shown in Fig. 6-27, to set the style rule for the select element: for selection list: display as a block, float it on the left, set font size to <code>0.9 em</code>, set margin space to <code>7px 0px</code>. Save the changes to the file. ➤ Open survey.htm in web browser, and verify that the survey form displays a selection list for the type of order, as shown in Fig. 6-28 <p style="text-align: center;">* * * *</p> <ul style="list-style-type: none"> ➤ Return to survey.htm in text editor and verify that selected attribute is added to the <code>Dine in</code> option, a shown in Fig. 6-29 is added. Save changes to the file. ➤ Reopen to survey.htm in web browser, and verify that <code>Dine in</code> option is preselected in the order type list <p style="text-align: center;">* * * *</p>	<p>1.5 HTML 422 through HTML 429</p>	










<ul style="list-style-type: none"> ➤ Return to survey.htm in text editor and add the label and selection list directly below the email field, as shown in Fig. 6-31. Save changes to file. ➤ Reopen survey.htm in web browser, and verify that there are five options displayed in the selection list, as shown in Fig. 6-32. * * * * ➤ To allow for multiple selections, return to survey.htm in text editor, and add the text to the label element for the infoSrc selection list, and the attribute for multiple sections to the select element, as shown in Fig. 6-33. Save changes to file. ➤ Reopen survey.htm in web browser, and verify that you can select multiple items from the information source list. 		
<p>Creating Option Buttons This section starts on HTML429 and extends to HTML 434.</p> <ul style="list-style-type: none"> ➤ To create Option buttons, return to survey.htm in text editor, and scroll down to the second fieldset, directly after the ordertype selection list, enter the code for option button group for the serviceFriendly field as shown in Fig. 6-36. Save changes to the file. ➤ Reopen survey.htm in web browser, and verify that the rendered option buttons are in display, as shown in Fig. 6-37. Next, click on one option button, and verify that other options are deselect. Also verify that when labels next to option buttons are clicked, the option buttons become selected. * * * * ➤ To revise the styles for the option button group, go to the forms.css style sheet in a text editor, and insert the style rules for the option button group, at the bottom of the file, as shown in Fig. 6-38. Save the changes to fie. ➤ Reopen survey.htm in web browser, and verify that the revised appearance of the option button group is displayed, as shown in Fig. 6-39. * * * * ➤ To add the remining option button group, return to survey.htm in text editor. Directly below the fieldset element for the serviceFriendly field, add the HTML code in option button groups for the orderCorrect and foodHot fields, as shown in Fig. 6-40. Save changes to the file. ➤ Reopen survey.htm in web browser, and verify all the option button groups in the survey form, is displayed, as shown in Fig. 6-41. 	<p>1.5 HTML 429 through HTML 434</p>	
<p>Creating a Area Box This section starts on HTML434 and extends to HTML 437.</p> <ul style="list-style-type: none"> ➤ To create the comments text area box, return to survey.htm in text editor. Directly below the fieldset element for the foodHot option group, enter the code for a text area control, as shown in Fig. 6-42. Save your changes to the file. ➤ Return to forms.css style sheet in text editor. Enter the style rule for the text area box at the bottom of the style sheet file, as shown in Fig. 6-43. Save the changes to the file. 	<p>0.5 HTML 434 through HTML 437</p>	

<ul style="list-style-type: none"> ➤ Reopen survey.htm in web browser, and verify that newly added text box area appears, as shown in Fig. 6-44. Type some sample text into the text area box, and verify that the text wraps to a new line as you exceed the width of the box. 		
<p>Creating Check Boxes This section starts on HTML437 and extends to HTML 440.</p> <ul style="list-style-type: none"> ➤ To create a check box for inviting customers to subscribe, return to survey.htm in text editor. Insert the code directly above the closing <code></form></code> tag, to create check boxes for the <code>newsdb</code> field, as shown in Fig. 6-45. Save your changes to the file. <p style="text-align: center;">* * * *</p> <ul style="list-style-type: none"> ➤ Go back to the forms.css style sheet in text editor, and create style rule for the label text and check box control, at the bottom of the file, as shown in Fig. 6-46. Save changes to the file. ➤ Refresh survey.htm in web browser, and verify that current state of the web form resembles Fig. 6-47. 	<p>0.5 HTML 437 through HTML 440</p>	
<p>Section 6.3 (HTML5 Data Types)</p> <p>Exploring HTML5 Data Types This section starts on HTML444 and extends to HTML 454.</p> <ul style="list-style-type: none"> ➤ To apply the <code>email</code> and <code>tel</code> (phone) data types, return to survey.htm in text editor. Scroll down to the <code>input</code> element for the <code>phone</code> field and insert the <code>attribute</code>, as shown in step2. Next, insert the <code>attribute</code> for the <code>input</code> element for the <code>email</code> field. Fig. 6-50 displays the newly added code for the <code>tel</code>, and <code>email</code> data types. Save changes to the file. ➤ If you have access to a mobile device with a virtual keyboard that are commonly used as an on-screen input method, access the survey.htm file, and verify that the keyboard layouts changes when the focus is placed on the input boxes for the <code>phone</code>, and <code>email</code> fields. <p style="text-align: center;">* * * *</p> <ul style="list-style-type: none"> ➤ Next return to survey.htm in text editor, and apply the changes to the data type of the <code>visitdate</code> field, as shown in Fig. 6-51. Save your changes to the file. ➤ If you have access to the Opera, Chrome for Windows, or another browser that supports calendar data types, open the survey.htm file in web browser, and click the input box for the date of the customer's visit to Red Ball Pizza. <p>An example of Opera browser that displays a calendar object from which a user can select the date of the visit, is shown in Fig. 6-52. If your browser does not support calendar data type, the input box might not be treated as a text box with no calendar object displayed. Nonetheless, please note that most of the modern browsers should display calendar object from which a user can select the date of the visit.</p>	<p>1.5 HTML 444 through HTML 454</p>	
<ul style="list-style-type: none"> ➤ To use the <code>number</code> data type that will be contained in input boxes using a <code>spinner</code> control, return to survey.htm in text editor, and insert the HTML code directly below the selection list for the <code>infoSrc</code> field to apply a number data type, as seen in Fig. 6-53. Save changes to the file. 	<p>1.5 HTML 448 through HTML 451</p>	

<p>➤ Go to forms.css style sheet in text editor, and create a style rule for the <code>inbox</code>, with 70 pixels wide, to reflect that the style rule for the <code>orderPerMonth</code> field, as shown in Fig. 6-54. Save changes to the file.</p> <p>Refresh survey.htm in web browser, and view the number spin box in the Opera browser of your choice. Click the up and down spin arrows in your browser to verify that they can increase and decrease the value in input box by 1 unit, and that the field value is limited to a range 0 to 10. If Opera is your favorite browser, Fig. 6-55 demonstrates the number spin in Opera browser.</p> <p style="text-align: center;">* * * *</p> <p>➤ To create the two range boxes, return to survey.htm in text editor. Insert the code as displayed in step2, directly above the label for the <code>textarea</code>s element Fig. 6-56 displays adding input elements with the range data type. Save changes to the file.</p> <p>➤ Refresh survey.htm web browser, and verify the appearance of the two sliders in the browser. Fig. 6-57 demonstrates the appearance of two sliders in Opera browser, where the marker on the slider drag back and forth to confirm that the widget works as expected.</p> <p>Notice that the default value of 5 is represented by placing the slider marker in the exact center of the 0 to 10 range. If your browser does not support the range slider, you might only see a text box with default value of 5. Most modern browsers should support range slider, specifying numeric range with the <code>range</code> data type, fairly similar to the Opera browser.</p>		
<p>➤ To add descriptive labels to the range slider, return to survey.htm in text editor. Directly before the <code>input</code> element for the service range slider, insert the code for the <code>label</code> element that portrays the minimum value of the range slider “0” as displayed in step2.</p> <p>➤ Next, insert the code for the <code>label</code> element after the <code>input</code> element for the service range slider that portrays the maximum value of the range slider “10” as displayed in step3.</p> <p>➤ Repeat step 2, and step3 for the <code>input</code> element for the food quality range slider, as shown in step4. Fig 6-58 highlights the revised code. Save changes to the file.</p> <p>➤ Next, you want to create a style rule for these new labels, so that they appear alongside the range sliders, and have to resize the labels and the sliders to make them fit within the width of the field set.</p> <p>➤ Go to forms.css style sheet in text editor. At the bottom of the file, follow the instructions to insert style rules for the range slider, as shown in Fig. 6-59. Note that you have to specifically <i>not</i> clear the label text so that it floats alongside the other objects, within that line on the form. Save changes to the file.</p> <p>Refresh survey.htm in web browser, and verify the appearance of the range sliders in your browser. Fig. 6-60 demonstrate the range slider in Opera browser.</p>	<p>1.5 HTML 452 & HTML 453</p>	
<p>Suggesting Options with Data List This section starts on HTML454 and extends to HTML 456.</p>	<p>1.0 HTML 454 through HTML 456</p>	

<p>➤ To create suggesting options with data list, return to survey.htm in text editor, and directly below the input box for the ordersPerMonth field, enter the code for adding a data list, as shown in Fig. 6-61. Save changes to the file.</p> <p>Reopen survey.htm in your favorite browser. Verify that when the input box for favDish field is clicked, with a letter p entered, the browser displays two menu items that begin with letter P. Fig. 6-62 demonstrate how to access a data list of options.</p>		
<p>Creating Form Buttons This section starts on HTML456 and extends to HTML 460</p> <p>➤ To add the submit and reset buttons to the survey form, return to survey.htm in text editor, scroll down, and directly above the closing <code></form></code> tag, insert the code to create submit and reset buttons, as shown in Fig. 6-63. Save changes to the file.</p> <p>➤ To create style rules for the submit and reset buttons, go to forms.css in text editor, and insert the code to create submit and reset buttons, as shown in Fig. 6-64. Save changes to the file.</p> <p>Refresh survey.htm file in web browser, and verify that the completed web page with all the form elements and controls are present. Fig. 6-65 displays the final layout of the survey page as displayed in Opera browser. Most modern browsers should perform displaying the layout of the survey page, as expected.</p> <p>➤ Enter some example data into the form. Click the Cancel button to test the actions of the reset button, and verify that the customer survey form is reset to its original state and default values. Next, design and create custom button, as shown in Fig. 6-66.</p>	<p>1.0 HTML 456 through HTML 460</p>	
<p style="text-align: center;">Validating a Web Form For Experimentation Only</p> <p>Carefully Read this section starting on HTML460 and extends to HTML 470</p> <p>The final major task after designing and developing a web form is to test and validate the form to ensure that the user of your web form can enter reasonable values in correct format. This process is known as validation. Proper validation of form data is a crucial process. Web form validation is accomplished via the use of form controls, limiting the user to a set of pre-approved values.</p> <p>Validation can take place on a web server, known as server-side validation, or within the user's own browser, referred to as client-side validation. HTML5 provides a collection of attributes for web form validation, that user should be able to control on client-side;</p> <ol style="list-style-type: none"> 1. Indicating Required Values. 2. validating based on Data Type, 3. testing for a Valid Pattern, 4. using CSS3 pseudo-classes to apply Inline Validation. 	<p>Read HTML 460 through HTML 470</p>	
<p>Indicating Required Values This section starts on HTML461 and extends to HTML 463</p> <p>➤ To apply and test the required attribute, return to the survey.htm file in text editor. Add the attribute, as shown on step2, to the input element for the custname field. Scroll down to the input element for the email field, and add</p>	<p>2.5 HTML 461 through HTML 466</p>	

<p>the attribute, as shown on step3. Next, go to the <code>receipt</code> field and add the attribute, as shown in step4.</p> <ul style="list-style-type: none"> ➤ Fig. 6-67 highlights how to mark the required fields for <code>custname</code>, <code>email</code>, and <code>receipt</code>. Save changes to the file. ➤ Reopen survey.htm file in web browser, test these attributes by leaving the form blank, while clicking on Submit My Survey. Note that the browser does not submit the form to the server, but displays an error message for the first invalid field it encounters. As shown in Fig. 6-68. Carefully study this figure for additional error messages that may be encountered. <p style="text-align: center;">* * * *</p> <p>Validating Based on Data Type This section starts on HTML463 and extends to HTML 464</p> <ul style="list-style-type: none"> ➤ To apply <u>validation based on data type</u>, such as the <code>email</code> field, return to the survey.htm file in web browser, and place your first and last name as text in <code>email</code> field, and click Submit My Survey, and verify that the form requests a proper e-mail address, as shown in Fig. 6-69. <p style="text-align: center;">* * * *</p> <p>Testing for a Valid Pattern This section starts on HTML464 and extends to HTML 466</p> <ul style="list-style-type: none"> ➤ To apply and test regular <u>expression patterns</u>, return to the survey.htm file, in text editor, and follow the complete instructions provided in Fig. 6-70 on how to specify character patterns with regular expressions. Save changes to the file. ➤ Refresh survey.htm file in web browser, and enter some text in the Name input box. Type 321 in the input box for the postal code to represent a zip code. Note that the entry of 321 will be rejected by the browser, because it does not match the regular pattern for a postal code value, as shown in Fig.6-71. 		
<p>Applying Inline Validation This section starts on HTML466 and extends to HTML 470</p> <p><u>Inline Validation</u> describes CSS3 pseudo-classes, the focus Pseudo-Class, and Pseudo-Classes for Valid and Invalid data.</p> <ul style="list-style-type: none"> ➤ In order to apply the <u>focus pseudo-class</u>, on control elements that have the focus in the form, return to the forms.css file in text editor, and insert the required style rule, as shown in Fig.6-73. Save changes to the file. ➤ Refresh the survey.htm file in web browser, and click the input box for the <code>street</code> field, and verify that the background color changes, as shown in Fig. 6-74. <p>Next, let's explore the <code>valid</code> and <code>invalid</code> CSS3 pseudo-classes for applying inline validation.</p>	<p style="text-align: center;">1.5 HTML 466 through HTML 470</p>	
<ul style="list-style-type: none"> ➤ In order to apply the <code>valid</code> and <code>invalid</code> pseudo-classes, return to the forms.css file in text editor, and insert the style rules, for the content of the new style rules, as shown in Fig.6-75. Save changes to the file. 		

<ul style="list-style-type: none"> ➤ Refresh the survey.htm file in web browser, and test the inline validation by typing the postal code value 32175-6136, into the zip field. ➤ As shown in Fig. 6-76, the background of the input box provides immediate visual feedback on whether that data value entered is currently valid or invalid. ➤ Next, continue entering different values into different postal code input boxes, and verify immediate feedback on validity of your data. 		
The completed Web form resembles that in Fig 6-65 , and / or figure shown below **. The Web form has been tested against different browsers and complies with all validation clauses as discussed above. Server, as instructed.		
<p>**IMPORTANT** The following ending files for this tutorial project MUST be PALCED in the t6-Red Ball Pizza folder before uploaded to IVC Student Web Server. The Ending Files for this Tutorial Project (see below!)</p> <p>Name</p> <ul style="list-style-type: none">  customer.png  forms.css  formsubmit.js  go.png  modernizr-1.5.js  rb.css  redball.png  stop.png  survey.htm 		
TOTAL POSSIBLE POINTS:	20	0

Instructor Notes:

**




[home](#)
[menu](#)
[directions](#)
[coupons](#)
[orders](#)
[catering](#)
[reviews](#)

Customer Survey

Thank you for taking our customer survey. Your response helps Red Ball Pizza maintain the tradition that has made us the top-rated pizzeria in the metro area.

All participants are automatically entered into a monthly drawing to receive a *Red Ball Express PizzaFest* containing two large pizzas, a 2-liter soda, and a side order of chicken wings. Check your e-mail inbox for contest results.

Surveys are private and confidential. Red Ball Pizza will not share your contact information with third parties, ever.



Required values are marked by an asterisk (*)

Customer Information		Share Your Experience at Red Ball Pizza	
Name *	<input type="text" value="first and last name"/>	Date of visit	<input type="text" value="mm/dd/yyyy"/>
Street address	<input type="text"/>	Receipt number *	<input type="text" value="re-nnnnnn"/>
City	<input type="text" value="Ormond Beach"/>	Order type	<input type="text" value="Dine in"/>
State (abbr.)	<input type="text" value="FL"/>	Was your service friendly?	Yes <input type="radio"/> No <input type="radio"/>
Postal code	<input type="text" value="nnnnn (-nnnn)"/>	Was your order correct?	Yes <input type="radio"/> No <input type="radio"/>
Phone number	<input type="text" value="(nnn) nnn-nnnn"/>	Was your food hot?	Yes <input type="radio"/> No <input type="radio"/>
E-mail *	<input type="text"/>	Rate the overall service (0 = poor; 10 = great)	<input type="text" value="0"/> <input type="range"/> <input type="text" value="10"/>
Where did you hear about us? (select all that apply)	<input type="checkbox"/> Internet <input type="checkbox"/> Magazine <input type="checkbox"/> Newspaper <input type="checkbox"/> Word of Mouth <input type="checkbox"/> Other	Rate the food quality (0 = poor; 10 = great)	<input type="text" value="0"/> <input type="range"/> <input type="text" value="10"/>
How many times do you dine out per month?	<input type="text" value="1"/>	Tell us more about your experience!	<div><div></div></div>
What's your favorite Red Ball dish?	<input type="text"/>		

☐ E-mail me your newsletter for great coupons and specials!

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