|  |
| --- |
|  |

**Self -review Exercises**

**3.1** Fill in the blanks in each of the following:

1. The color input type enables the user to enter a color. At the time of this writing, most browsers render the color input type as a text field in which the user can enter a hexadecimal code.
2. The placeholder attribute allows you to place temporary text in a text field.
3. If you want to bypass validation, you can add the form no validate attribute to input type submit.
4. The required attribute forces the user to enter a value before submitting the form.
5. The spinner control is typically displayed for the number input type and includes only the valid numbers.
6. The time input type enables the user to enter an hour, minute, second and fraction of second.
7. The datalist element provides input options for a text input element.
8. The article element describes content that’s separate from the main content of the page and could potentially be used or distributed elsewhere, such as a news article, forum post or blog entry.
9. The footer element describes the text that usually appears at the bottom of the content or the bottom of a section element.
10. The wbr element indicates the appropriate place to break a word when the text wraps to multiple lines.

**3.2** State whether each of the following is true or false. If false, explain why.

1. Any particular HTML5 form input types must render identically in every HTML5- compliant browser.

**False. The rendering of input types can vary among browsers.**

1. When the focus is placed in the text field (i.e., the cursor is in the text field), the placeholder text is submitted to the server.

**False. When the focus is placed in the text field, the placeholder text disappears. It’s not “submitted” when the user clicks the Submit button (unless the user types the same text).**

1. You do not need to include autofocus in your forms.

**true**

1. The new HTML 5 input types are self validating on the client side, eliminating the need to add complicated scripts to your forms to validate user input and reducing the amount of invalid data submitted.

**true**

1. The range input type is inherently self-validating when it’s rendered by the browser as a slider control, because the user is unable to move the slider outside the bounds of the minimum or maximum value.

**true**

1. HTML5 self validates the tel input type.

**False. The length and format of telephone numbers varies greatly based on location, making validation quite complex, so HTML5 does not self validate the tel input type. To ensure that the user enters a phone number in a proper format, we use the pattern attribute**

1. If the user enters an improperly formatted URL in a url input type, it will not validate. HTML5 does not validate that the URL entered actually exists.

**true**

1. The nav element displays a drop-down menu of hyperlinks.

**False. The nav element groups navigation links.**

1. The header element may be used only one time on a page.

**False. The header element may be used multiple times on a page and often includes HTML headings (<h1> through <h6>)**

1. nav elements can be nested in an aside element.

**true**

1. You might use the brk to prevent awkward word breaks.

**False. You might use the wbr to prevent awkward word breaks.**

**Exercises**

* 1. **Fill in the blanks in each of the following:**

1. The autofocus attribute—used in a single input element on a form—automatically highlights the input element and, if appropriate, places the cursor in the text field after the browser loads and renders the page.
2. The new HTML 5 input types are self-validating on the client side.
3. The number input type enables the user to enter a numerical value.
4. The range input type is inherently self-validating when it’s rendered by the browser as a slider control, because the user is unable to move the slider outside the bounds of the minimum or maximum value.
5. The autocomplete=” on” attribute can be used on input types to automatically fill in the user's information based on previous input.
6. The figcaption element provides a caption for the image in the figure element.
7. The summary element displays a right-pointing arrow next to a summary or caption when the document is rendered in a browser. When clicked, the arrow points downward and reveals the content in the **details element**.
8. The mark element enables you to highlight text.

**3.4** **State whether each of the following is true or false. If false, explain why.**

1. Browsers that render the color input type as a text field require the user to enter a color name.

**False. Most browsers render the color input type as a text field in which the user can enter a hexadecimal code**.

1. When a user enters data into a form then submits the form (typically, by clicking the Submit button), the browser immediately checks that the data is correct.

**True**

1. HTML5 can validate whether an e-mail address entered by the user actually exists.

**True**

1. You can add required to any of the input types.

**True**

1. You can enable autocomplete only for specific input elements.

**False. You can enable autocomplete for an entire form or just for specific elements.**

1. The time element enables you to indentify a date, a time or both.

**False. Time element enables the user to enter an hour, minute, seconds and fraction of second.**

1. The caption element provides a caption for the image in a figure element.

**False. The figcaption element provides a caption for the image in the figure element.**

1. The details element displays a right-pointing arrow next to a summary or caption when the document is rendered in a browser. When clicked, the arrow points downward and reveals the content in the summary element.

**True**

1. The footer element describes content that usually appears at the bottom of the content or section element.

**True**

1. The highlight element enables you to highlight text.

**True**

**3.5** **Write an HTML5 element (or elements) to accomplish each of the following tasks:**

1. **Students were asked to rate the food in the cafeteria on a scale of 1 to 10. Use a meter element with text to its left and right to indicate that the average rating was 7 out of 10.**

<html>

<head>question 3.5 a</head>

<body>

<form method="post" action="http://www.google.com">

<p>

<label>item name:

0 <meter min = "0"

max = "10"

value = "8" /> 10 (average rating was 7 out of 10)

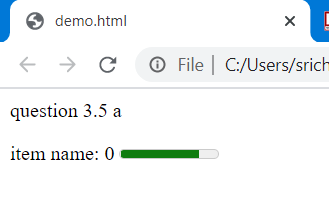
</label>

</p>

</form>

</body>

</html>



1. **Create a details element that displays the summary text "Survey Results" for Part (a). When the user clicks the arrow next to the summary text, an explanatory paragraph about the survey should be displayed**.

<html>

<head>question 3.5 b</head>

<body>

<form method="post" action="http://www.google.com">

<p>

<label>item name:

0 <meter min = "0"

max = "10"

value = "8" ></meter> 10 (average rating was 7 out of 10)

</label>

</p>

<details>

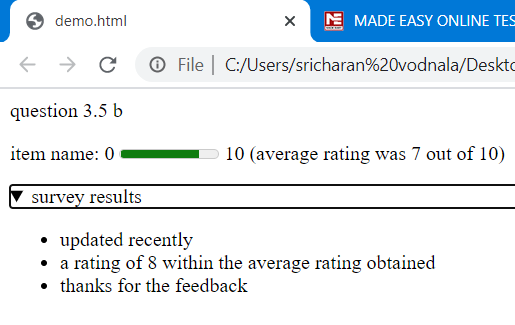
<summary>survey results</summary>

<ul><li> updated recently</li><li> a rating of 8 within the average rating obtained</li><li>thanks for the feedback</li></ul>

</form>

</body>

</html>



1. **Create a text input element for a first name. The element should automatically receive the focus when the form is rendered in a browser.**

<html>

<head>question 3.5 a</head>

<body>

<form method="post" autocomplete="on" >

<p>

<label>full name:

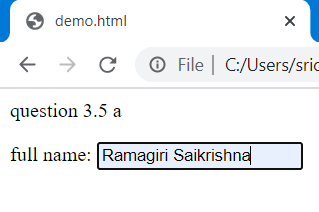
<input type = "text" id="fullname"/>

</label>

</form>

</body>

</html>

****

1. **Modify Part (c) to eliminate the label element and use placeholder text in the input element.**

<html>

<head>question 3.5 d</head>

<body>

<form method="post" autocomplete="off" >

<p>

<label>full name:

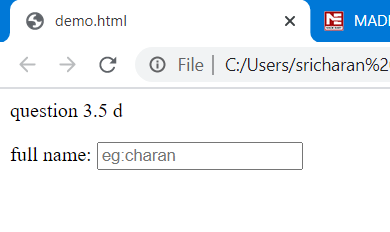
<input type = "text" id="full name" placeholder="eg:annebella"/>

</label>

</form>

</body>

</html>



1. **Use a datalist to provide an autocomplete list for five states.**

<html>

<head>question 3.5 a</head>

<body>

<form method="post" autocomplete="on" >

<p><label>first name:

<input type = "text" id="firstname"/>

</label></p>

<p><label>last name:

<input type = "text" id="lastname"/>

</label></p>

<p><label>Email:

<input type = "email" id = "email"

placeholder = "name@domain.com" /> (name@domain.com)

</label></p>

<p><label for ="txtlist">favourite day:

<input type = "text" id = "txtList"

placeholder = "Select an item" list = "day" />

<datalist id = "day">

<option value="sunday">

<option value="thursday">

<option value="tuesday">

<option value="friday">

<option value="saturday">

</datalist>

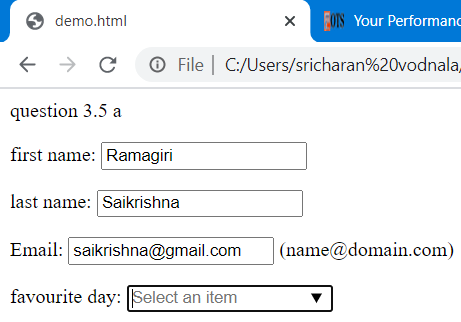
</label>

</p>

</form>

</body>

</html>



1. **Create a range input element that allows the user to select a number from 1 to 100.**

<html>

<head>question 3.5 f</head>

<body>

<form method="post" autocomplete="on" >

<label>item name:

1 <input type = "range"

min = "1"

max = "100"

value = "60" /> 100

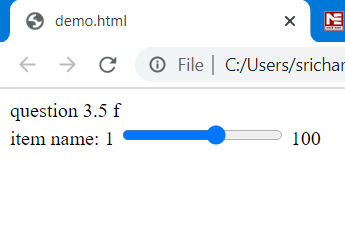
</label>

</p>

</form>

</body>

</html>



**g) Specify that autocomplete should not be allowed for a form. Show only the form’s**

**opening tag.**

<html>

<head>question 3.5 g</head>

<body>

<form method="post" autocomplete="off" >

.

.

.

</html>

**h) Use a mark element to highlight the second sentence in the following paragraph.**

**<p>Students were asked to rate the food in the cafeteria**

**on a scale of 1 to 10. The average result was 7.</p>**

<html>

<head>question 3.5 h</head>

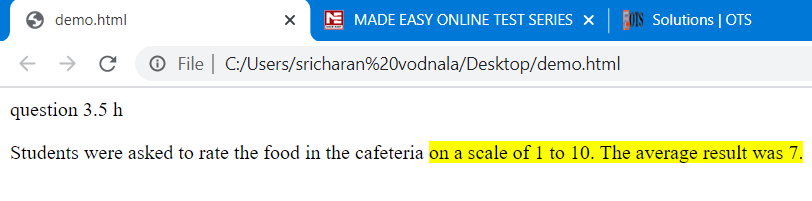
<body>

<p>Students were asked to rate the food in the cafeteria

<mark>on a scale of 1 to 10. The average result was 7.</mark></p>

</body>

</html>



**3.6** **(Website Registration Form with Optional Survey) Create a website registration form to obtain a user’s first name, last name and e-mail address. In addition, include an optional survey question that asks the user’s year in college (e.g., Freshman). Place the optional survey question in a details element that the user can expand to see the question.**

<html>

<head>question 3.6</head>

<body>

<form method="post" autocomplete="on" active="www.google.com">

<p><label>first name:

<input type = "text" id="firstname"/>

</label></p>

<p><label>last name:

<input type = "text" id="lastname"/>

</label></p>

<p><label>Email:

<input type = "email" id = "email"

placeholder = "name@domain.com" /> (name@domain.com)

</label></p>

<p><label for ="txtlist">year of study:

<input type = "text" id = "txtList"

placeholder = "Select an item" list = "year of study" />

<datalist id = "year of study">

<option value="fresher">

<option value="2nd year">

<option value="3rd year">

<option value="final year">

<option value="passed out">

</datalist><details>

<summary>any queries?</summary>

<ul><li>to confirm the student details</li><li>thanks for the feedback</li></ul>

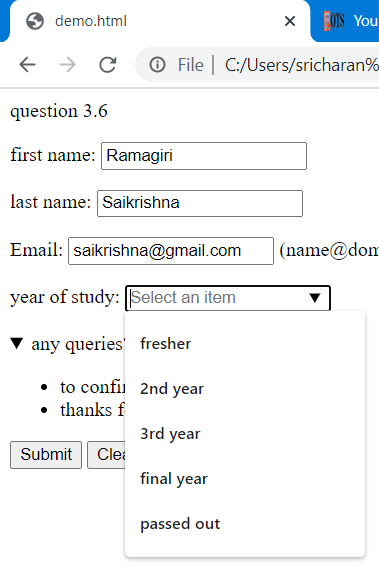
<p><input type = "submit" value = "Submit" />

<input type = "reset" value = "Clear" /></p>

</form>

</body>

</html>



**3.7 (Creating an Autocomplete Form) Create a simple search form using a search input element in which the user can enter a search query. Using the Firefox web browser, test the form by entering January and submitting the form. Then enter a J in the input element to see previous entries that started with J—January should be displayed below the input element. Enter June and submit the form again. Now enter a J in the input element to see previous entries that started with J— January and June should be displayed below the input element. Try this with your own search queries as well.**

<html>

<head>question 3.7</head>

<body>

<form method="post" autocomplete="on" active="www.google.com">

<p><label for="txtlist">search:

<input type = "search" id="txtlist" placeholder = "search query" list="months"/>

<datalist id="months">

<option value = "January">

<option value = "February">

<option value = "March">

<option value = "April">

<option value = "May">

<option value = "June">

<option value = "July">

<option value = "August">

<option value = "September">

<option value = "October">

<option value = "November">

<option value = "December">

</datalist>

</p>

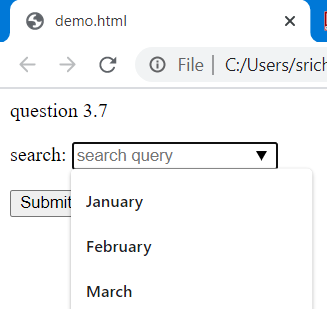
<p><input type = "submit" value = "Submit" />

<input type = "reset" value = "Clear" /></p>

</form>

</body>

</html>



**3.8 (Creating an Autocomplete Form with a datalist) Create an autocomplete input element with an associated datalist that contains the days of the week.**

<html>

<head>question 3.8</head>

<body>

<form method="post" autocomplete="on" active="www.google.com">

<p><label for ="txtlist">favourite day:

<input type = "text" id = "txtList"

placeholder = "Select an item" list = "day" />

<datalist id = "day">

<option value="sunday">

<option value="monday">

<option value="tuesday">

<option value="wednesday">

<option value="thursday">

<option value="friday">

<option value="saturday">

</datalist>

</label>

</p>

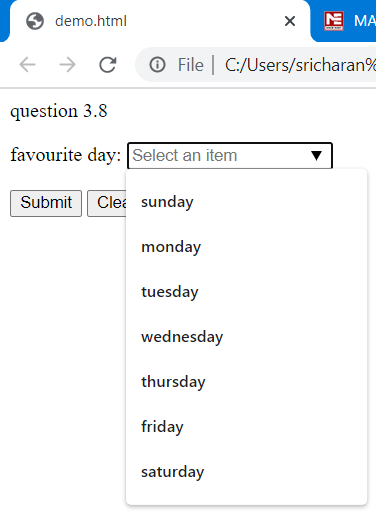
<p><input type = "submit" value = "Submit" />

<input type = "reset" value = "Clear" /></p>

</form>

</body>

</html>



**3.9 (Laying Out Book Pages in HTML5: Creating the Sections) Mark up the paragraph text from Section 3.2.1 of this chapter as a web page using page-structure elements. The text is provided in the exerciseTextAndImages folder with this chapter’s examples. Do not include the figures in this exercise.**

<html>

<head>question 3.9</head>

<body>

<section id = "1">

<nav>

<h2> Recent Publications</h2>

<ul>

<li><a href = "http://www.deitel.com/books/iw3htp5">

Internet & World Wide Web How to Program, 5/e</a></li>

<li><a href = "http://www.deitel.com/books/androidfp/">

Android for Programmers: An App-Driven Approach</a>

</li>

<li><a href = "http://www.deitel.com/books/iphonefp">

iPhone for Programmers: An App-Driven Approach</a></li>

<li><a href = "http://www.deitel.com/books/jhtp9/">

Java How to Program, 9/e</a></li>

<li><a href = "http://www.deitel.com/books/cpphtp8/">

C++ How to Program, 8/e</a></li>

<li>

<a href = "http://www.deitel.com/books/vcsharp2010htp">

Visual C# 2010 How to Program, 4/e</a></li>

<li><a href = "http://www.deitel.com/books/vb2010htp">

Visual Basic 2010 How to Program</a></li>

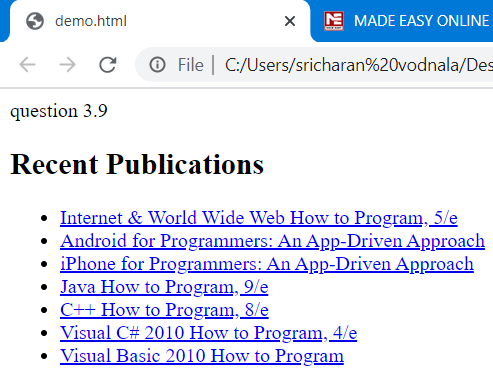
</ul>

</nav>

</section>

</body>

</html>



**3.10 (Laying Out Book Pages in HTML5: Adding Figures) Modify your solution to Exercise 3.9 to add the section’s graphics as figures. The images are provided in the exerciseTextAndImages folder with this chapter’s examples.**

<html>

<head>question 3.10</head>

<body>

<section id = "2">

<h2>wallpapers</h2>

<h3><em>most recent themes</em></h3>

<figure>

<img src = "F:\mypic.jpg" height="65" width="50" alt = "most recent themes" />

<figcaption><em>wallpapers</em>

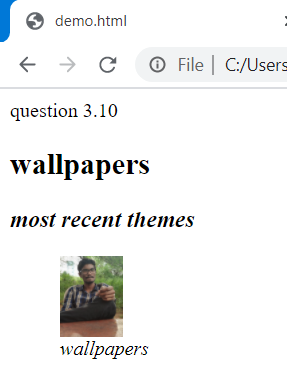
</figcaption>

</figure>

</section>

</body>

</html>



**3.11 (Laying Out Book Pages in HTML5: Adding a details Element) Modify your solution to Exercise 3.10 to add the table in Fig. 3.5. Use the figure caption as the summary and format the table as an HTML table element inside the details element.**

<html>

<head>question 3.11</head>

<body>

<section id = "1">

<details>

<summary>table of contents:</summary>

<figure>

<img src = "F:\mypic.jpg" height="65" width="50" alt = "most recent themes" />

<figcaption><em>wallpapers</em>

</figcaption>

</figure>

<table border="1">

<thead>

<tr>

<th>input type</th>

<th>format</th>

</tr>

</thead>

<tbody>

<tr>

<td>color</td>

<td>hexadecimal code</td>

</tr>

<tr>

<td>date</td>

<td>yyyy-mm-dd</td>

</tr>

<tr>

<td>datetime</td>

<td>yyyy-mm-dd</td>

</tr>

<tr>

<td>datetime-local</td>

<td>yyyy-mm-ddThh:mm</td>

</tr>

<tr>

<td>month</td>

<td>yyyy-mm</td>

</tr>

<tr>

<td>number</td>

<td>any numerical value</td>

</tr>

<tr>

<td>email</td>

<td>name@domain.com</td>

</tr>

<tr>

<td>url</td>

<td>http://www.domainname.com</td>

</tr>

<tr>

<td>time</td>

<td>hh.mm</td>

</tr>

<tr>

<td>week</td>

<td>yyyy-wnn</td>

</tr>

</tbody>

</details>

</section>

</body>

</html>

