

Internet Programming I

Chapter 2 Web Development Using HTML



Chere L. (M. Tech) and Biruk G. (MSc.)
Lecturer, Dept. of Software Eng.

Nov 2021, AASTU

Objectives



After success completion of the chapter you will be able to:

- Identify the core web development technologies
- Understand the HTML document structure and contents models.
- Identify and use HTML elements and attributes
- Build website using HTML

Lesson 5

HTML Elements

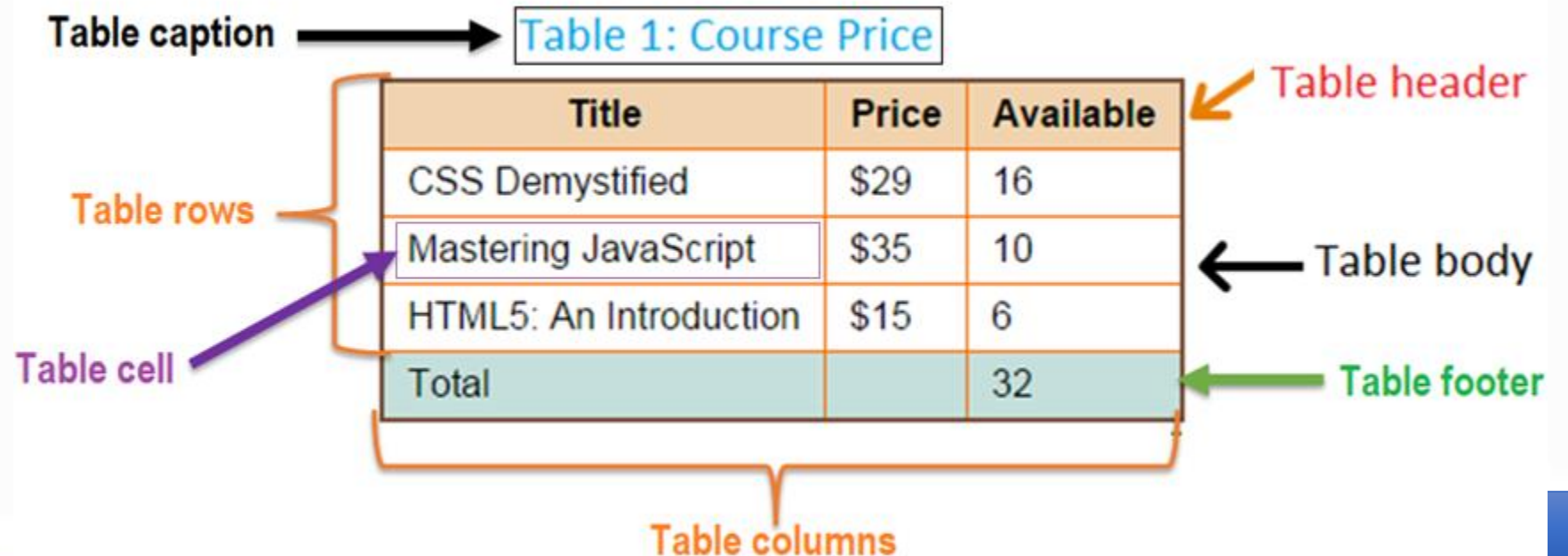
(Designing a Web Table and Form)

- Table Elements
 - Table structure
 - Basic table elements
 - Other table elements
- Working with Forms
 - Form elements
 - Form controls
 - Form validation
- Summary

Working with HTML Table

1. Creating Table in HTML

- Tables play an essential role in organizing complex data in a simple and logical way, and presenting it in easy follow.
- The HTML table model allows authors to **arrange data (text, preformatted text, images, links, forms, form fields, other tables, etc.)** in tabular.
- Before taking a look how to use tables in HTML, let's take a look at *Table structure*.



HTML Table Elements

- Basic HTML table elements - required tags

Element	tag	Description
Table	<code><table> </table></code>	<ul style="list-style-type: none">➤ Defines the beginning and end of the table➤ A container tag that holds the other components of the html table.
Table rows	<code><tr> </tr></code>	<ul style="list-style-type: none">➤ Defines the beginning and end of a horizontal row.➤ May be grouped into a head, foot, and body sections
Table cells	<code><td> </td></code>	<ul style="list-style-type: none">➤ Defines an individual cell.➤ Cells are always placed inside a row.

Example 1: Creating a table with default style

```
<table>
<tr>
  <td> Title </td>
  <td> Price </td>
  <td> Available </td>
</tr>
<tr>
  <td> CSS Demystified </td>
  <td> $29 </td>
  <td> 16 </td>
</tr>
<tr>
  <td> Mastering JavaScript </td>
  <td> $35 </td>
  <td> 10 </td>
</tr>
<tr>
  <td> HTML5: An Introduction </td>
  <td> $15 </td>
  <td> 6 </td>
</tr>
</table>
```

Title	Price	Available
CSS Demystified	\$29	16
Mastering JavaScript	\$35	10
HTML5: An Introduction	\$15	6

Note:

➤ *By default table doesn't have borders*



Element	tag	Description
Table caption	<code><caption> </caption></code>	<ul style="list-style-type: none"> ➤ Specifies the caption (or title) of a table that provides a short description of the table's purpose. ➤ Goes after the table tag but not inside “tr” or “td.” ➤ Only single caption per table. ➤ Appears centered above the table.
Table header	<code><th> </th></code>	<ul style="list-style-type: none"> ➤ Defines a table header row. Used instead of “td.” ➤ Text is bold & centered within the cell
Table section	<code><thead> </thead></code>	Defines a set of rows defining the head of the columns of the table
	<code><tbody> </tbody></code>	Encapsulates a set of table rows (<tr> elements), indicating that they comprise the body of the table
	<code><tfoot> </tfoot></code>	Defines a set of rows summarizing the columns of the table

Example 2: Table with caption, header and footer

```
<table>
  <caption> Course Price</caption>
  <thead>
    <tr><th> Title </th> <th> Price </th> <th> Available </th> </tr>
  </thead>

  <tbody>
    <tr><td> CSS Demystified </td> <td> $29 </td> <td> 16 </td> </tr>
    <tr><td> Mastering JavaScript</td><td> $35 </td> <td> 10 </td> </tr>
    <tr><td> HTML5: An Introduction </td><td> $15 </td><td> 6 </td></tr>
  </tbody>

  <tfoot>
    <tr><td> Total </td>    <td></td>    <td> 32 </td></tr>
  </tfoot>
</table>
```

Course Price

Title	Price	Available
CSS Demystified	\$29	16
Mastering JavaScript	\$35	10
HTML5: An Introduction	\$15	6
Total		32

Table elements attribute

- All most all attributes of the table elements are deprecated.
- Only *<td>* and *<tr>* tags have some attributes described below

<td> tag and <tr> tag attributes

Attribute	Value	Description
colspan	<i>Positive integer ≤1000</i>	<ul style="list-style-type: none">➤ Indicates for how many columns the cell extends (span)➤ Its default value is 1
rowspan	<i>Positive integer ≤65534</i>	<ul style="list-style-type: none">➤ Indicates for how many rows the cell extends (span).➤ Its default value is 1; if its value is set to 0, it extends until the end of the table section
headers	<i>Header_id</i>	Specifies one or more header cells a table cell is related to
scope	<i>row / col / rowgroup / colgroup</i>	<i>Enumerated attribute defines the cells that the header (defined in the <th>) element relates to</i>

Example 3: Merging Cells with rowspan and colspan attributes

```
<table>
  <thead>
    <tr>
      <th rowspan = "2">Browser</th>
      <th colspan = "2">Vistor</th>
    </tr>
    <tr>
      <th>Number</th>
      <th>Precentage</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Mozila Firefox</td><td>163</td><td>59%</td> </tr>
      <tr><td>Google Chrome</td><td>78</td> <td>28%</td> </tr>
      <tr><td>Safari</td><td>35</td><td>13%</td> </tr>
    </tbody>
  </table>
```

Browser	Visitors	
	Number	Percentage
Mozilla Firefox	163	59%
Google Chrome	78	28%
Safari	35	13%



The <colgroup> element

- It defines a **group of columns** within a table for formatting .
- It is useful for applying styles to entire columns, instead of repeating the styles for each cell, for each row.
- To define different properties to a column within a <colgroup> element, it require the **<col> tag**

The <col> element

- It specifies column properties (common semantics) for each column (all common cells) within a **<colgroup> element**.
- The common semantic (like color, padding etc.) is specified using CSS.

Attributes :

- Both elements (<colgroup> and <col>) has single attribute named “span” which a positive integer indicating the number of consecutive columns the elements span.

Example :

	Section A		Section B	
Class	Male	Female	Male	Female
8	30	40	30	20
9	20	30	30	20
10	10	20	20	30
Total	60	90	80	70

Example – using CSS:

```
<h3>The colgroup element</h3>
<table>
  <colgroup>
    <col span="2" style="background-color:red">
    <col style="background-color:yellow">
  </colgroup>
  <tr><th>ISBN</th><th>Title</th><th>Price</th></tr>
  <tr>
    <td>3476896</td><td>My first HTML</td><td>$53</td>
  </tr>
  <tr><td>5869207</td><td>My first CSS</td><td>$49</td>
  </tr>
</table>
```

The colgroup element

ISBN	Title	Price
3476896	My first HTML	\$53
5869207	My first CSS	\$49

Summary of Table Elements



Current HTML5 Table Elements

Tag	Use
<code><table>...</table></code>	Indicates a table.
<code><caption>...</caption></code>	Creates a caption for the table (optional).
<code><colgroup>...</colgroup></code>	Encloses one or more columns in a group.
<code><col></code>	Used to define the attributes of a column in a table.
<code><thead>...</thead></code>	Creates a row group that defines the heading of the table. A table can contain only one heading.
<code><tfoot>...</tfoot></code>	Creates a row group that defines the footer of the table. A table can contain only one footer. Must be specified before the body of the table is rendered.
<code><tbody>...</tbody></code>	Defines one or more row groups to include in the body of the table. Tables can contain more than one body section.
<code><tr>...</tr></code>	Defines a table row, which can contain heading and data cells.
<code><th>...</th></code>	Defines a table cell that contains a heading. Heading cells are usually indicated by boldface and centered both horizontally and vertically within the cell.
<code><td>...</td></code>	Defines a table cell containing data. Table cells are in a regular font and are left-aligned and vertically centered within the cell.

Working with Web Forms

2. Working with Web Form

- **Web form**

- Allows users to enter data that can be saved and processed.
- Common way to accept user input (collect some data from the site visitor)

E.g. user registration – capture user info like name, address, credit card, etc.

- It will post the data captured from user to a back-end application such as CGI, ASP Script or PHP script etc. which will perform required processing on the passed data based on defined business logic inside the application.
- Allows the creation of interactive websites for user feedback

- **The <form> HTML element**

- *Represents a web document section containing interactive controls for collecting and submitting information.*

Interaction between the web form and the server



Data from the web form (name of each control element along with its value) sent to the programming running on the server



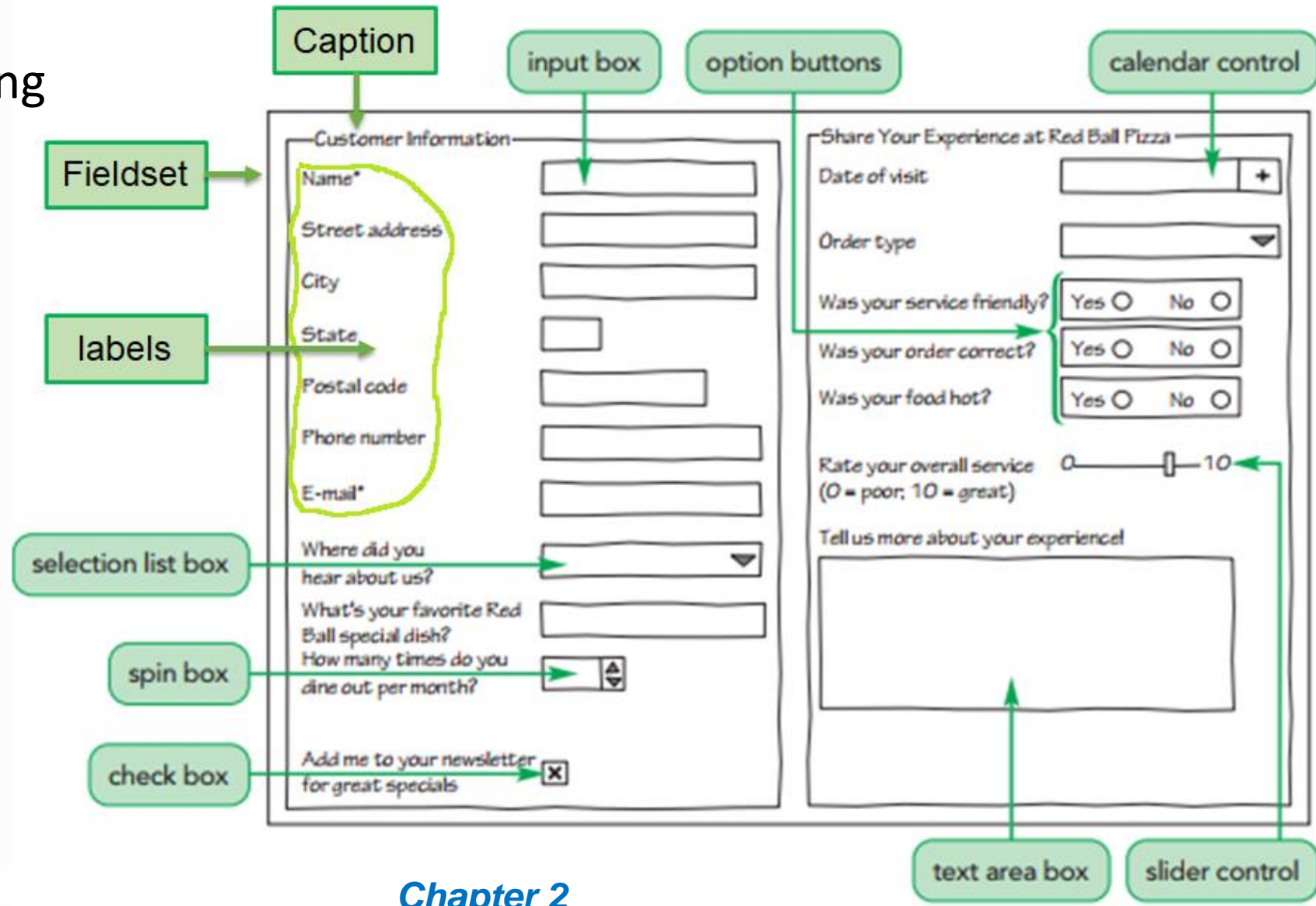
The server processes the submitted information using a server side program such as PHP, C#, or Java.

The server may also store the information in a database.

Feedback or processed data can be sent back to the browser

- The web form can contain one or more of the following

- `<input>` element
- `<textarea>` element
- `<button>` element
- `<select>` element
- `<option>` element
- `<optgroup>` element
- `<fieldset>` element
- `<legend>` element
- `<label>` element
- `<output>` element
- `<datalist>` element
- `<progress>` element
- `<meter>` element



The diagram illustrates a web form with two main sections: 'Customer Information' and 'Share Your Experience at Red Ball Pizza'. Various input controls are labeled with green boxes and arrows:

- Fieldset:** Points to the 'Customer Information' section.
- labels:** Points to the labels for 'Name*', 'Street address', 'City', 'State', 'Postal code', 'Phone number', and 'E-mail*'.
- input box:** Points to the text input fields for 'Name*', 'Street address', 'City', 'Postal code', 'Phone number', and 'E-mail*'. It also points to the 'Date of visit' input box in the second section.
- option buttons:** Points to the radio buttons for 'Was your service friendly?', 'Was your order correct?', and 'Was your food hot?'.
- calendar control:** Points to the date picker icon next to the 'Date of visit' input box.
- selection list box:** Points to the dropdown menu for 'Where did you hear about us?'.
- spin box:** Points to the spinner control for 'How many times do you dine out per month?'.
- check box:** Points to the checkbox for 'Add me to your newsletter for great specials'.
- text area box:** Points to the large text area for 'Tell us more about your experience!'.
- slider control:** Points to the range input for 'Rate your overall service (0 = poor; 10 = great)'.

From control elements

- These are the objects that allow a user to interact with a form.
- Each *data entry control element* is associated with a **data field** that stores the data values supplied by a user.
- Types of **controls**
 - **Text Input boxes**
 - ✓ *Single line input*
 - ✓ *Multi-line input*
 - ✓ *Password input*
 - **Choice/Selection**
 - ✓ *Selection lists*
 - ✓ *Radio buttons*
 - ✓ *Check boxes*
 - **Widget elements**
 - ✓ *Spin boxes*
 - ✓ *Slider controls*
 - ✓ *Calendar controls*
 - ✓ *Color pickers*

(a) Creating Web forms

- Web forms are marked using the *form element*

<form id="text" attributes>

form control elements goes here

</form>

- **id attribute** used to uniquely identifies the form
 - **attributes** specify additional attributes of the form
 - Some of the form attributes control the behavior during form submission.
- A form element can be **placed anywhere** within the body of a page
 - Forms also can contain other web page elements such as *tables, paragraphs, inline images, and headings*

Example:

```
<html>
<body>

<h1>The fieldset element</h1>

<form id = "registration" action=" " method="get">
  <fieldset>
    <legend>Personalia:</legend>
    <label for="fname">First name:</label>
    <input type="text" id="fname" name="fname"><br><br>
    <label for="lname">Last name:</label>
    <input type="text" id="lname" name="lname"><br><br>
    <label for="email">Email:</label>
    <input type="email" id="email" name="email"><br><br>
    <label for="birthday">Birthday:</label>
    <input type="date" id="birthday" name="birthday"><br><br>
    <input type="submit" value="Submit">
  </fieldset>
</form>

</body>
</html>
```

The Form element

Personalia: _____

First name:

Last name:

Email:

Birthday: 



Attribute	Value	Description
action	<i>URL</i>	Specifies where to send the form-data when a form is submitted (provide the location of web server program)
method	get / post	Specifies the HTTP method to use when sending form-data
novalidate	Novalidate	Specifies that the form should not be validated when submitted
accept-charset	<i>character_set</i>	Specifies the character encodings that are to be used for the form submission
autocomplete	<i>On / off</i>	<i>Specifies whether a form should have autocomplete or not</i>
name	String	<i>The name of the form. The value must be unique among the form elements and also must not be the empty</i>
target	<i>_blank / _self / _parent / _top</i>	<i>Specifies where to display the response that is received after submitting the form</i>

GET	POST
Only limited amount of data can be sent because data is sent in header.	Large amount of data can be sent because data is sent in body.
Get request is not secured because query string appended in the URL bar.	Post request is secured because data is not exposed in the URL bar.
Get request can be bookmarked	Post request cannot be bookmarked.
A Get request is often cacheable.	A Post request can hardly cacheable.
Get request is more efficient and used more than post.	Post request is less efficient and used less than Get.

(b) Grouping From Control Elements

▪ Field set:

- Groups fields (control elements) that share a common purpose
- Field sets are created using the **fieldset element**

▪ Legend:

- Describes the content of a field set using the **legend element**
- Contains only text and no nested elements
- By default, it placed in the top-left corner of the field set box and can be moved to a different location using the CSS positioning styles



(c) Form Input Element

- The `<input type=" " >` is an important element of HTML form.
- It is the most commonly used element to create interactive controls for web-based forms in order to accept data from the user.
- A wide variety of the *types of input data and control widgets* are available, depending on the device and user agent.
- **Syntax:**

`<input name="name" id="id" type="type" />`

- **Name attribute** – provides the name of the **data field** associated with the control
- **id attribute** – uniquely identifies the control in which the user enters the value
- **type attribute** - indicates the input type (data type) of the field

HTML Input Types

- The "type" attribute of input element can be various types, which defines data field.
- List the types of <input> element given in the table.

type=" "	Description
text	Defines a one-line text input field
password	Defines a one-line password input field
radio	Defines a radio button which allows select one option
checkbox	Defines checkboxes which allow select multiple options form.
file	Defines to select the file from device storage.
submit	Defines a submit button to submit the form to server.
reset	Defines a reset button to reset all values in the form.
button	Defines a simple push button, which programmed to perform a task on an event
image	Defines a graphical submit button.



type=" "	Description
color	Defines an input field with a specific color.
date	Defines an input field for selection of date.
datetime-local	Defines an input field for entering a date without time zone.
email	Defines an input field for entering an email address.
month	Defines a control with month and year, without time zone.
number	Defines an input field to enter a number.
url	Defines a field for entering URL
week	Defines a field to enter the date with week-year, without time zone.
search	Defines a single line text field for entering a search string.
tel	Defines an input field for entering the telephone number.

Customer Survey

Required values are marked by an asterisk (*)

Field set legend

default value for the "custCity" filed

Customer Information

Name*

Street address

City

State

Postal code

Where did you hear about us?

☐ Internet
☐ Magazine
☐ Newspaper
☐ Word of Mouth
☐ Other

☐ Add me to your mailing list.

placeholder text for the "phone" input box

Selection list box control showing five items; the user can select more than one option.

Share Your Experience at Red Ball Pizza

Date of visit

Order type

Was your service friendly?

Yes ☐ No ☐

Tell us more about your experience!

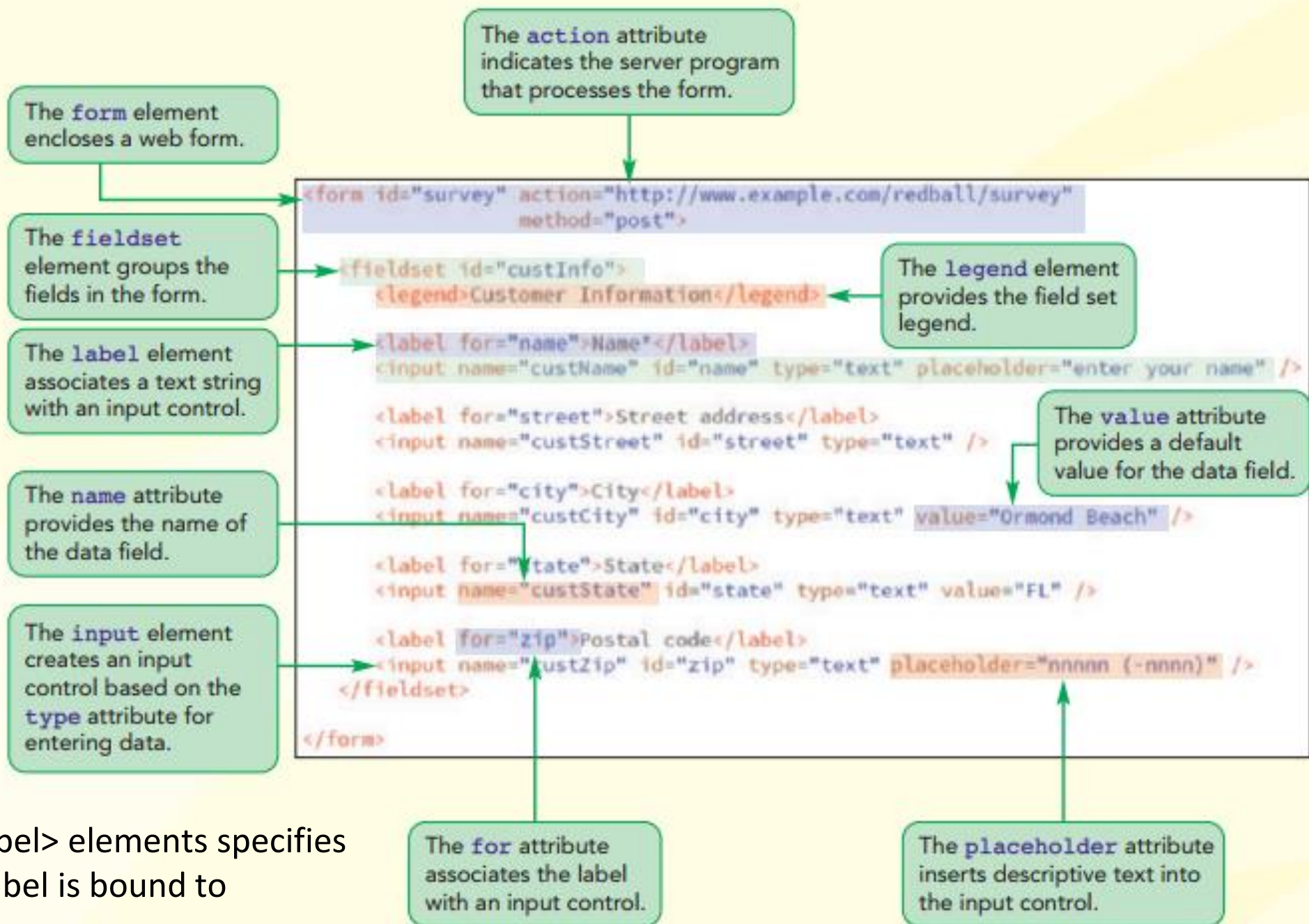
The user can select only one option button control.

The user can type in the text area box control.

Selection list displayed as a drop-down list box control with the default option displayed.

Cont'd

Examine the form design code



The `for` attribute

- When used with the `<label>` elements specifies which form element a label is bound to

Cont'd

Examine the form design code

The `select` element creates a drop-down list box control.

Each option in a selection list is marked with the `option` element.

The `selected` attribute identifies the default option in a selection list.

Each radio button within an option group belongs to the same data field.

The `textarea` element marks a text area box control.

The `size` attribute sets the number of visible options.

The `multiple` attribute allows for multiple selections from the drop-down list.

The `checkbox` data type creates a checkbox control.

The `radio` data type creates an option button control.

```
<label for="infoSrc">Where did you hear about us?</label>
<select name="infoSrc" id="info" size="5" multiple>
  <option value="internet">Internet</option>
  <option value="mag">Magazine</option>
  <option value="news">Newspaper</option>
  <option value="word">Word of Mouth</option>
  <option value="other">Other</option>
</select>

<input name="mailMe" id="mailCB" value="yes" type="checkbox" />
<label for="mailCB">Add me to your mailing list.</label>

<label for="orderType">Order type</label>
<select name="orderType" id="order">
  <option value="order1">Carry out</option>
  <option value="order2">Delivery</option>
  <option value="order3" selected>Dine in</option>
  <option value="order4">Take 'n bake</option>
</select>

<label>Was your service friendly?</label>
<fieldset class="optGroup">
  <label for="fYes">Yes</label>
  <input name="sFriend" id="fYes" value="yes" type="radio" />
  <label for="fNo">No</label>
  <input name="sFriend" id="fNo" value="no" type="radio" />
</fieldset>

<label for="commBox">Tell us more about your experience!</label>
<textarea name="custExp" id="commBox"></textarea>
```




Attribute	Description
<u>autocomplete</u>	Hint for form autofill feature
<u>autofocus</u>	Automatically focus the form control when the page is loaded
<u>disabled</u>	Define whether the form control is disabled
<u>form</u>	Associates the control with a form element
<u>name</u>	Name of the form control. Submitted with the form as part of a name/value pair.
<u>value</u>	The initial value of the control.
<u>readonly</u>	Boolean. The value is not editable
<u>required</u>	Boolean. A value is required or must be check for the form to be submittable

Attributes that applied to “image” and “submit” to <input type = “ “>

Attribute	Description
formaction	URL to use for form submission
formenctype	Form data set encoding type to use for form submission
formmethod	HTTP method to use for form submission
formnovalidate	Bypass form control validation for form submission
formtarget	Browsing context for form submission
<u>height</u> , width (img only)	Same as <i>height</i> , <i>width</i> attributes for ;
src, alt (img only)	Same as <i>src</i> , <i>alt</i> attributes for ;

Attribute		Description
numeric types	<u>max, min</u>	Maximum and Minimum value respectively
numeric types	<u>step</u>	Incremental values that are valid.
password, search, tel, text, url	<u>maxlength, minlength</u>	Maximum and Minimum length (number of characters) of value respectively
password, search, tel, text, url	<u>placeholder</u>	Text that appears in the form control when it has no value set
email, password, tel, text, url	<u>size</u>	Size of the control
password, text, tel	<u>pattern</u>	Pattern the value must match to be valid
email, file	<u>multiple</u>	Boolean. Whether to allow multiple values
file	<u>accept</u>	Hint for expected file type in file upload controls

(d) Designing a Form Layout

- There are two general layouts
 - *Labels are placed directly above the input controls*
 - *Labels and controls are placed side-by-side*

Customer Information

Name *

Street address

City

State

one-column layout

Customer Information

Name *

Street address

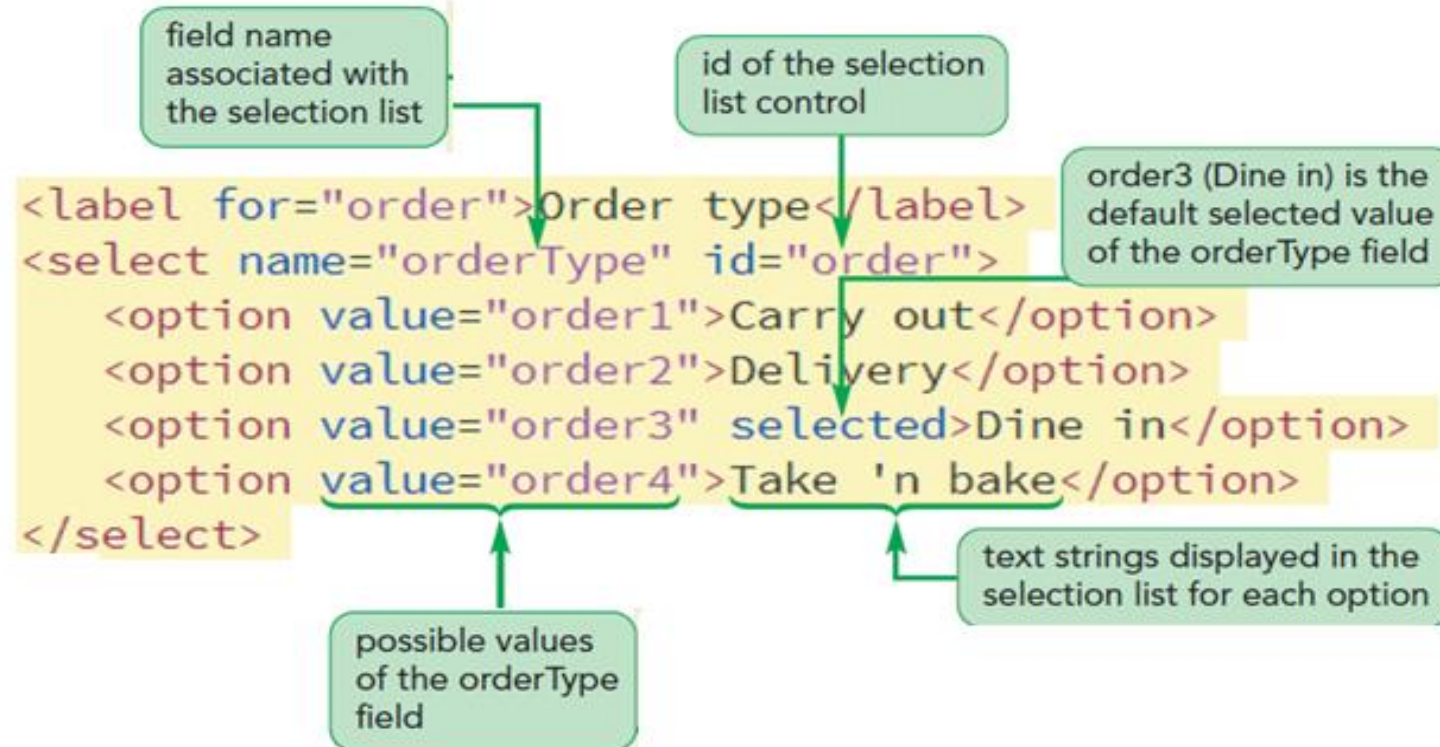
City

State (abbr.)

two-column layout

(e) Creating a Selection List

- A **selection list** is a list box that presents users with a group of possible values for the data field
- The list is created using the **select and option elements**



Share Your Experience at Red Ball Pizza

Date of visit:

Order type:

- Dine in
- Carry out
- Delivery
- Dine in**
- Take 'n bake

calendar control for selecting the date of the customer's visit

Dine in is the default selected option

a drop-down list box displays the text of the four options

- *By default, a selection list appears as a drop-down list box*
- *To display a selection list as a scroll box, use the size attribute*

Multiple items selection

- Two ways for users to select multiple items from a selection list
 - For non-contiguous selection, press and hold the Ctrl key while making the selections
 - For contiguous selection, select the first item, press and hold the Shift key, and then select the last item in the range

(f) Grouping Selection Options

- The selection list options can be organized by placing them in option groups using the **optgroup** element

```
<label for="appetizers">Starter Menu</label>
<select name="meal">
  <optgroup label="Appetizers">
    <option value="sms">Spicy Mozzarella Sticks</option>
    <option value="pr">Pepperoni Rolls</option>
    <option value="tr">Toasted Ravioli</option>
  </optgroup>
  <optgroup label="Salads">
    <option value="sms">Pasta Salad</option>
    <option value="tbs">Tuscan Bread Salad</option>
    <option value="pr">Caesar Salad</option>
  </optgroup>
</select>
```



(g) Data Lists - suggesting options

- **Data list** is a list of possible data values that a form field can have.
- It defined using the **datalist element**

```
<label for="dish">What's your favorite dish?</label>
<input name="favDish" id="dish" type="text" list="dishType" />
<datalist id="dishType">
  <option value="Anitpasto Pizza" />
  <option value="Big Kahuna Pizza" />
  <option value="BBQ Chicken Pizza" />
  <option value="Mediterranean Herb Pizza" />
  <option value="Pasta Rolls" />
  <option value="Pasto Artichoke Pizza" />
</datalist>
```

data list containing
suggested values

links the favDish
field to the dishType
data list



Where did you hear about us?
(select all that apply)

What's your favorite dish?

How many times do you dine out?

☐ Add me to your mailing list

Internet
Magazine
Newspaper
Word of Mouth
Other

p

Pasta Rolls
Pasto Artichoke Pizza

suggested values from
the data list starting
with the letter "p"

(h) Entering Date and Time Values

- Date and time fields ensure that users enter data in the correct format
- Indicated using type attributes: **date**, **time**, **datetime-local**, **month**, and **week**

August 2021

August 2021

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

type="month"

Week 33, 2021

August 2021

Week	Sun	Mon	Tue	Wed	Thu	Fri	Sat
31	1	2	3	4	5	6	7
32	8	9	10	11	12	13	14
33	15	16	17	18	19	20	21
34	22	23	24	25	26	27	28
35	29	30	31	1	2	3	4

type="week"

08/17/2021

August 2021

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

type="date"

08/17/2021 04:21 PM

August 2021

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

type="datetime-local"

```
<label for="visit">Date of visit</label>  
<input name="visitDate" id="visit" type="date" />
```

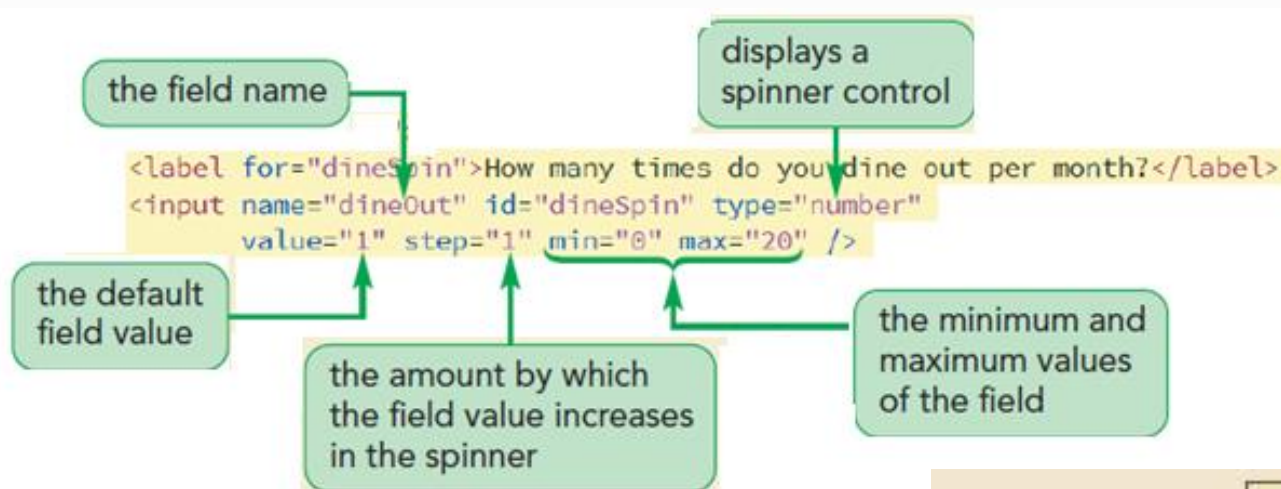
sets the data type of the visitDate field to "date"

04:21 PM

type="time"

(i) Spinner control - entering numeric values

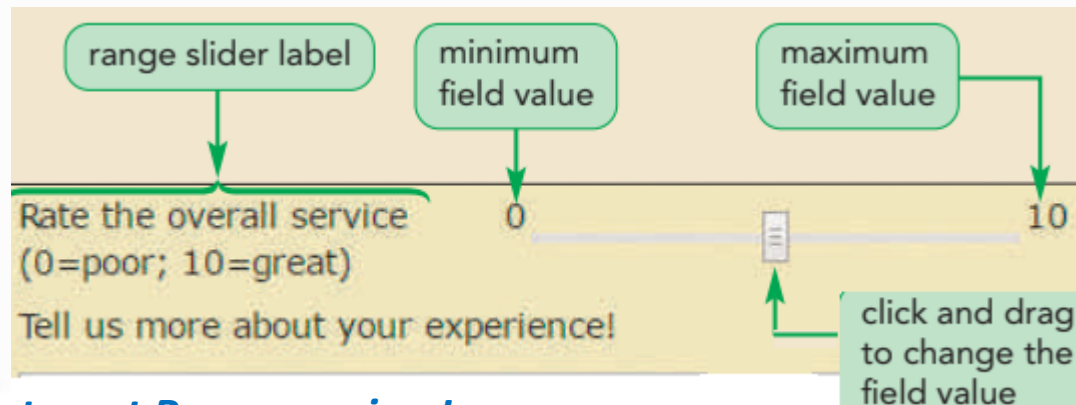
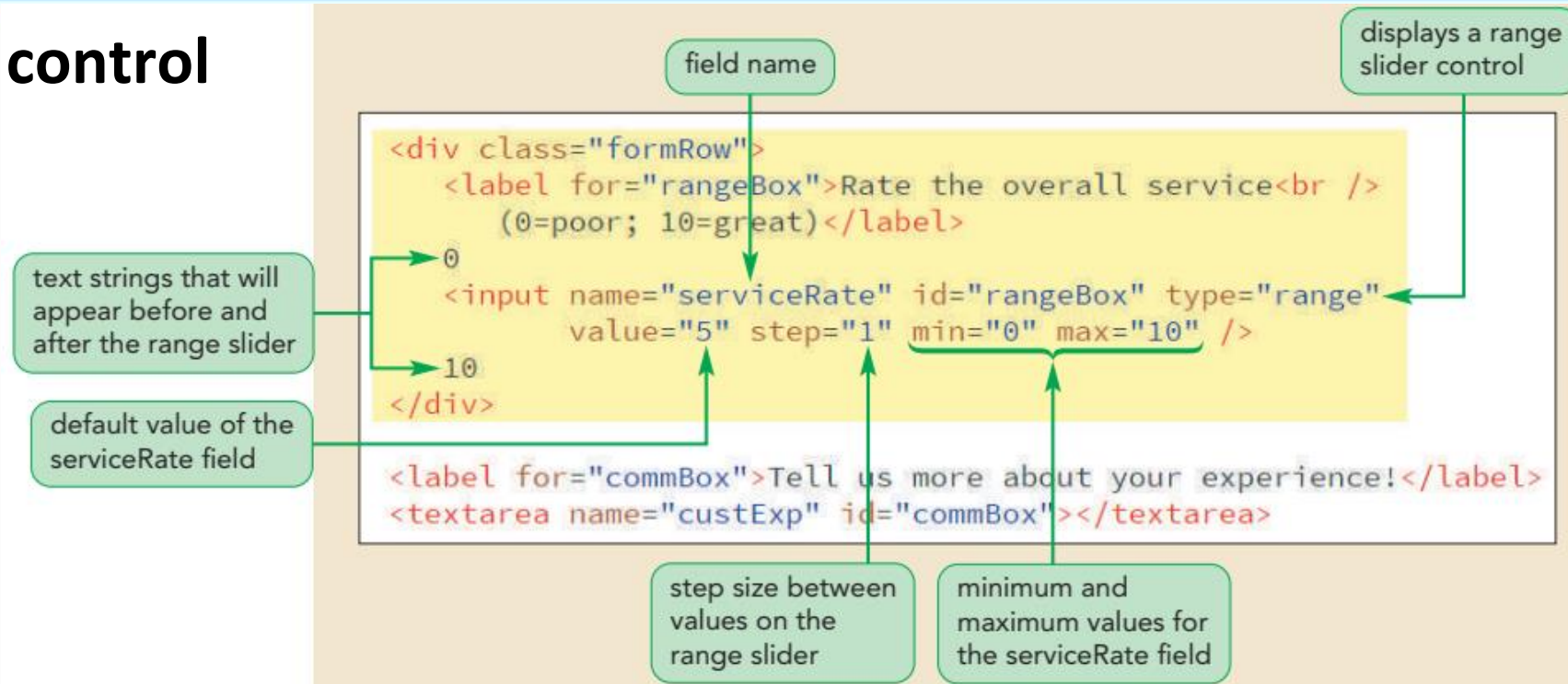
- **Spinner control:** Displays an up or down arrow to increase or decrease the field value by a set amount
- To create a spinner control, **apply the input element** using the **number** data type



The screenshot shows a web form with two sections. The top section is a multi-select dropdown menu titled "Where did you hear about us? (select all that apply)" with options: Internet, Magazine, Newspaper, Word of Mouth, and Other. The bottom section contains a spinner control for the question "How many times do you dine out per month?". The spinner shows the value "1". To the right of the spinner is a button with up and down arrows, labeled "click to increase or decrease the field value". Below the spinner is a checkbox labeled "Add me to your mailing list for great coupons and specials!".

(j) Creating a range slider control

- A **slider control** limits a numeric field to a range of possible values
- To create a slider control, apply the range data type in the input element




(k) The Output element

- A **container element** into which a site or app can inject the results of a calculation or the outcome of a user action.
- The `<output>` tag is a newly added tag and was introduced in HTML5.
- **Example 1:**

```
<!DOCTYPE html>
<html>
<body>
<p>Sum up the value of slider and spin controls</p>
<form oninput="result.value=parseInt(a.value)+parseInt(b.value)">
  <input type="range" id="b" name="b" value="50" /> +
  <input type="number" id="a" name="a" value="10" /> =
  <output name="result" for="a b">60</output>
</form>
</body>
</html>
```

Sum up the value of slider and spin controls

 + = 60

Example 2:

```
<!DOCTYPE html>
<html>
<head> <title>Output Tag</title> </head>
<body>
  <p>Calculate the Sum of the two Numbers</p>
  <form oninput="res.value=parseInt(a.value)+parseInt(b.value);">
    <label>Enter First Value.</label><br>
    <input type="number" name="a" value=""/><br>
    +<br>
    <label>Enter First Value.</label><br>
    <input type="number" name="b" value=""/><br>
    =<br>
    Output is:<output name="res"></output>
  </form>
</body>
</html>
```

Calculate the Sum of the two Numbers

Enter First Value.

+

Enter First Value.

=

Output is:10

(I) The progress and meter element

▪ **Progress element**

- Displays an indicator showing the completion progress of a task, typically displayed as a progress bar.
- It is mostly used to show the progress of activity like a file uploading or downloading on the web.

▪ **Meter element**

- Used to measure data within a given range.
- It represents either a scalar value within a known range or a fractional value.
- It is also known as a gauge.
- It should be used to display disk usage, voting population etc.

Example:

Value - defines that how much work the task has been completed.
Max - defines that how much work the task requires in total.

```
<!DOCTYPE html>
<html>
<head>
<title>Output Tag</title>
</head>
<body>
  Downloading:
  <progress value="43" max="100">43%</progress>
  <br><br>
  <label for="fuel">Fuel level:</label>
  <meter id="fuel"
    min="0" max="100"
    low="33" high="66"
    optimum="80"
    value="50">
    at 50/100
  </meter>
</body>
</html>
```

Value - a mandatory attribute which is used to specify a value in numbers (integer or floating point)..
High and low - specifies a range that is considered as high and low value respectively.
Max and min - specifies the maximum and minimum value defined in the range.
Optimum - specifies the optimum value for the gauge .

Downloading: 

Fuel level: 

(m) Working with Form Buttons

- **Form buttons:** A type of form control that performs an action
- **Actions performed**
 - *Run a command* from a program linked to the web form
 - *Submit the form* to a program running on the web server
 - *Reset the form* fields to their default values
- **Types of form buttons**
 - *Command button*
 - *Submit button*
 - *Reset button*
 - *Custom button*

Submit and Reset button

- **Submit button** - submits a form to the server for processing when clicked.
- **Reset button** - resets a form, changing all fields to their default values and deleting any field values that a user has entered
- Created using **input elements** with the type attribute set to “**submit**” and “**reset**” respectively.
- Example:

`<input value="text" type="submit" />`

`<input value="text" type="reset" />`

where **text** is the string that appears on the button

Command button

- Runs a program that *affects the content of a page* or the actions of a browser
- Created using the **input element** with the type **attribute set to button**
- e.g. `<input value="text" onclick="script" type="button" />`
 - text* - string that appears on the button
 - script* - the name of the program code that is run when the button is clicked

Custom button

- Appearance of a command, submit, and reset button is determined by the browser
- For more control over a button's appearance use the **button element**
- E.g. `<button type="text"> content </button>`

Where type attribute specifies the button type and the content are HTML elements placed within the button

```
<label for="city">City</label>
<input name="custCity" id="city" type="text" value="Ormond Beach"/>

<label for="state">State</label>
<input name="custState" id="state" type="text" value="FL" />
```

sets the default value
for the custCity field

sets the default value
for the custState field

Default value

- Specified using the **value attribute**

Placeholder

- A text that appear within a form control, providing a hint about the kind of data that should be entered into a field
- Defined using the **placeholder attribute**

The diagram shows a form titled "Customer Information" with the following fields and annotations:

- Name***: Input box with placeholder text "first and last name".
- Street address**: Input box.
- City**: Input box with default value "Ormond Beach".
- State**: Input box with default value "FL".
- Postal code**: Input box with placeholder text "nnnnn (-nnnn)".
- Phone number**: Input box with placeholder text "(nnn) nnn-nnnn".
- E-mail***: Input box.

Annotations on the left side:

- formatted field set legend (points to the form title)
- default value for the custCity field (points to the City input box)
- placeholder text for the phone input box (points to the Phone number input box)

Annotations on the right side:

- placeholder text for the name input box (points to the Name input box)
- default value for the custState field (points to the State input box)
- placeholder text for zip input box (points to the Postal code input box)

3. Validating a Web Form

- **Validation**: Process of ensuring that a user has supplied valid data
- **Types of validation**
 - Server-side validation – validation occurs on the web server
 - Client-side validation – validation occurs in the user's browser

What to validate?

- **Identifying Required Values**
 - The first validation test is to verify if data is supplied for all the required data fields
 - Add the **required attribute** to the control to identify the required data fields
- **Defining the Length of the Field Value**
 - For example the syntax to define the maxlength attribute is `<input maxlength="value"/>`
 - **Example:** `<input name="custZip" maxlength="5" />`
 - The **maxlength** attribute does not distinguish between *characters and digits*

Validating a Web Form



- **Validating Based on Data Type**

- A form fails the validation test if the data values entered into a field do not match the field type
- **Example:**
 - ✓ Entering a **nonnumeric data** for data field with the number type
 - ✓ Providing **invalid email or url types** that does not match the **format of a URL**

- **Testing for a Valid Pattern**

- To test whether a field value follows a valid pattern of characters, test the character string against a regular expression
- Regular expression or regex is a concise description of a character pattern
- To validate a text value against a regular expression, add the pattern attribute to the input element.
- **Example:** see next slide

Data Validation (examine the code)

The **pattern** attribute specifies the general pattern that the characters in the field value must follow.

The **min** and **max** attributes define the range of possible field values; the **step** sets the interval between values.

The **datalist** element defines a set of suggested field values.

The **submit** data type creates a button to submit the form for processing.

```
<label for="name">Name *</label>
<input name="custName" id="name" type="text" required />

<label for="phone">Phone number</label>
<input name="custPhone" id="phone" type="tel"
pattern="^\d{10}$|*(\(\d{3}\)\s*)?\d{3}[\s-]?\d{4}$" />

<label for="dineSpin">How often do you dine out per month? </label>
<input name="dineOut" id="dineSpin" type="number"
value="1" step="1" min="0" max="20" />

<label for="dish">What's your favorite dish?</label>
<input name="favDish" id="dish" type="text" list="dishType" />
<datalist id="dishType">
  <option value="Big Kahuna Pizza" />
  <option value="BBQ Chicken Pizza" />
  <option value="Pasta Rolls" />
  <option value="Pasto Artichoke Pizza" />
</datalist>

<label for="rangeBox">Rate the overall service at Red Ball<br />
(0=poor; 10=great)</label>
<input name="serviceRate" id="rangeBox" type="range"
value="5" step="1" min="1" max="10" /> 10

<input type="submit" value="Submit My Survey" />
<input type="reset" value="Cancel" />
```

The **required** attribute indicates that a field value is required.

The **number** data type creates a spin box control for data entry.

The **reset** data type creates a button that restores the form to its default values.

The **range** data type creates a range slider for data entry.

Data Validation (look on the screen)

Customer Survey

Name* Alice Nichols

Phone number 555-7499

How often do you dine out per month? 6

What's your favorite dish? Big Kahuna Pizza

Rate the service at Red Ball (0=poor; 10=great)

Submit My Survey **Cancel**

The Submit My Survey button is used to submit the form to the server for processing.

The Cancel button is used to reset form fields to their default values, deleting any user input.

A spinner control is used to select a field value by clicking spin arrows to increase or decrease the value by a set amount.

The favDish field displays a suggested value from the dishType data list.

A range slider control is used to select a field value by dragging a slider across a range of values.

Use inline validation to highlight invalid data as it is being entered by the user.

Forms that contain invalid data generate error messages when submitted by the browser for processing.

Please match the requested format.

- The table model allows authors to arrange data into rows and columns of cells.
- Table cells may either contain "header" information or "data".
- Cells may span multiple rows and columns
- Table rows may be grouped into a *head*, *foot*, and *body sections*, (via the THEAD, TFOOT and TBODY elements, respectively).
- User agents may exploit the head/body/foot division to support scrolling of body sections independently of the head and foot sections.
- Authors may also group columns to provide additional structural information that may be exploited by user agents.
- Each table may have an associated caption that provides a short description of the table's purpose.

- Whenever you want to collect information from visitors you will need a form, which lives inside a `<form>` element.
- Information from a form is sent to web server in name/value pairs.
- Each form control is given a name, and the text the user types in or the values of the options they select are sent to the server.
- HTML5 introduces new form elements which make it easier for visitors to fill in forms.
- Form buttons are a type of form control that performs an action.
- The form buttons are *control*, *submit*, *reset* and *custom button*.
- Form validation is a process of ensuring that a user has supplied valid data,
- The validation can be performed either on the client or server side.

Exercise 1: Designing a Table

1. Discuss when to use table and things that are appropriate in tables.
2. Can a table nested, putting a table inside a single table cell?
3. What are the basic parts of a table, and which tags identify them?
4. **[Practical exercise]** Write a HTML code that produce the table below.

Day	Seminar		
	Schedule		Topic
	Begin	End	
Monday	8:00 a.m.	5:00 p.m.	Introduction to XML
			Validity: DTD and Relax NG
Tuesday	8:00 a.m.	11:00 a.m.	XPath
	11:00 a.m.	2:00 p.m.	
	2:00 p.m.	5:00 p.m.	XSL Transformations
Wednesday	8:00 a.m.	12:00 p.m.	XSL Formatting Objects

Exercise 2: Working with Form

Customer Information

Thank you for using our online ordering form for quick and easy orders. Please enter your name and phone number and whether you want to place an order for delivery or pickup. Note that required values are marked by an asterisk (*). If you want to place your order over the phone, call us at [\(386\) 555-7499](tel:3865557499).

Name* ✓

Phone*

☒ Delivery ☐ Pickup

Delivery Options

811 Beach Drive
Ormond Beach, FL 32175

Delivery Time (leave blank for earliest delivery)

07:45 AM

Pickup Options

Pickup Time (leave blank for earliest pickup)

Exercise 3: Working with Form

Build your Pizza

Everyone has their own pizza preferences. Build your special pizza by selecting from the ingredients below. You can divide your pizza into two regions to give your friends a wide choice of toppings and flavors.

Quantity

Pizza Size 

Pizza Crust

☐ Double Cheese ☒ Double Sauce

Meat Toppings

Location	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Pepperoni	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ham	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Pork	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Sausage	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Chicken	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Vegetable Toppings

Location	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mushrooms	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Green Peppers	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Onions	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tomatoes	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Jalapenos	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>

Reading Resources/Materials

Chapter 6,7:

- ✓ Jon Duckett; HTML and CSS Design and Build Websites, 2011 John Wiley & Sons, Inc., Indianapolis, Indiana

Chapter 2, 3:

- ✓ Paul Deitel, Harvey Deitel, Abbey Deitel (2014). Internet & World Wide Web How To Program (5th Edition), Pearson Education.

Chapter 9, 10:

- ✓ Musciano, C., Kennedy, B. (2006). HTML and XHTML: The Definitive Guide, 6th edition. O'Reilly Media, Inc.

Thank You
For Your Attention!!

Any Questions

