Web Development Using HTML

Internet Programming I: Chapter 2 – Part III

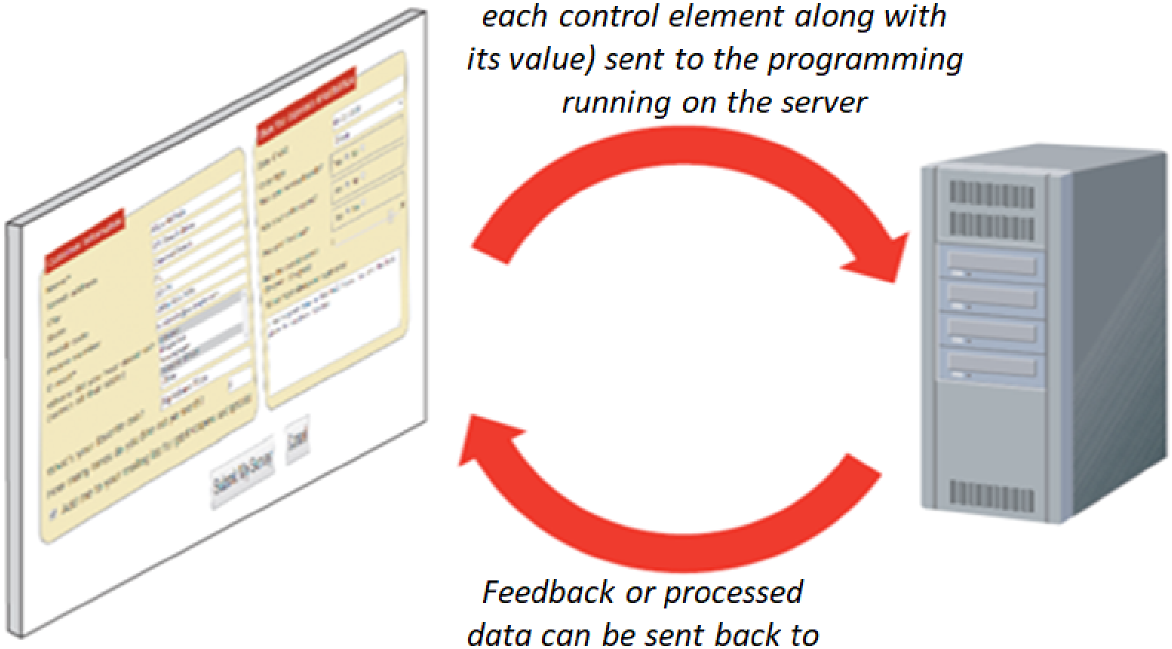
ADDIS ABABA SCIENCE AND TECHNOLOGY UNIVERSITY

Department of Software Engineering

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.
  - The backend app 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

Working with Web Form cont’d



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

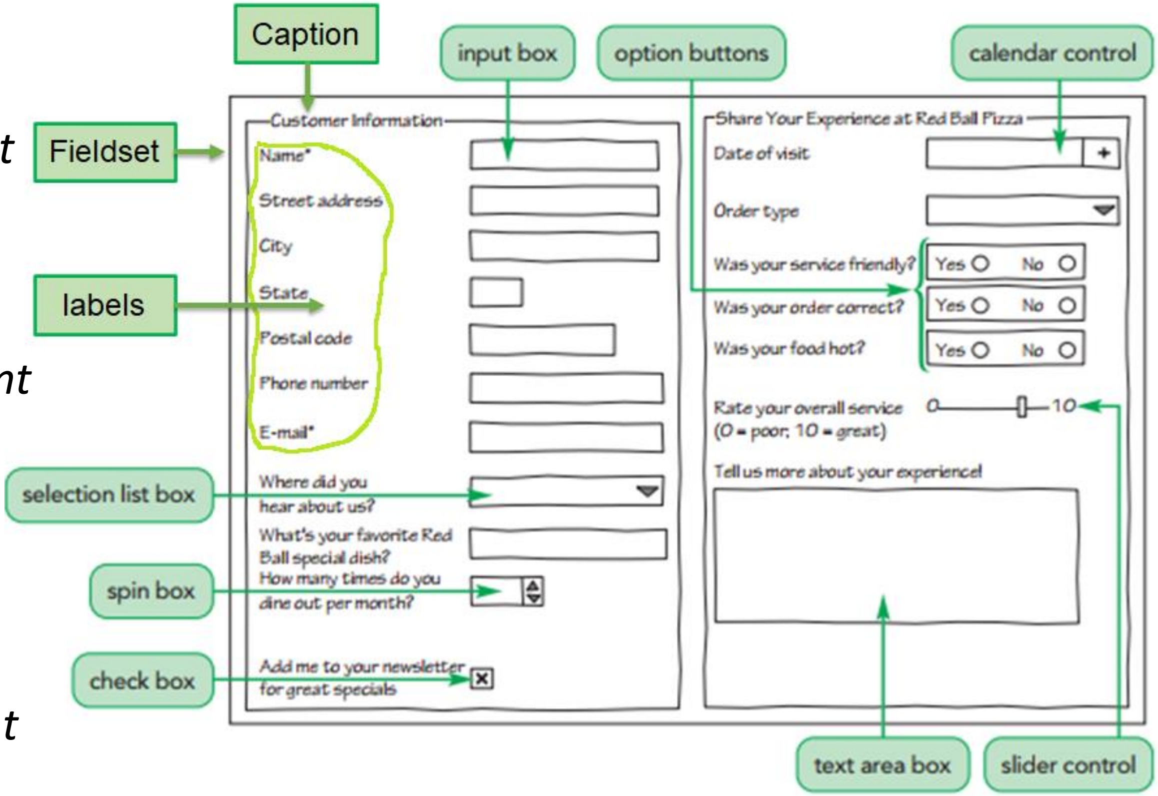
The server processes the submitted information using a server side program such as such as PHP, C#, or Java.

The server may also store the information in a database.

Feedback or processed data can be sent back to the browser

Working with Web Form cont’d

- 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*


Caption
input box
option buttons
calendar control
Fieldset
labels
selection list box
spin box
check box
text area box
slider control
Customer Information
Name*
Street address
City
State
Postal code
Phone number
E-mail*
Where did you hear about us?
What's your favorite Red Ball special dish?
How many times do you dine out per month?
Add me to your newsletter for great specials
Share Your Experience at Red Ball Pizza
Date of visit
Order type
Was your service friendly?
Was your order correct?
Was your food hot?
Yes ○ No ○
Rate your overall service (0 = poor; 10 = great)
0
10
Tell us more about your experience!

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

Creating Web forms

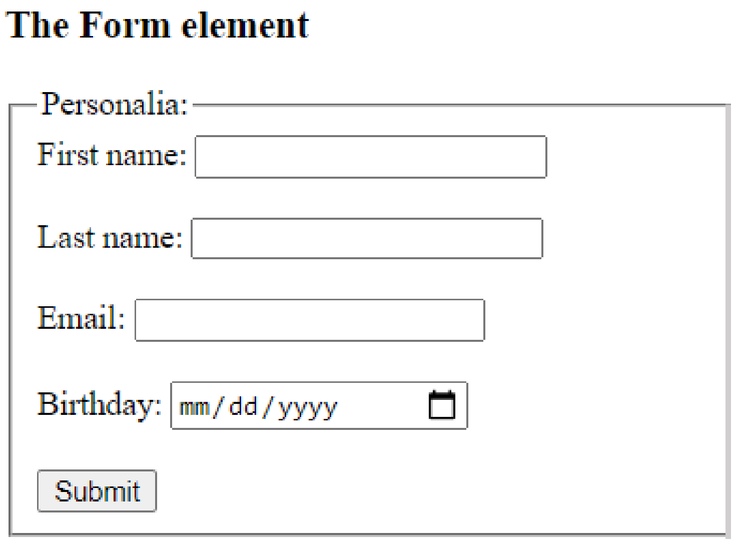
- Web forms are marked using the form element

```
<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

Creating Web forms example

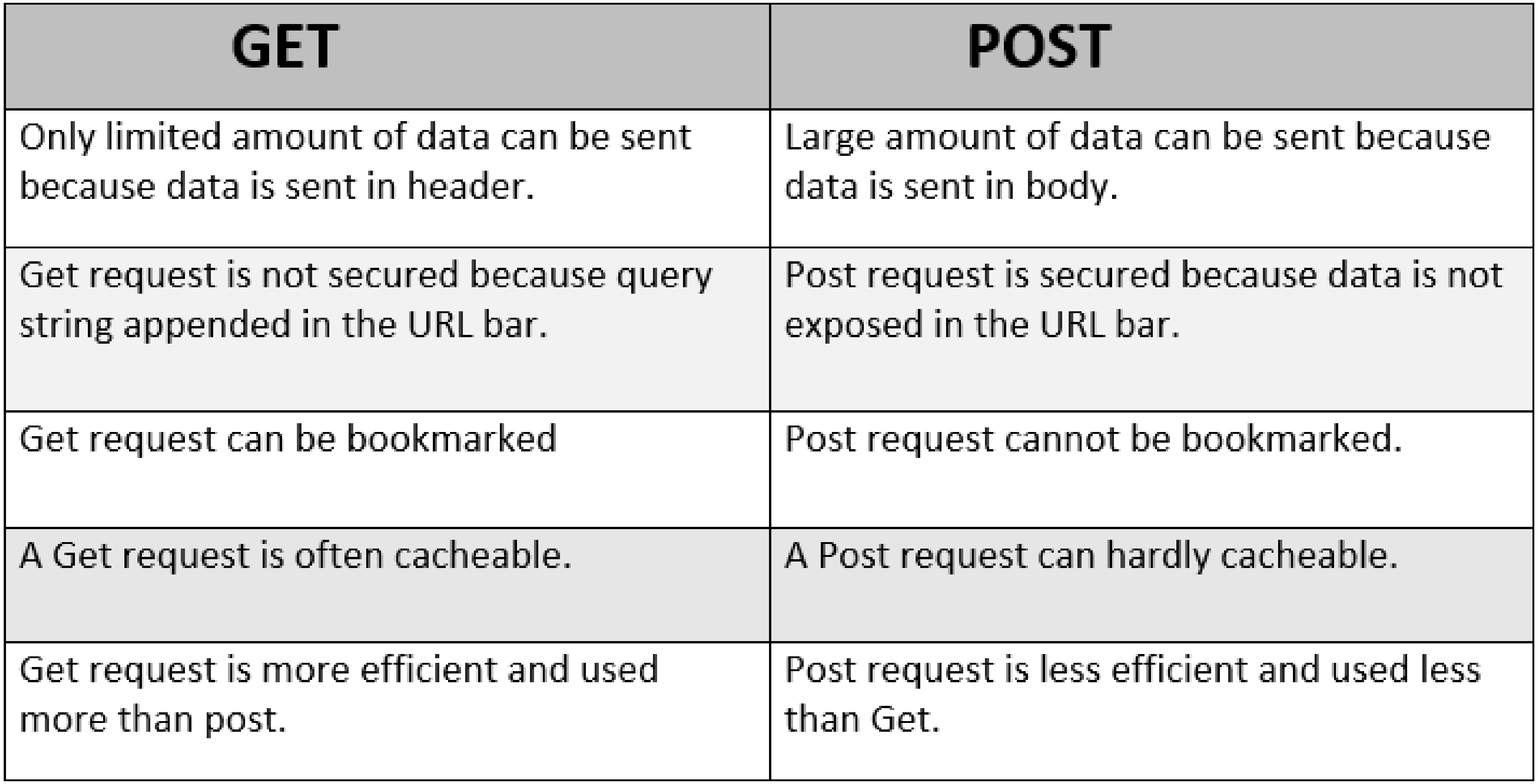
```
<html>
<body>

<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>
```



Most common form attributes

Attribute	Value	Description
action	*URL*	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	*character_set*	Specifies the character encodings that are to be used for the form submission
autocomplete	*On / off*	*Specifies whether a form should have autocomplete or not*
name	String	*The name of the form. The value must be unique among the form elements and also must not be the empty*
target	*_blank / _self / _parent / _top*	*Specifies where to display the response that is received after submitting the form*

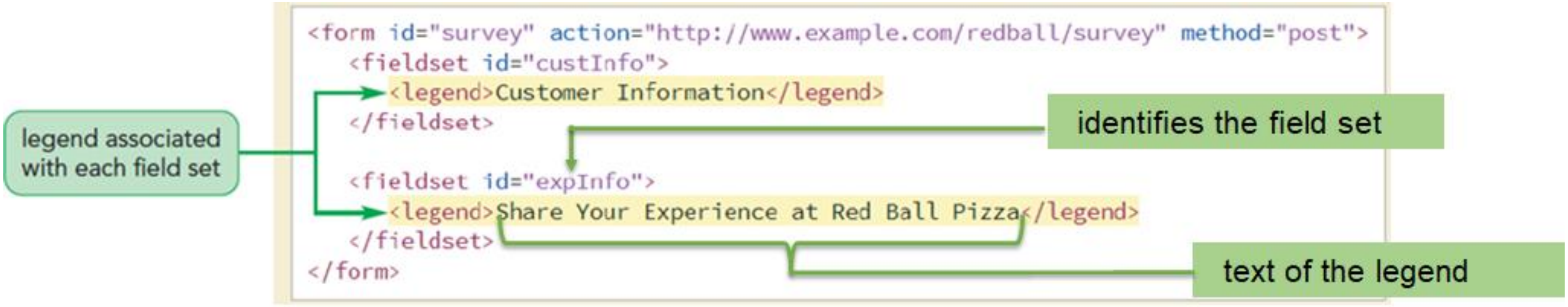
GET vs POST methods



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.

Grouping Form 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



```
<form id="survey" action="http://www.example.com/redball/survey" method="post">
   <fieldset id="custInfo">
      <legend>Customer Information</legend>
   </fieldset>

   <fieldset id="expInfo">
      <legend>Share Your Experience at Red Ball Pizza</legend>
   </fieldset>
</form>
```

legend associated with each field set

identifies the field set

text of the legend

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

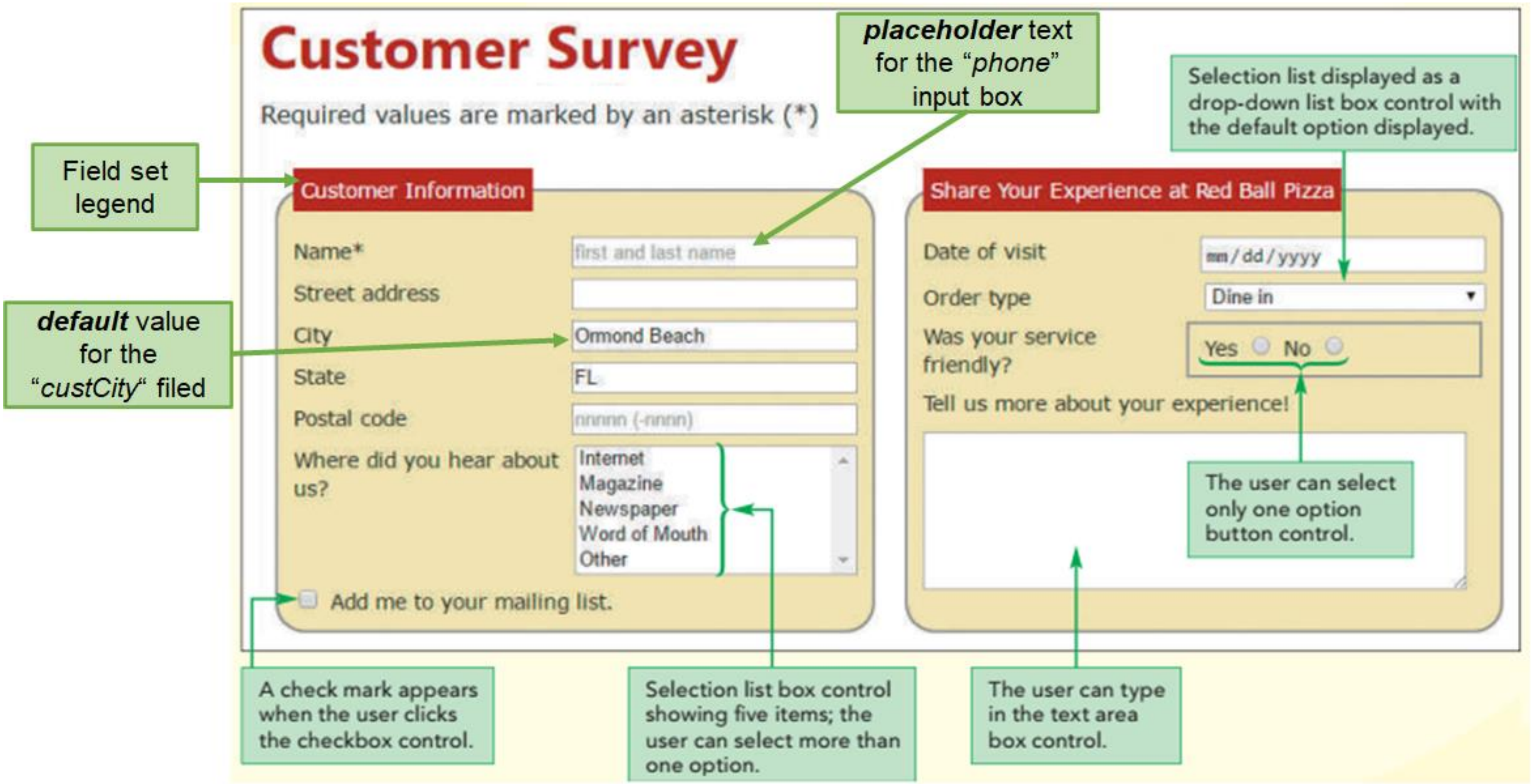
- The "type" attribute of input element can be various types, which defines data field
- List the types of <input> element given in the table

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.

HTML5 added new types on <input> element

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

Design and layout of the survey form example



Customer Survey

Required values are marked by an asterisk (*)

Customer Information

Name*	first and last name
Street address	
City	Ormond Beach
State	FL
Postal code	nnnnn (-nnnn)
Where did you hear about us?	Internet Magazine Newspaper Word of Mouth Other

□ Add me to your mailing list.

Share Your Experience at Red Ball Pizza

Date of visit	mm/dd/yyyy
Order type	Dine in ▾
Was your service friendly?	Yes ○ No ○

Tell us more about your experience!

Field set legend

placeholder text for the "*phone*" input box

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

default value for the "*custCity*" filed

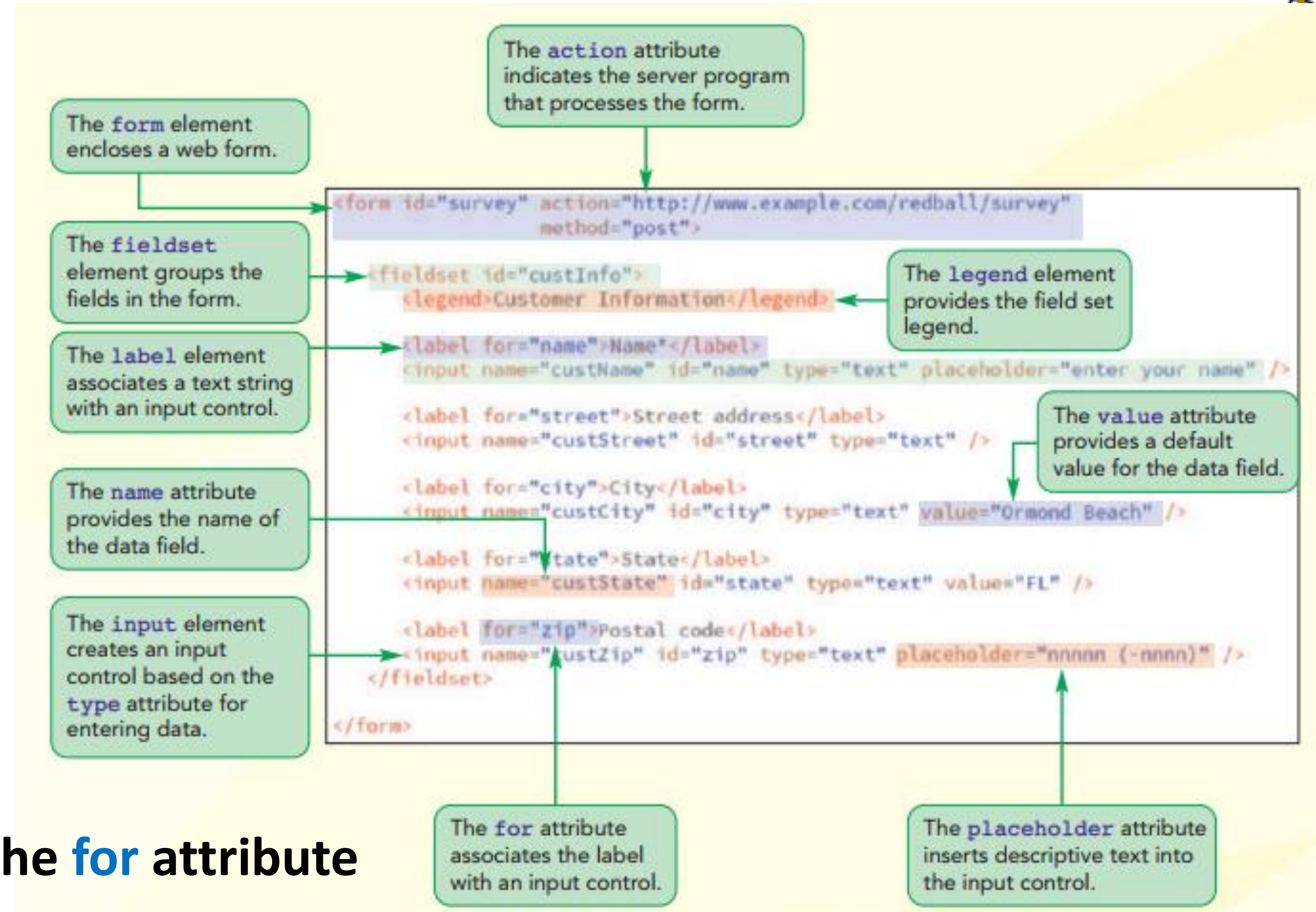
The user can select only one option button control.

A check mark appears when the user clicks the checkbox control.

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

The user can type in the text area box control.

Design and layout of the survey form example



The form element encloses a web form.

The action attribute indicates the server program that processes the form.

```
<form id="survey" action="http://www.example.com/redball/survey"
                  method="post">

   <fieldset id="custInfo">
      <legend>Customer Information</legend>

      <label for="name">Name*</label>
      <input name="custName" id="name" type="text" placeholder="enter your name" />

      <label for="street">Street address</label>
      <input name="custStreet" id="street" type="text" />

      <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" />

      <label for="zip">Postal code</label>
      <input name="custZip" id="zip" type="text" placeholder="nnnnn (-nnnn)" />
   </fieldset>

</form>
```

The fieldset element groups the fields in the form.

The legend element provides the field set legend.

The label element associates a text string with an input control.

The value attribute provides a default value for the data field.

The name attribute provides the name of the data field.

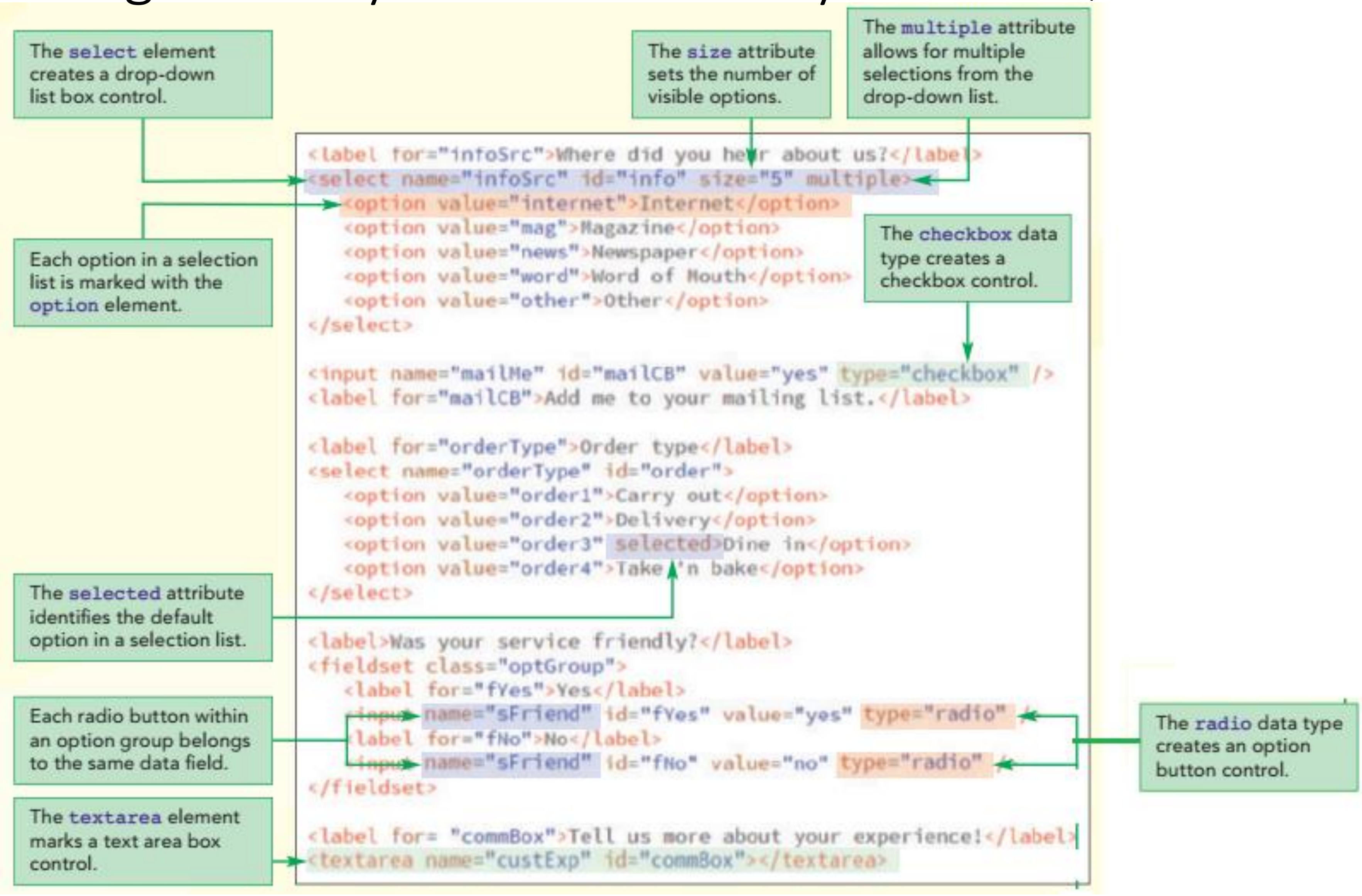
The input element creates an input control based on the type attribute for entering data.

The for attribute associates the label with an input control.

The placeholder attribute inserts descriptive text into the input control.

- **The for attribute**
  - When used with the <label> elements specifies which form element a label is bound to

Design and layout of the survey form example



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

The size attribute sets the number of visible options.

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

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

The checkbox data type creates a checkbox control.

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 radio data type creates an option button control.

The textarea element marks a text area box 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>
```

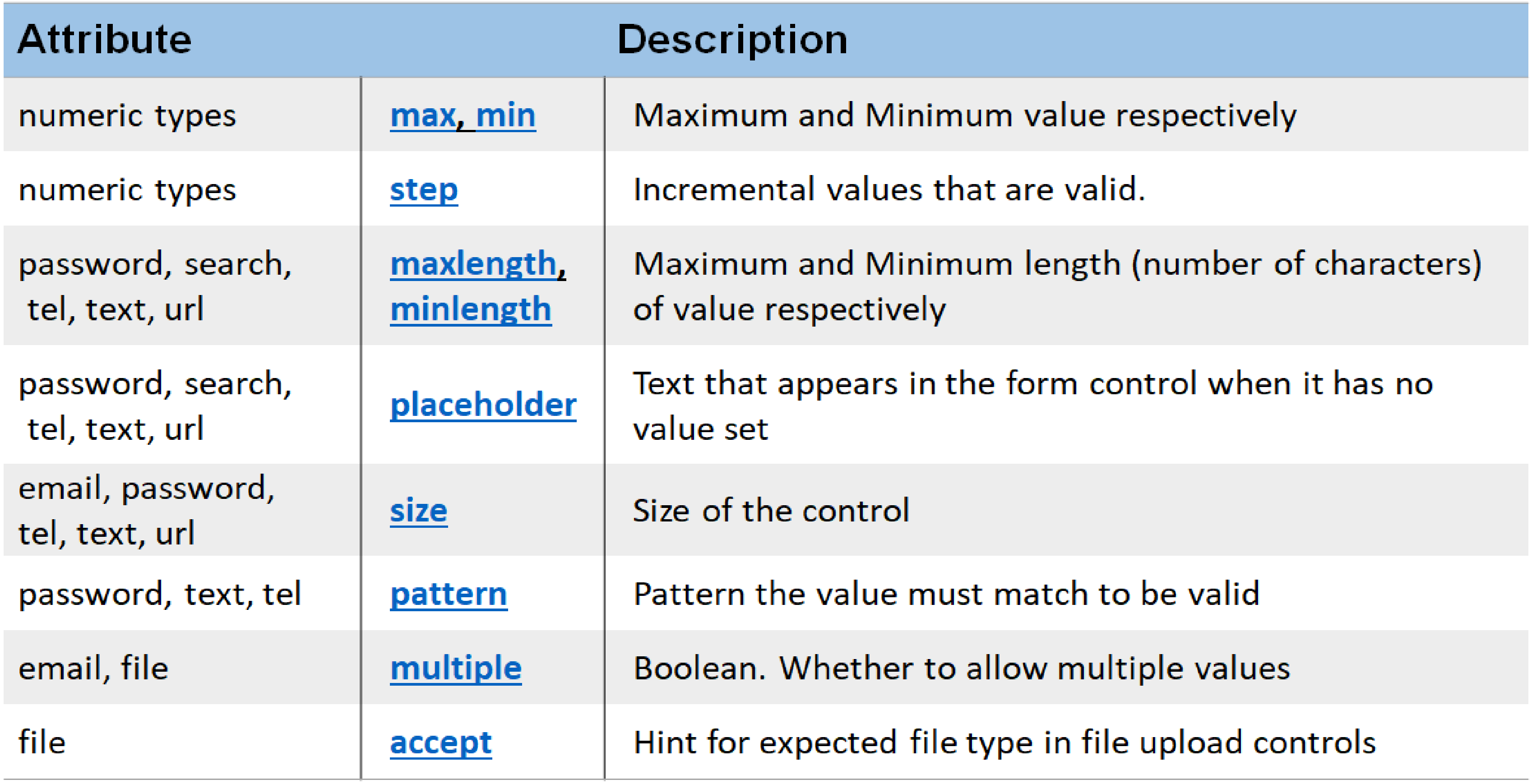
Common Attributes to all Input Types

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

Form Attributes

Attribute	Description
action	URL to use for form submission
enctype	Form data set encoding type to use for form submission
method	HTTP method to use for form submission
novalidate	Bypass form control validation for form submission

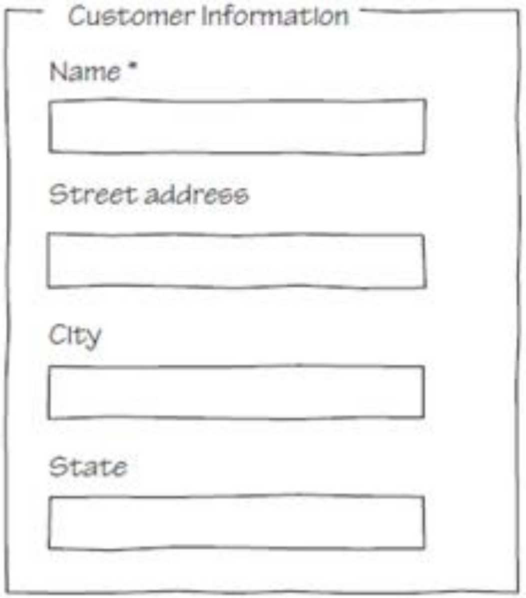
Other Attributes



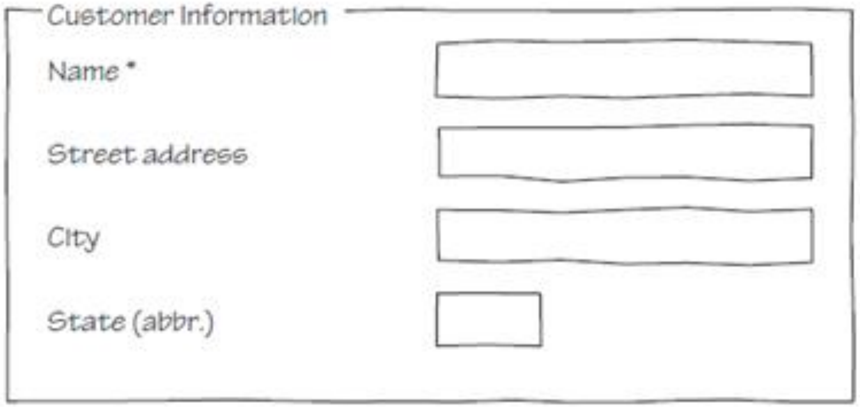
Attribute		Description
numeric types	max, min	Maximum and Minimum value respectively
numeric types	step	Incremental values that are valid.
password, search, tel, text, url	maxlength, minlength	Maximum and Minimum length (number of characters) of value respectively
password, search, tel, text, url	placeholder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

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*



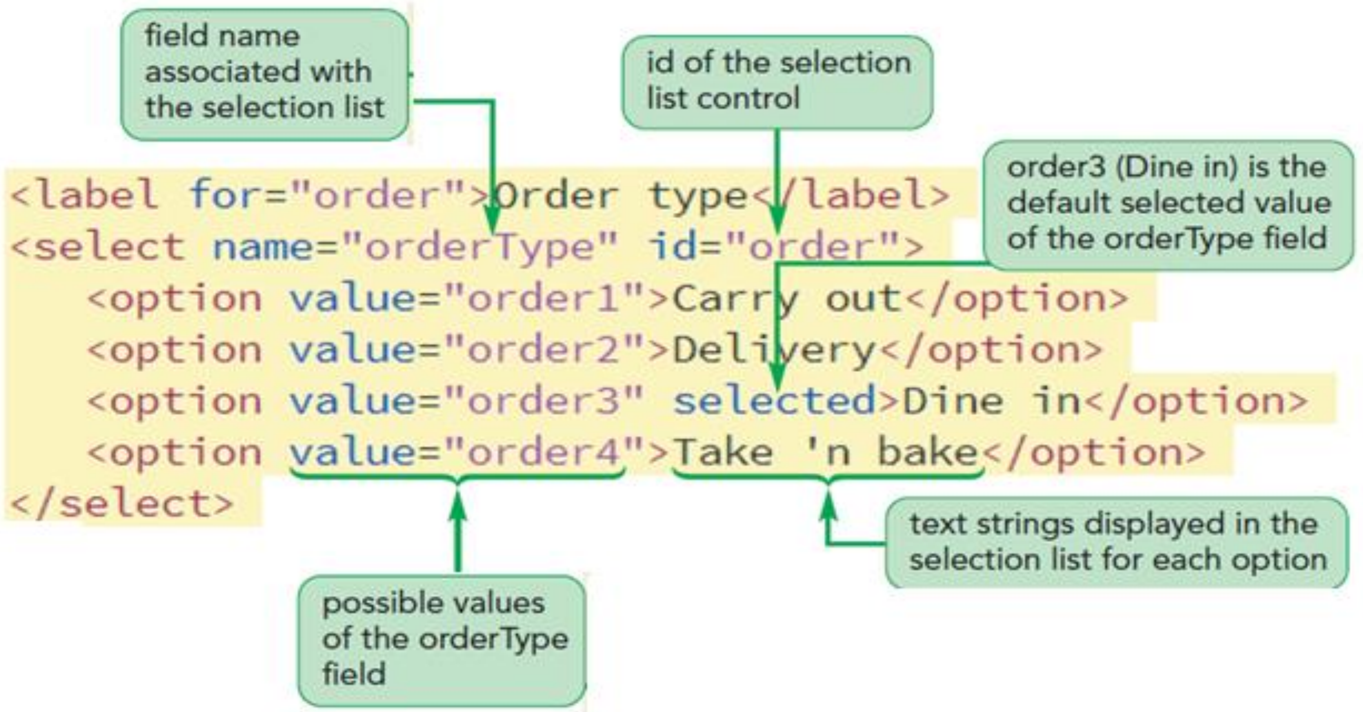
one-column layout



two-column layout

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

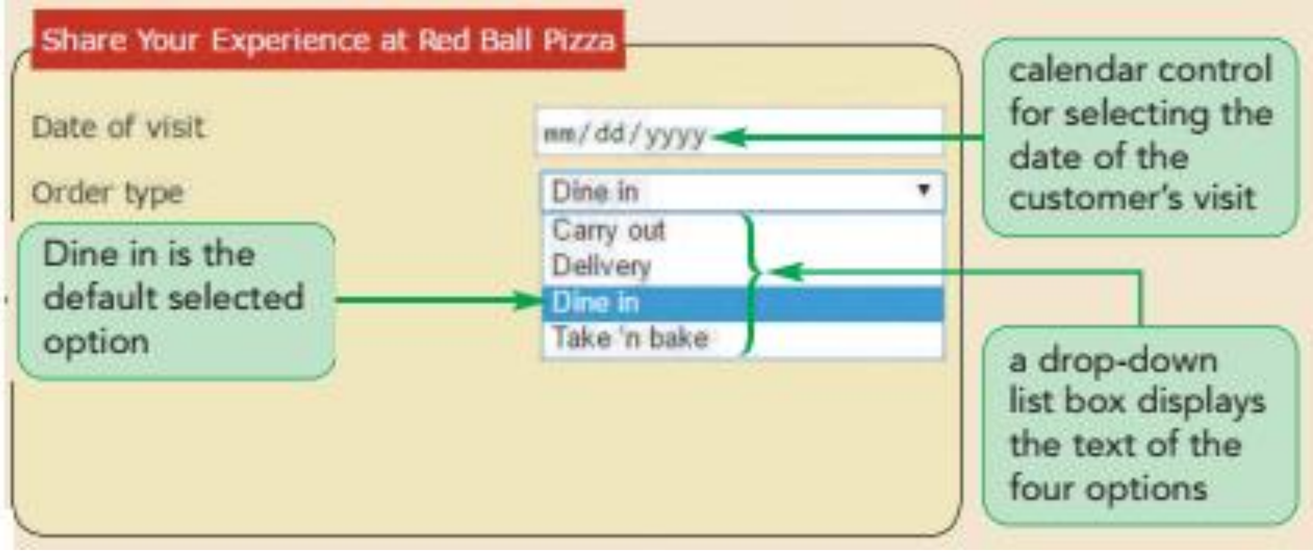


```
<label for="order">Order type</label>
<select name="orderType" id="order">
   <option value="order1">Carry out</option>
   <option value="order2">Delivery</option>
   <option value="order3" selected>Dine in</option>
   <option value="order4">Take 'n bake</option>
</select>
```

field name associated with the selection list
id of the selection list control
order3 (Dine in) is the default selected value of the orderType field
possible values of the orderType field
text strings displayed in the selection list for each option

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



Share Your Experience at Red Ball Pizza

Date of visit mm/dd/yyyy

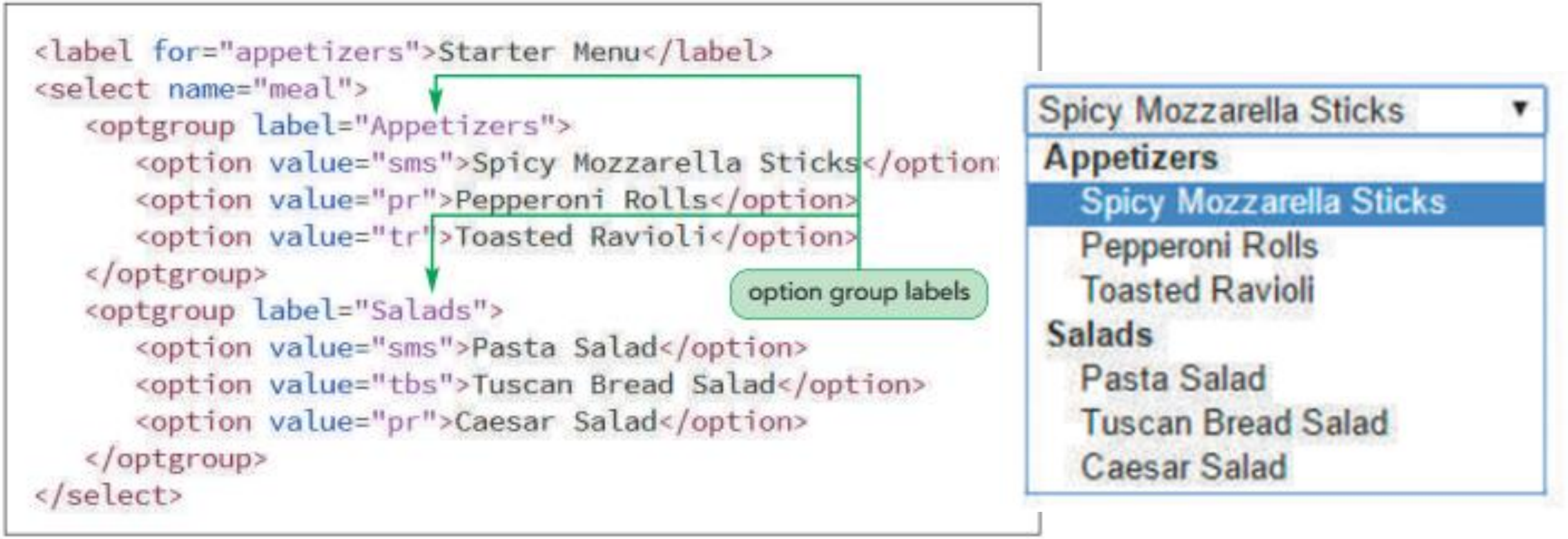
Order type Dine in ▾

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*

Grouping Selection Options

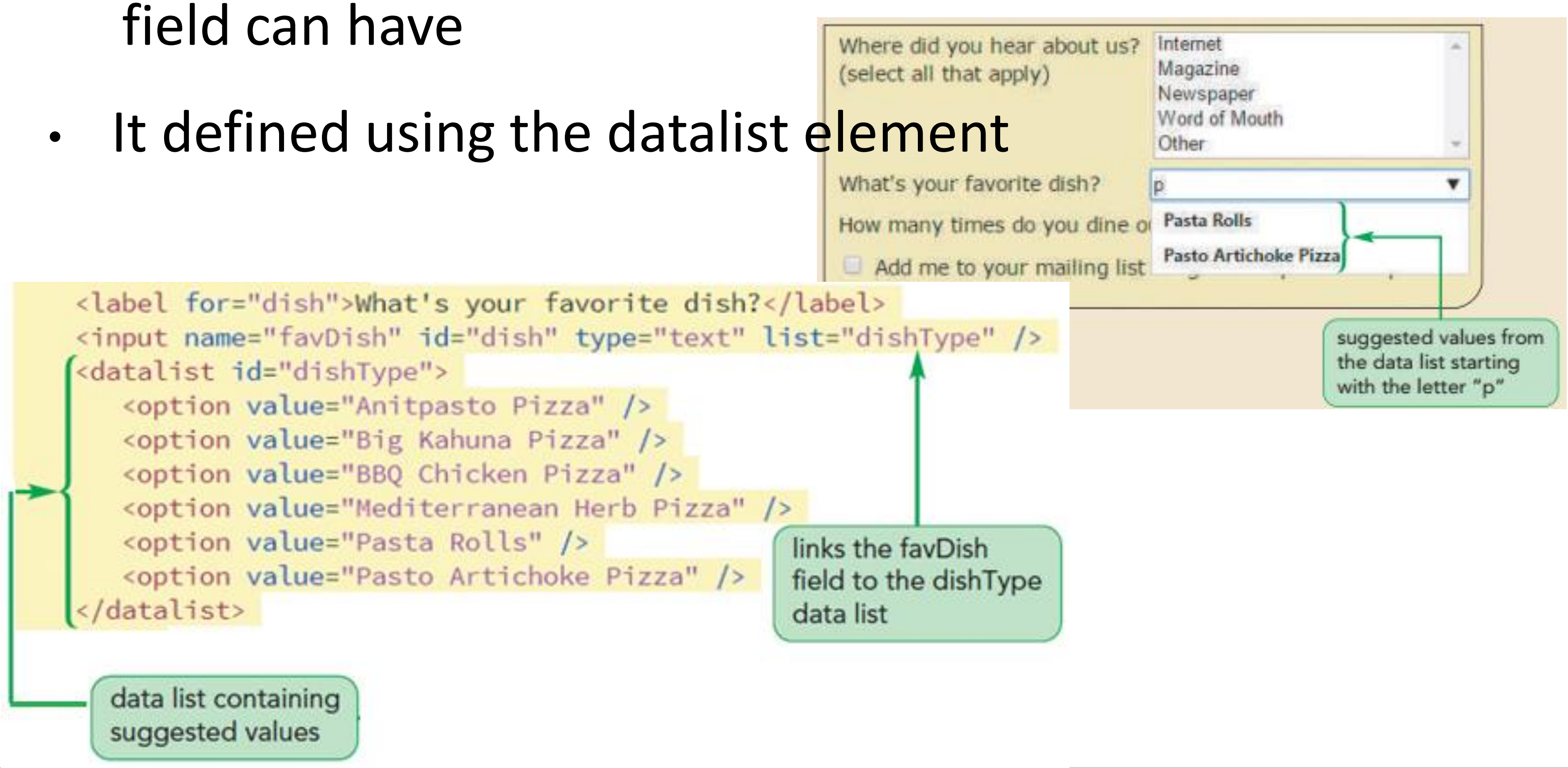
- The selection list options can be organized by placing them in option groups using the optgroup element



```
<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>
```

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



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

Internet
Magazine
Newspaper
Word of Mouth
Other

What's your favorite dish? p ▼

Pasta Rolls
Pasto Artichoke Pizza

How many times do you dine o

□ Add me to your mailing list

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

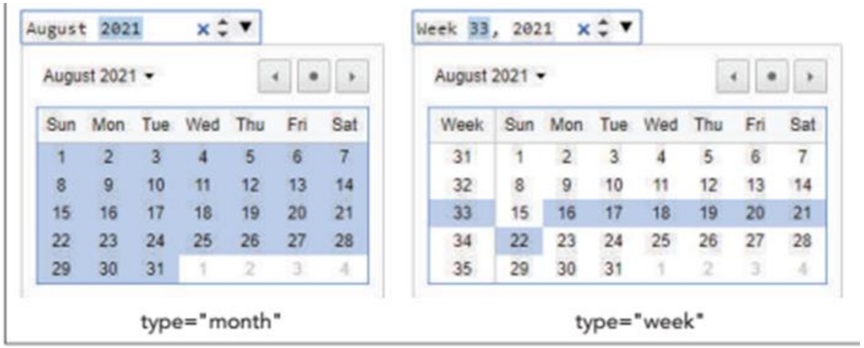
```
<label for="dish">What's your favorite dish?</label>
<input name="favDish" id="dish" type="text" list="dishType" />
<datalist id="dishType">
   <option value="Anitpasto Pizza" />
   <option value="Big Kahuna Pizza" />
   <option value="BBQ Chicken Pizza" />
   <option value="Mediterranean Herb Pizza"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

links the favDish field to the dishType data list

data list containing suggested values

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

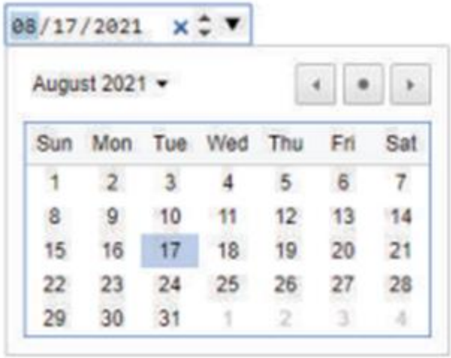


August 2021

Week 33, 2021

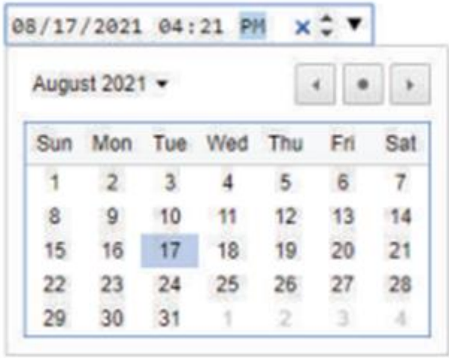
type="month"

type="week"



08/17/2021

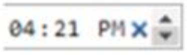
type="date"



08/17/2021 04:21 PM

type="datetime-local"

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



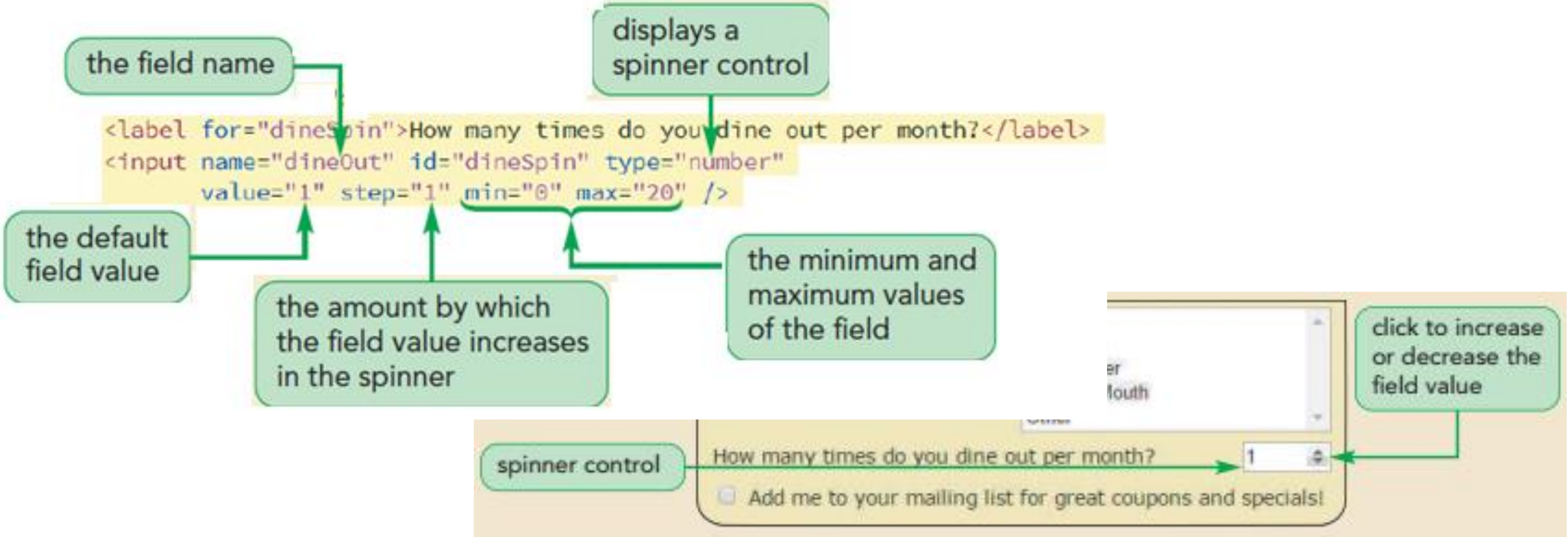
04:21 PM

type="time"

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

Spinner control - entering numeric values

- **pinner 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



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

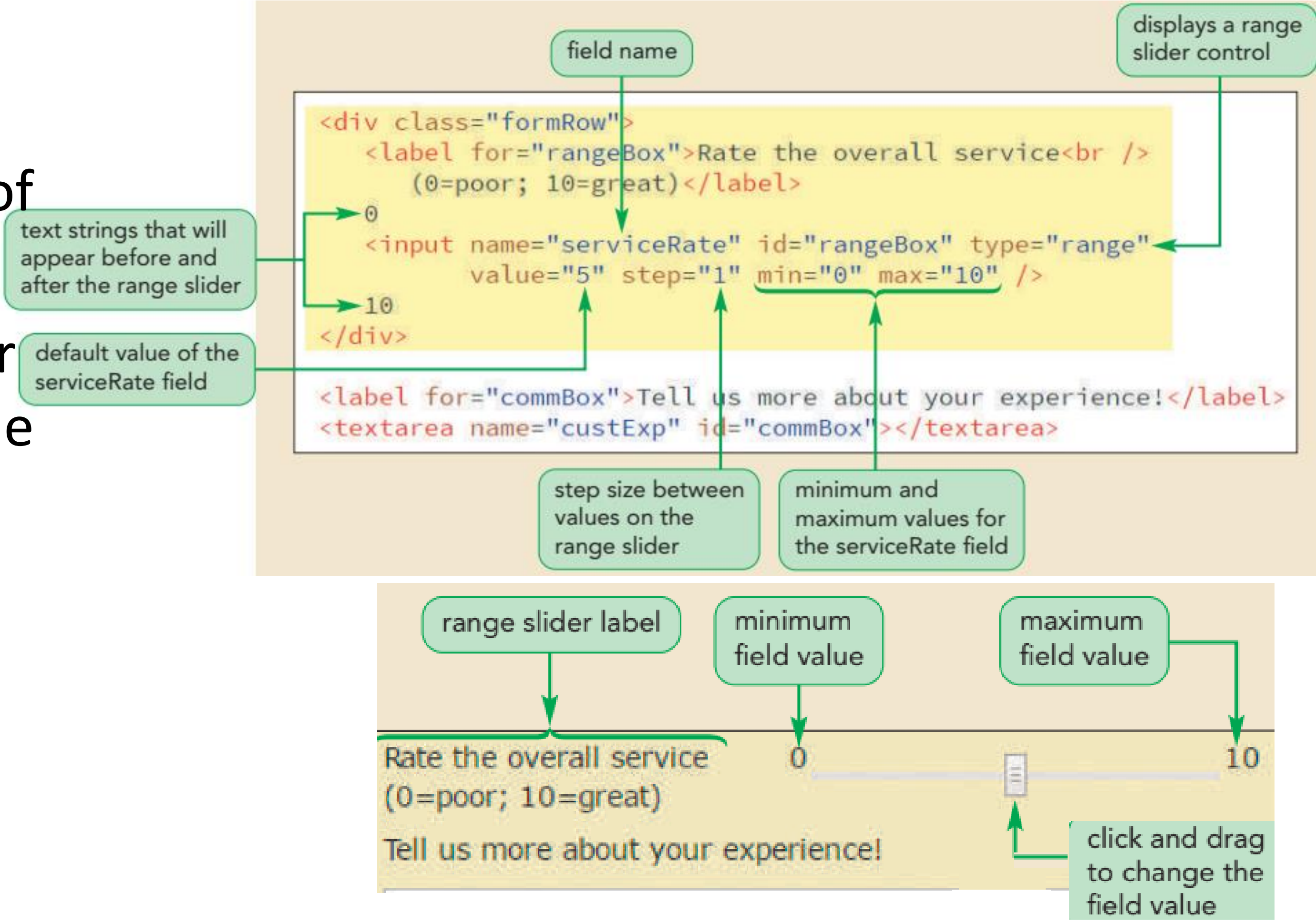
How many times do you dine out per month? 1

□ Add me to your mailing list for great coupons and specials!

the field name
displays a spinner control
the default field value
the amount by which the field value increases in the spinner
the minimum and maximum values of the field
click to increase or decrease the field value
spinner control
er
Mouth

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



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

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

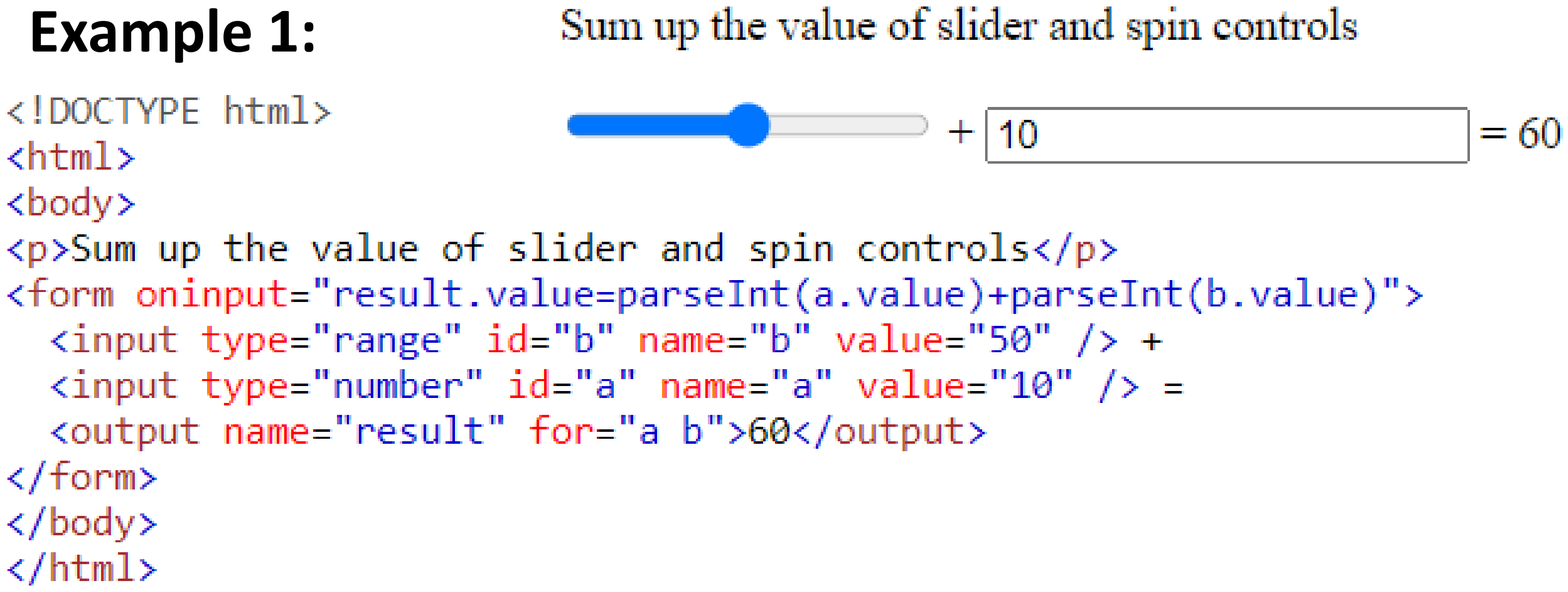
Rate the overall service (0=poor; 10=great) 0 10

Tell us more about your experience!

field name
displays a range slider control
text strings that will appear before and after the range slider
default value of the serviceRate field
step size between values on the range slider
minimum and maximum values for the serviceRate field
range slider label
minimum field value
maximum field value
click and drag to change the field value

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:**



Sum up the value of slider and spin controls

+ 10 = 60

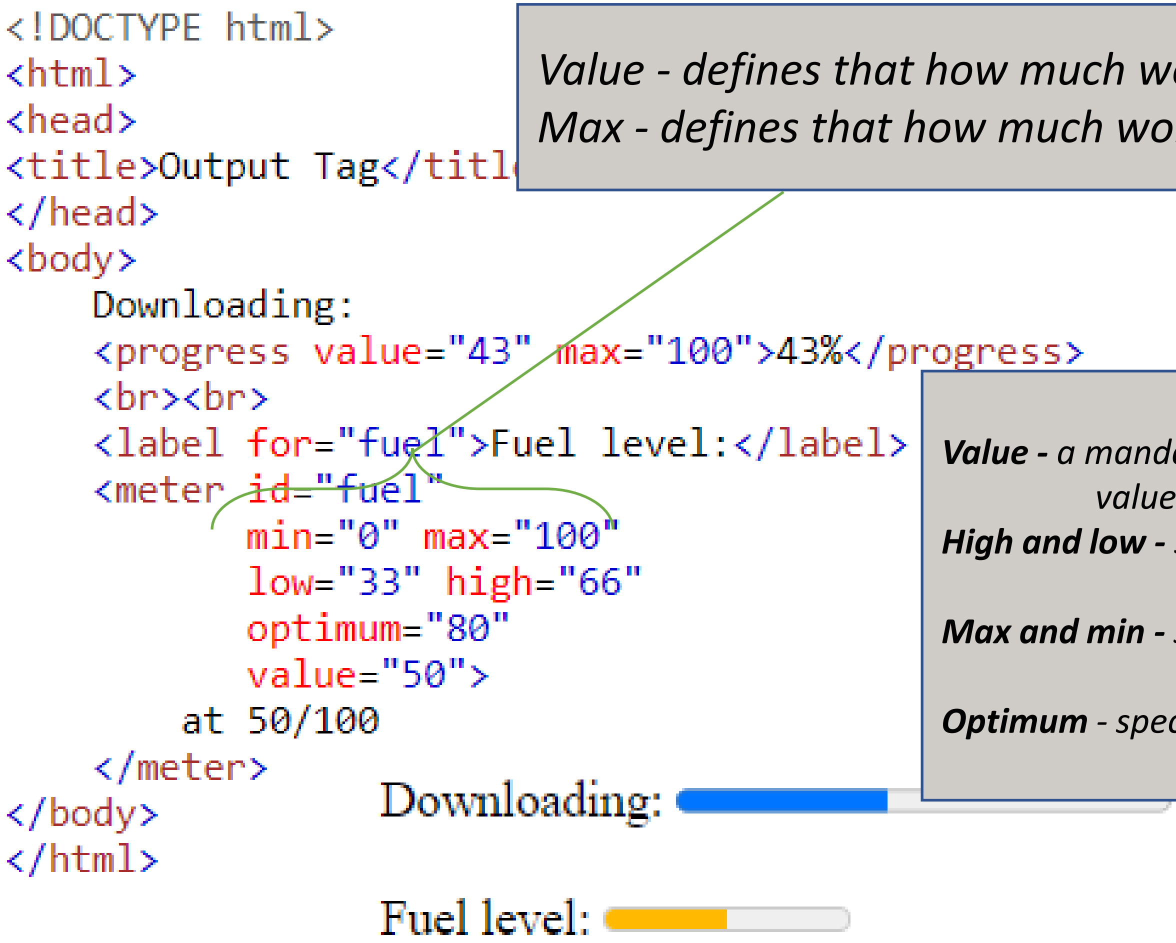
```
<!DOCTYPE html>
<html>
<body>
<p>Sum up the value of slider and spin controls</p>
<form oninput="result.value=parseInt(a.value)+parseInt(b.value)">
  <input type="range" id="b" name="b" value="50" /> +
  <input type="number" id="a" name="a" value="10" /> =
  <output name="result" for="a b">60</output>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

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 gause
  - It should be used to display disk usage, voting population etc.

The progress and meter element 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</titl
</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>
```

Downloading:

Fuel level:

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*

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 and Custom Buttons

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

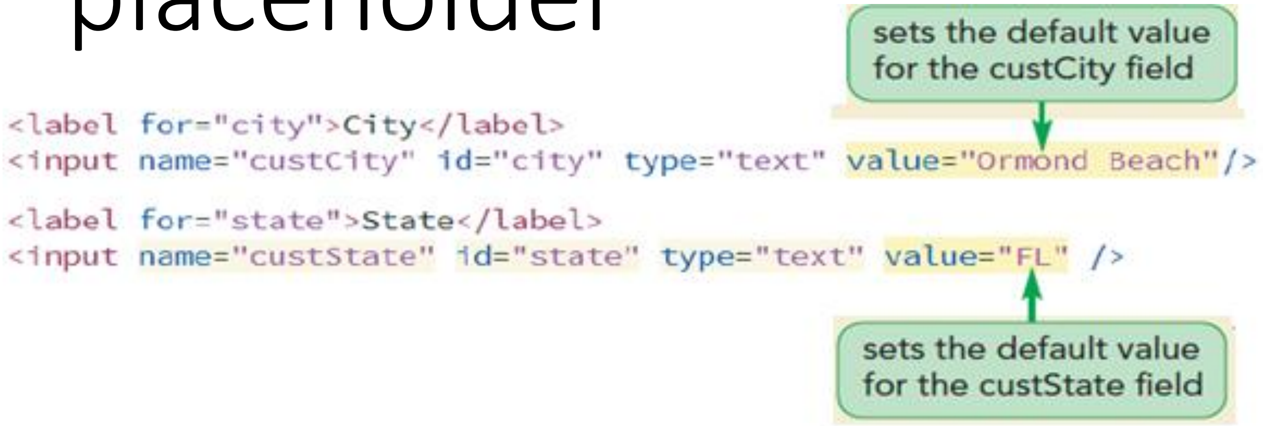
  script - the name of the program code that is run when the button is clicked

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

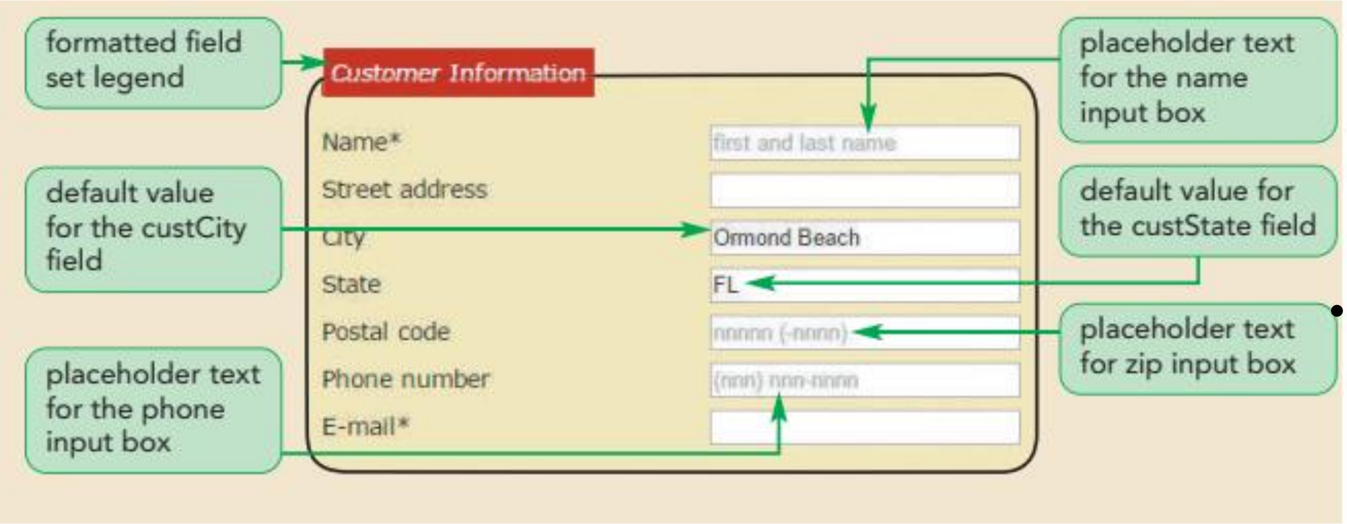
Defining default value and placeholder



sets the default value for the custCity field

```
<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 custState field



formatted field set legend

Customer Information

Name*	first and last name
Street address	
City	Ormond Beach
State	FL
Postal code	nnnnn (-nnnn)
Phone number	(nnn) nnn-nnnn
E-mail*	

placeholder text for the name input box

default value for the custCity field

default value for the custState field

placeholder text for zip input box

placeholder text for the phone input box

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

Form Validation

- 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?

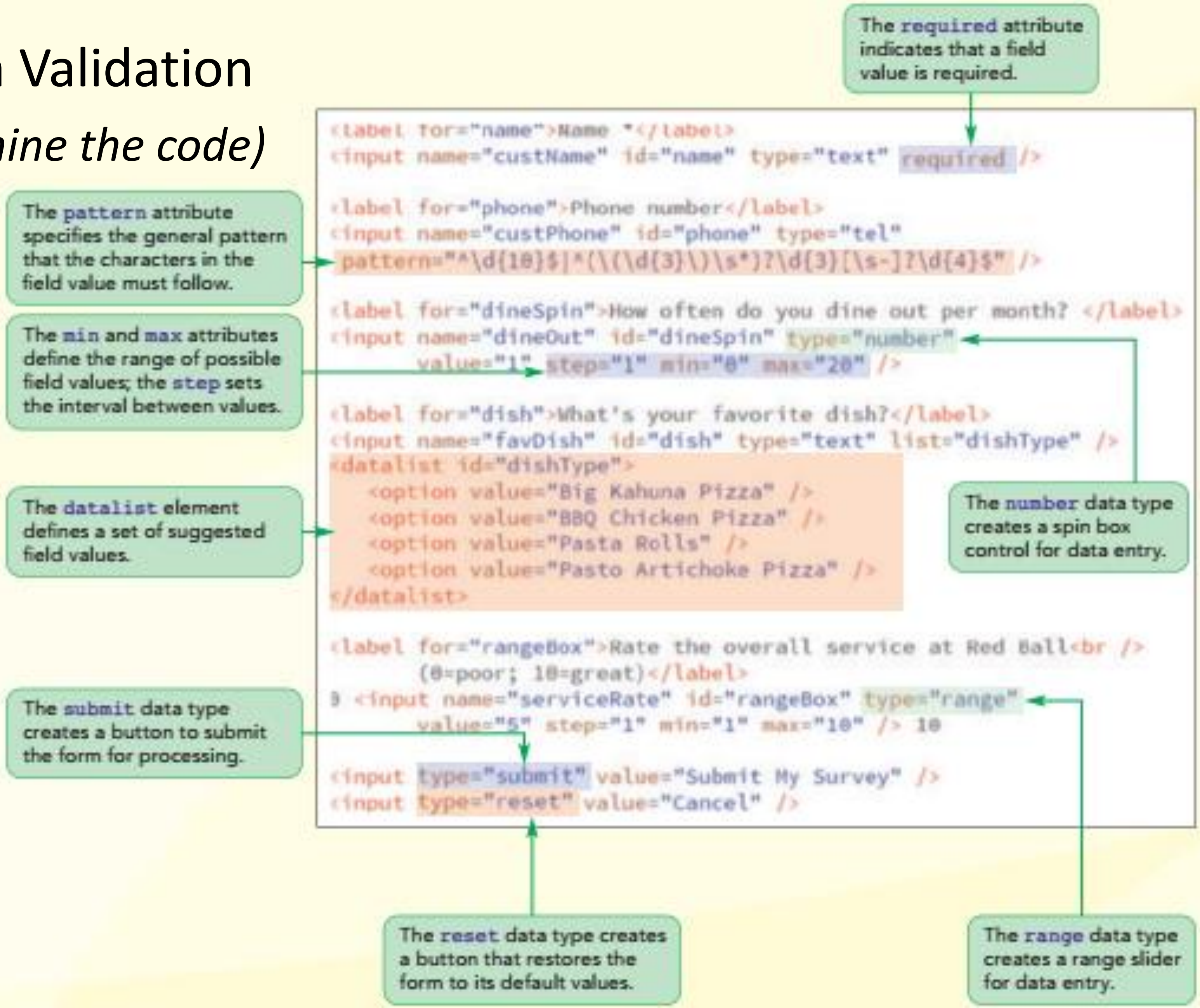
- Identifying Required Values
  - The first validation test is to verify if data is supplied for all the required data fields
  - Add the required attribute to the control to identify the required data fields
- Defining the Length of the Field Value
  - For example the syntax to define the maxlength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

Form Validation cont’d

- 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)



```
<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>
0 <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 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 number data type creates a spin box control for data entry.

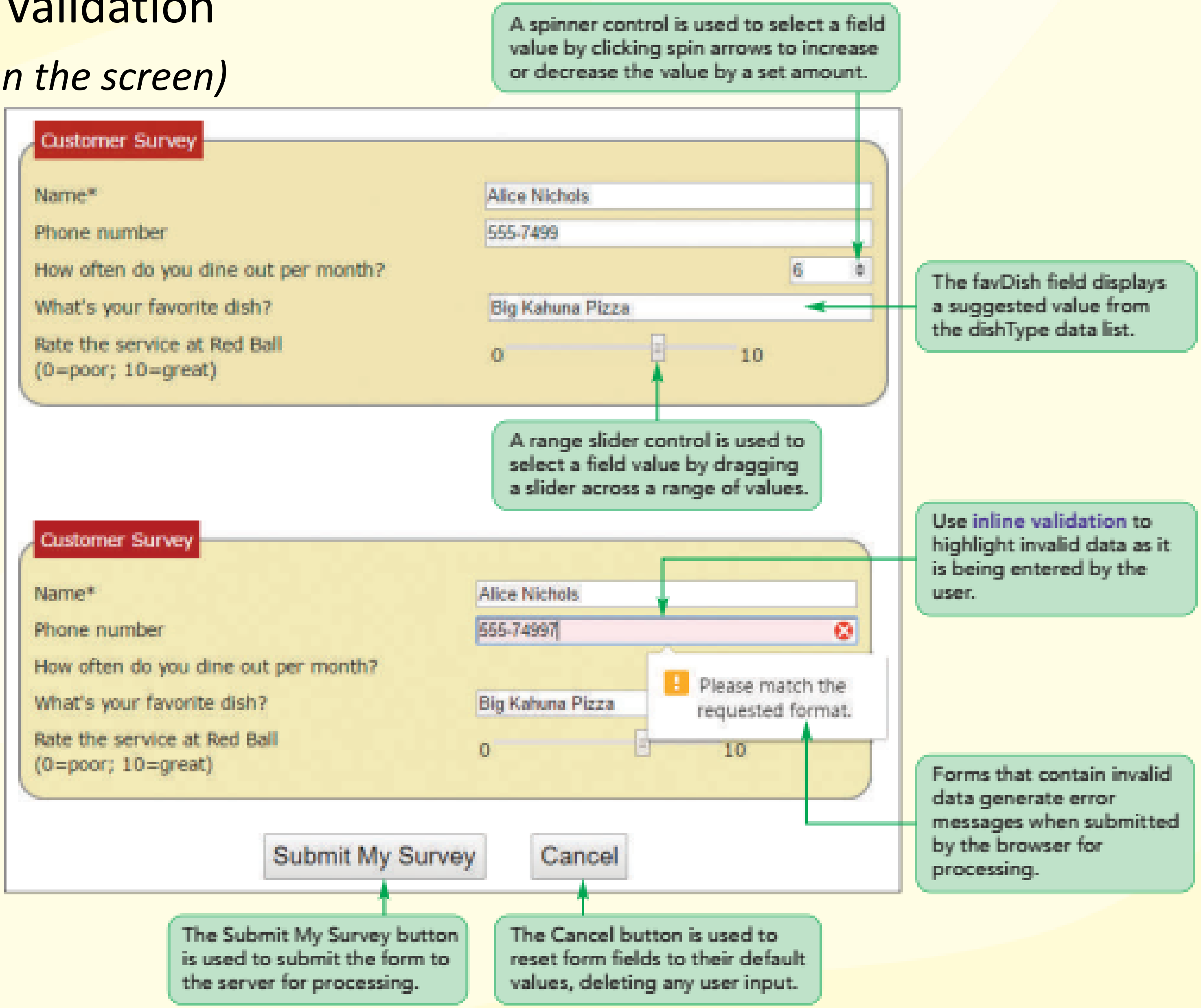
The submit data type creates a button to submit the form for processing.

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)



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

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) 0 10

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.

Customer Survey

Name* Alice Nichols

Phone number 555-74997

Please match the requested format.

How often do you dine out per month?

What's your favorite dish? Big Kahuna Pizza

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

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.

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.

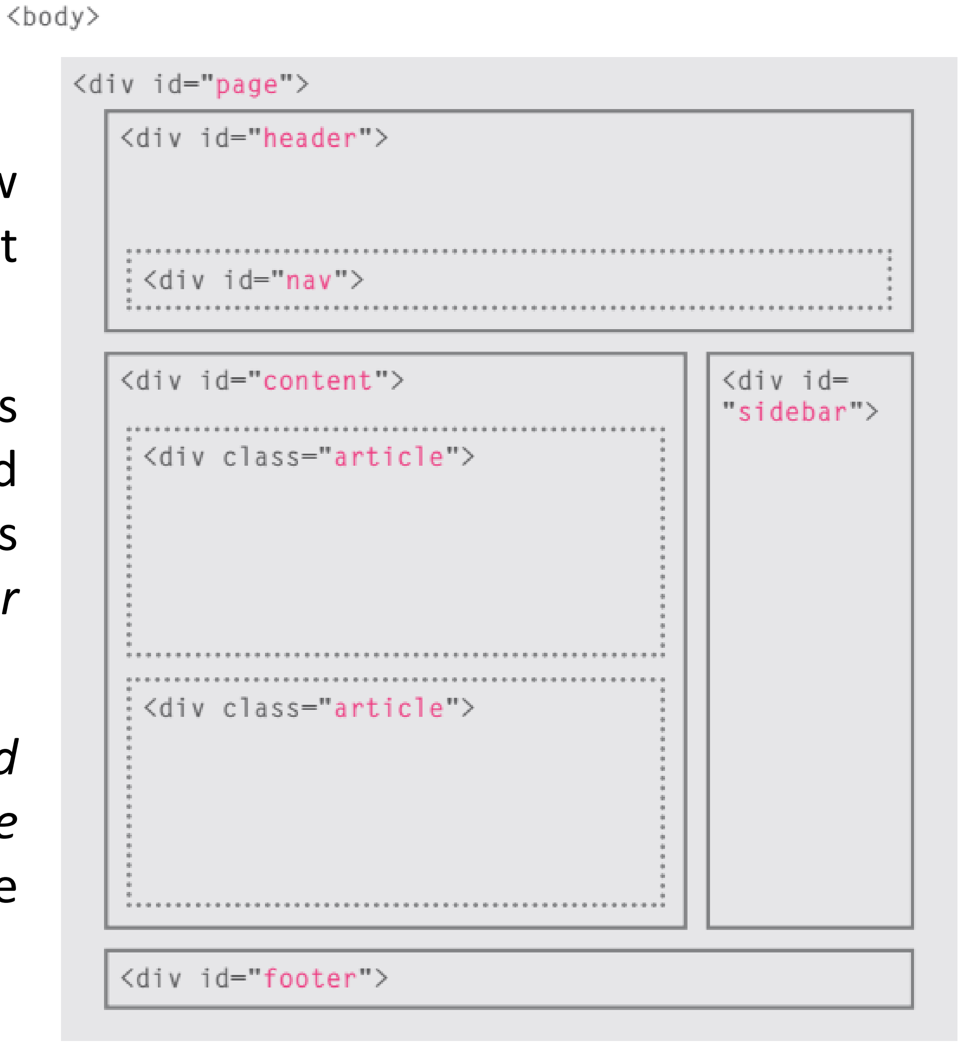
HTML Semantic Elements

Traditional HTML Layout

- Content sectioning elements allow you to organize the document content into logical pieces.

- For a long time, the *<div> element* is used to group together related elements on the web page (such as *a header, an article, footer or sidebar*)

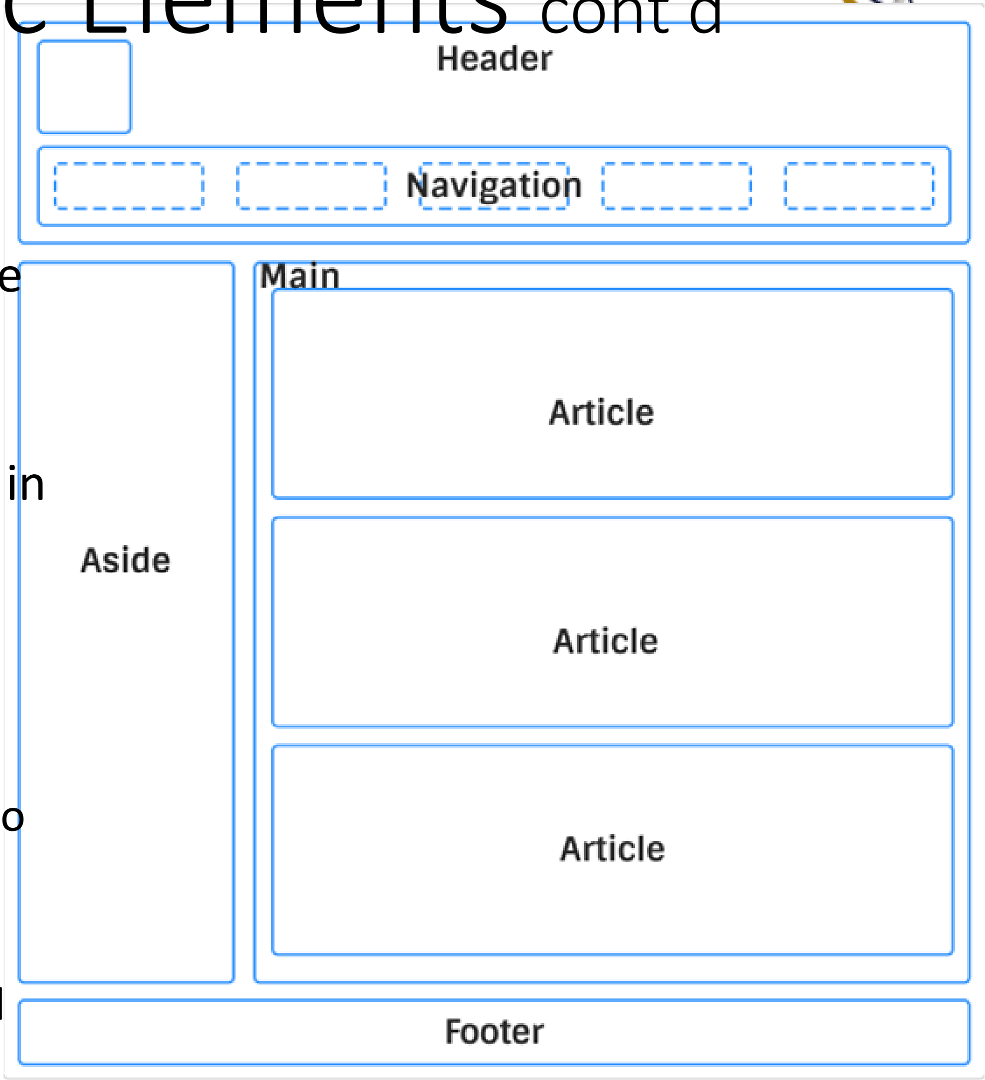
- Web developers are used *class or id attributes* to indicate the *role of the <div> element* in the structure of the page



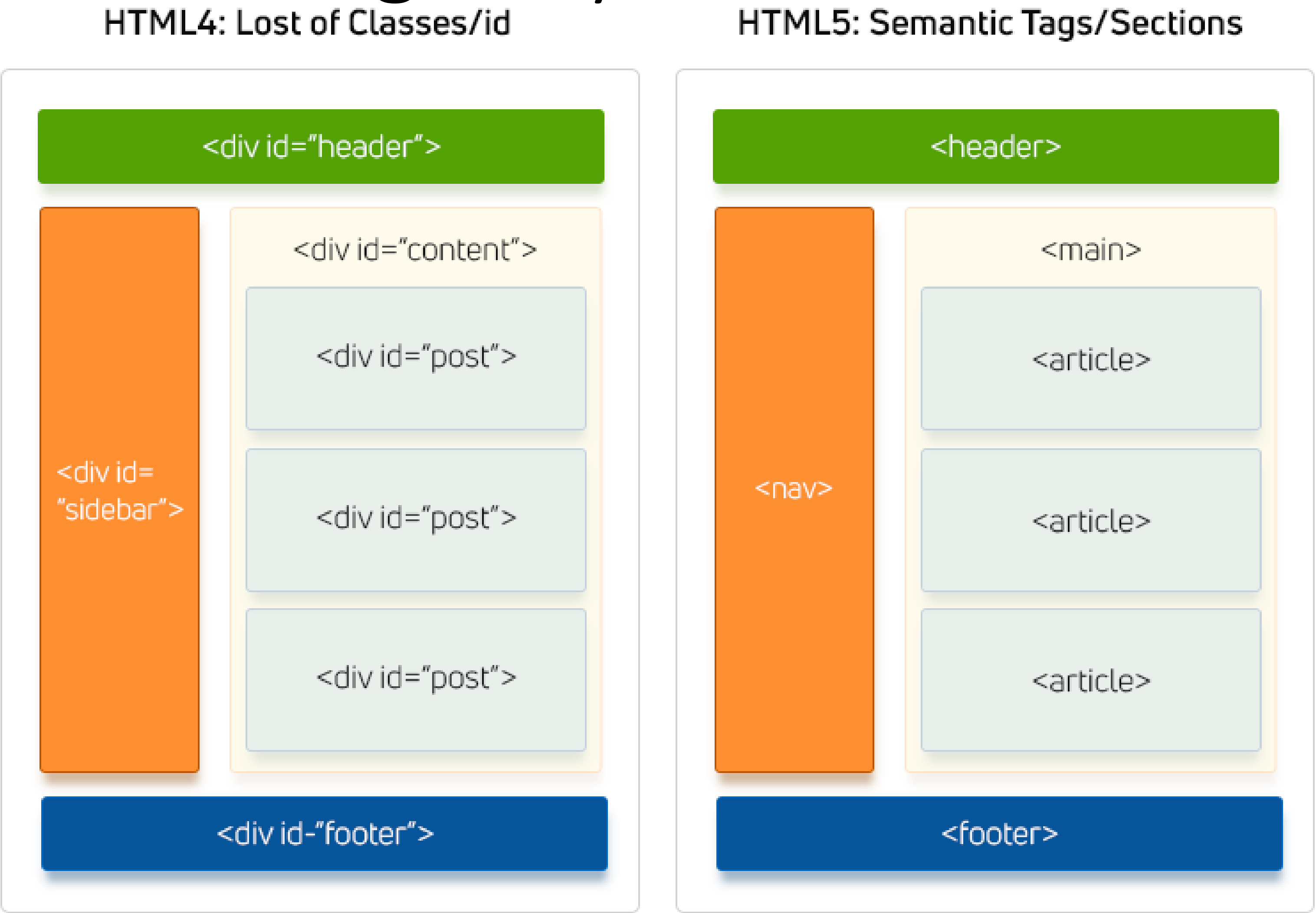
HTML Semantic Elements cont’d

HTML5 Layout Elements

- HTML5 introduces a new set of elements that allow you to divide up the parts of a page
- The names of these elements indicate the kind of content find in them
- HTML5 content sectioning - semantic elements
- **Why semantics?**
    - Clearly describes its meaning to both the browser and the developer
    - Help to structure the code and add readability



Traditional Vs. HTML5 Page Layout Elements



The <header> element

- Specifies a header for a document or a section.
- For introductory content <header> element should be used as container
- Generally it contains one or more heading elements, logo or icons or author's information
- Several <header> elements can be used in one document,
- However, it cannot be placed within a <footer>, <address> or another <header> element.
- It is not sectioning content and therefore does not introduce a new section in the outline
- Intended to usually contain the surrounding section's heading (h1–h6 element)

The <footer> element

- Specifies a footer for a document or a section.
- It is generally used in the last of the section (bottom of the page).
- Typically contains information about the
  - author , contact, and copyright information
  - Sitemap, back to top links,
  - Links to related documents etc.
- Not sectioning content and therefore doesn't introduce a new section.
- To put information like address, e-mail etc. about the author on your web page, all the relevant elements should be included into the footer element.
- For example, enclose information about the author in an <address> element that can be included into the <footer> element.

The <nav> element

- Represents a section of a page whose purpose is to provide navigation links, either within the current document or to other documents
- Common examples of navigation sections are menus, tables of contents, and indexes
- It's not necessary for all links to be contained in a <nav> element
- It is intended only for major block of navigation links; typically the <footer> element often has a list of links.
- A document may have several <nav> elements, for example, one for site navigation and one for intra-page navigation
- User agents, such as screen readers, can use this element to determine whether to omit the initial rendering of navigation-only content

The <main> element

- Represents the dominant content of the <body> of a document.
- It is written within <body> tag
- It is used to accurately describe the primary content of a page.
- Its content should directly related to or expands upon the central topic of a page, or the central functionality of an application.
- Doesn’t contribute to the document's outline
- It should be unique to the document and cannot be nested inside other semantic elements like <article>, <aside>, <header>, <footer> elements
- Content that is repeated across other document sections such as sidebars, navigation links, site logos, and search forms shouldn't be included
- If the search form is the main function of the page, it can be included

The <article> element

- Defines an independent self-contained content in a document, page, application or a site
- The content makes sense on its own and intended to be independently distributable or reusable (e.g., in syndication)
- It generally used on Forum post, a magazine or newspaper article, Blog post, News story, a product card, a user-submitted comment, an interactive widget or gadget, or any other independent item of content
- Each <article> should be identified, typically by including a heading (<h1>-<h6> element) as a child of the <article> element
- When an <article> element is nested, the inner element represents an article related to the outer element

The <section> element

- Represents a generic standalone section of a document, which doesn't have a more specific semantic element to represent it.
- Sections should always have a heading, with very few exceptions.
- When you put your content on a web page, it may contains many chapters, headers, footers, or other sections on a web page that is why HTML <section> tag is used.
- Example: A home page could normally be divided into sections for introduction, content, and contact information
- To reiterate, each <section> should be identified, typically by including a heading (<h1>-<h6> element) as a child of the <section> element, wherever possible

The <aside> element

- Represents a portion of a document whose content is only indirectly related to the document's main content
- According to W3C definition, the <aside> element represents content that forms the main textual flow of a document
- It frequently presented as sidebars or call-out boxes
- Do not use the <aside> element to tag parenthesized text, as this kind of text is considered part of the main flow

?