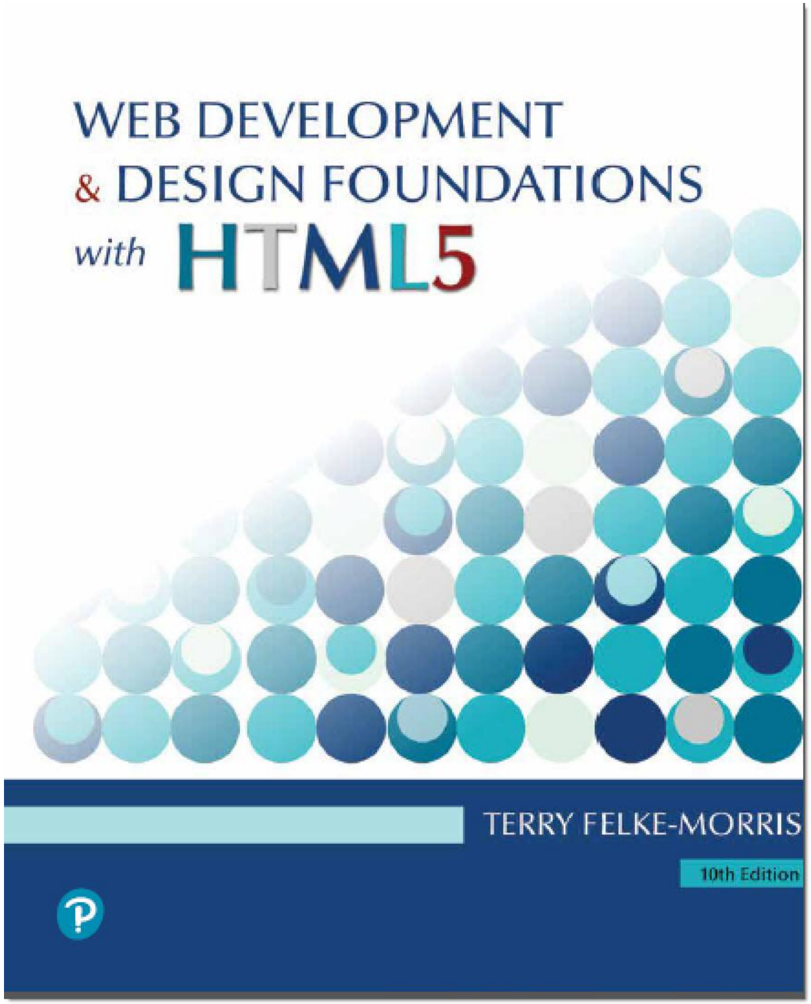
Web Development & Design Foundations with HTML5

Tenth Edition



Chapter 9

Forms

Learning Outcomes (1 of 2)

In this chapter, you will learn how to ...

- Describe common uses of forms on web pages
- Create forms on web pages using the form, input, textarea, and select elements
- Create forms that provide additional accessibility features using the accesskey and tabindex attributes
- Associate form controls and groups using the label, fieldset, and legend elements
- Create custom image buttons and use the button element

Learning Outcomes (2 of 2)

- Use CSS float to style a form
- Use CSS Grid Layout to style a form
- Configure HTML5 form controls including the email, URL, datalist, range, spinner, calendar date-picker, and color controls
- Describe the features and common uses of server-side processing
- Invoke server-side processing to handle form data

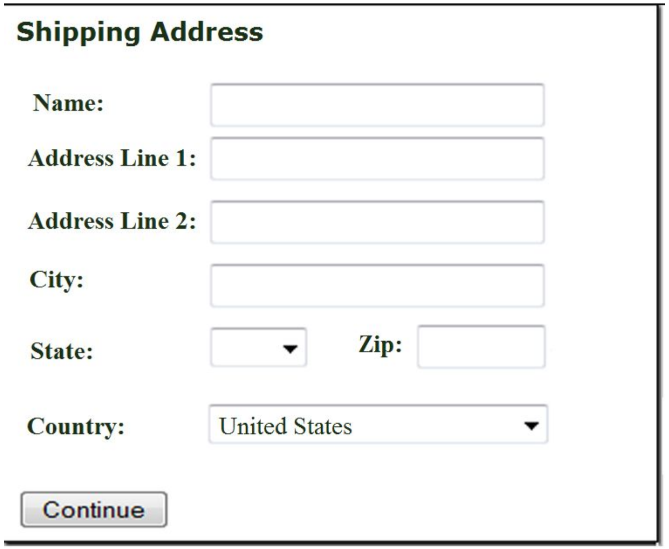
Overview of Forms

Forms are used
all over the Web to

- Accept information
- Provide interactivity

Types of forms:

- Search form, Order form,
- Newsletter sign-up form,
- Survey form, Add to Cart form, and so on…



Shipping Address

Name:

Address Line 1:

Address Line 2:

City:

State: Zip:

Country: United States

This form accepts shipping information

Overview of Forms

Form

- An HTML element that contains and organizes
- form controls such as text boxes, check boxes, and buttons

that can accept information from website visitors.

Two Components of a Form

1. The HTML form - the web page user interface
2. The server-side processing

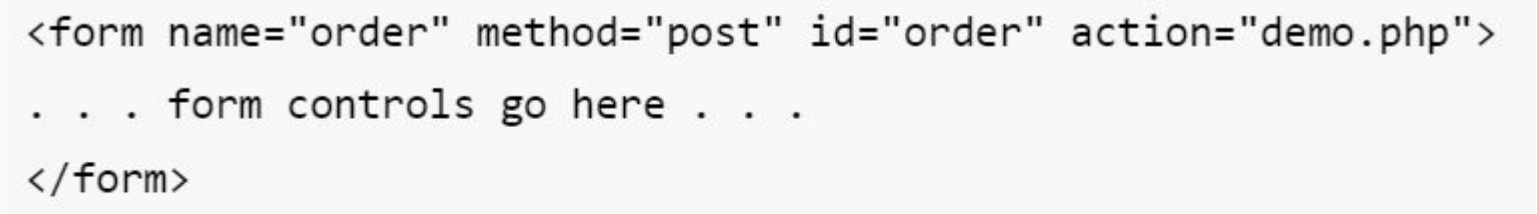
Server-side processing works with the form data and sends e-mail, writes to a text file, updates a database, or performs some other type of processing on the server.

HTML Using Forms

<form>…</form>

- Contains the form elements on a web page
- Container tag
    - Can Have Multiple Forms, but cannot Nest Forms
- Attributes
    - **action="url"**
        - Specifies Location of Resource
            - Browser Executes it When Form Data is Submitted
                - Mailto that Submits Data in an Email
                - Script that Transmits Data to Web Server

- **autocomplete=“on | off”**
- **enctype=“type”**
  - Specifies Type & Format of Submitted Form Data
    - **application/x-www-form-urlencoded**
    - **multipart/form-data**
    - **text/plain**
      - Separated Field / Value Pairs
- **method=“get | post”**
- **name=“type”**
  - Specifies the Name of a Form



```
<form name="order" method="post" id="order" action="demo.php">
. . . form controls go here . . .
</form>
```

input Submit Button

<input>

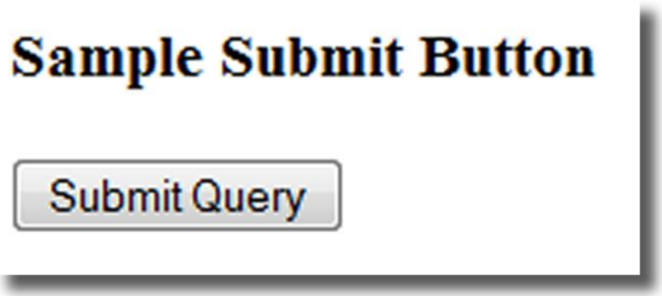
Submits the form information

When clicked:

- Triggers the **action** method on the **<form>** tag
- Sends the form data (the name=value pair for each form element) to the web server.

Attributes:

- type="submit"
- name
- id
- value



Sample Submit Button

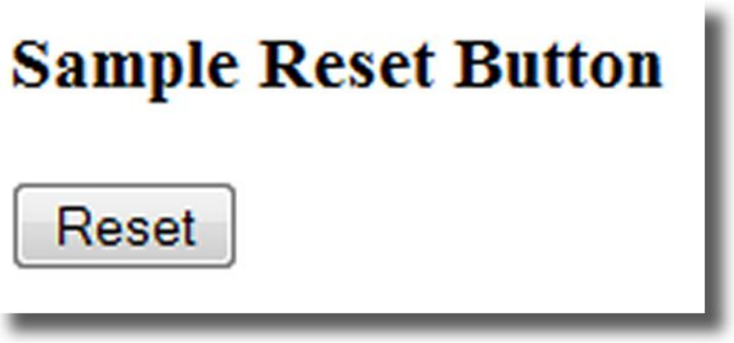
input Reset Button

<input>

Resets the form fields to their initial values

Attributes:

- type="reset"
- name
- id
- value



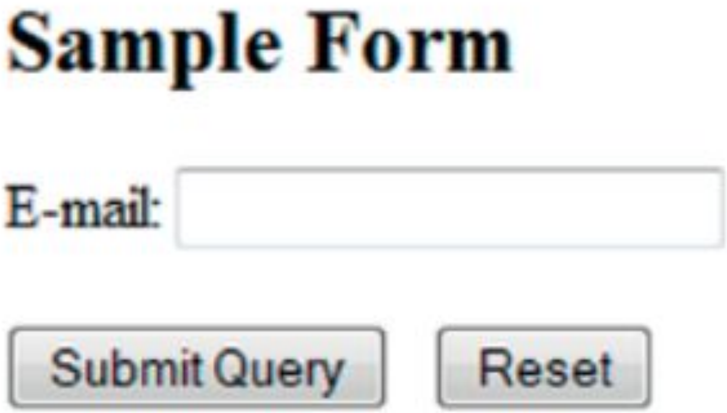
Sample Reset Button

Sample Form HTML

<input>

Configures a variety of form elements including
text boxes, radio buttons, check boxes, and buttons

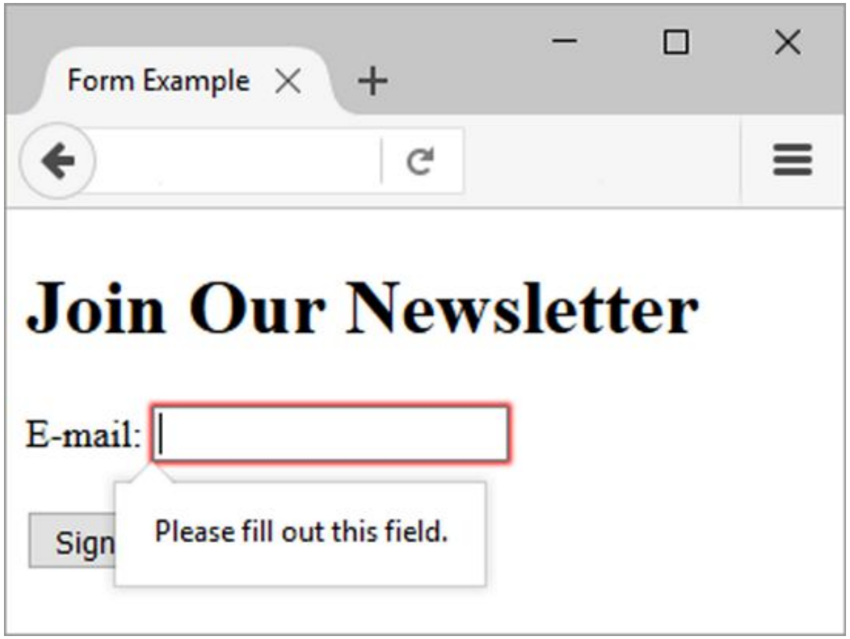
Stand alone tag, defines the type and appearance of form input elements.



```
<form>
    E-mail: <input type="text" name
    ="email"  id="email" required><br>
    <input type="submit“ value =“Submit
    Query”>
    <input type="reset">
</form>
```

Input Text box

<input> Accepts text information



Join Our Newsletter

E-mail:

Please fill out this field.

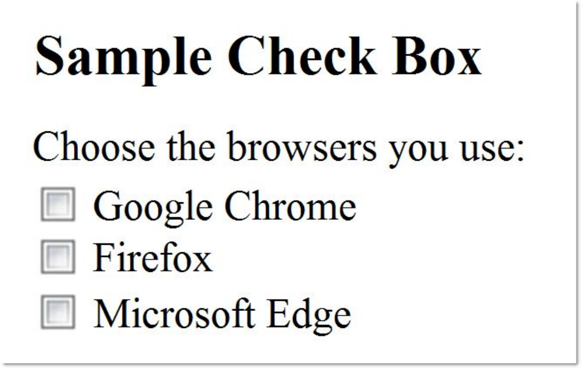
Email: <input type=“text” name = “email” id=“email” required>

input Check box

<input>

Allows the user to select one or more of a group of predetermined items

- Attributes:
  - type="checkbox"
  - name
  - id
  - checked
  - value



Sample Check Box

Choose the browsers you use:

□ Google Chrome
□ Firefox
□ Microsoft Edge

Figure 9.6 Sample check box

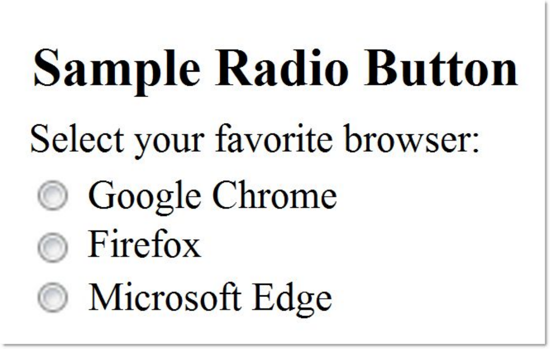
input Radio Button

<input>

Allows the user to select exactly one from a group of predetermined items

Each radio button in a group is given the same name and a unique value

- Attributes:
  - type="radio"
  - name
  - id
  - checked
  - value



Sample Radio Button

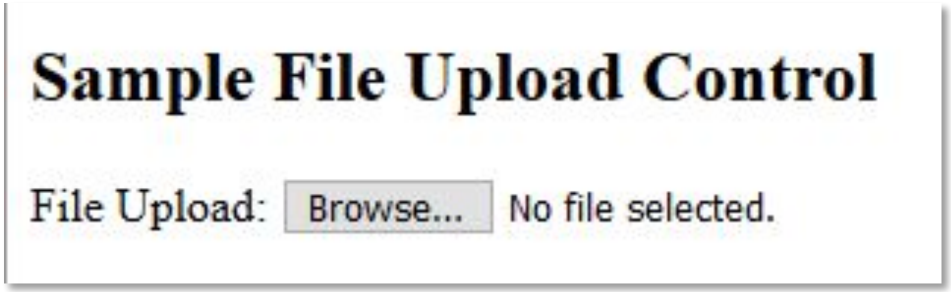
Select your favorite browser:

○ Google Chrome
○ Firefox
○ Microsoft Edge

Use radio buttons when only one choice is an appropriate response

input File Upload Control

- Attributes:
  - type="file"
  - name
  - id
  - accepts
  - capture
  - Multiple



Sample File Upload Control

File Upload: No file selected.

File Upload: <input type= “file" name= “photo“ id= “pword“>

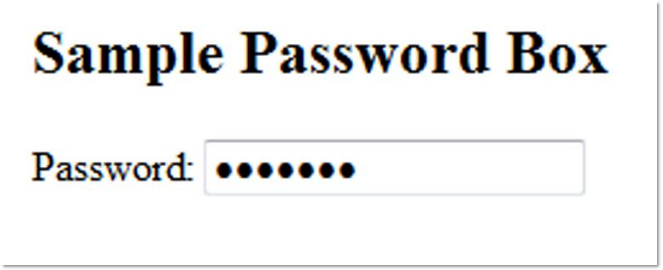
input Password box

<input>

```
Password: <input type= "password" name= “pword“ id= “pword“>
```

Accepts text information that needs to be hidden as it is entered

- Attributes:
  - type="password"
  - name
  - id
  - size
  - maxlength
  - value



Sample Password Box

Password: •••••••

The characters secret9 were typed, but the browser does not display them. (Note: Your browser may use a different symbol, such as a stylized circle,to hide the characters.)

textarea Scrolling Text Box

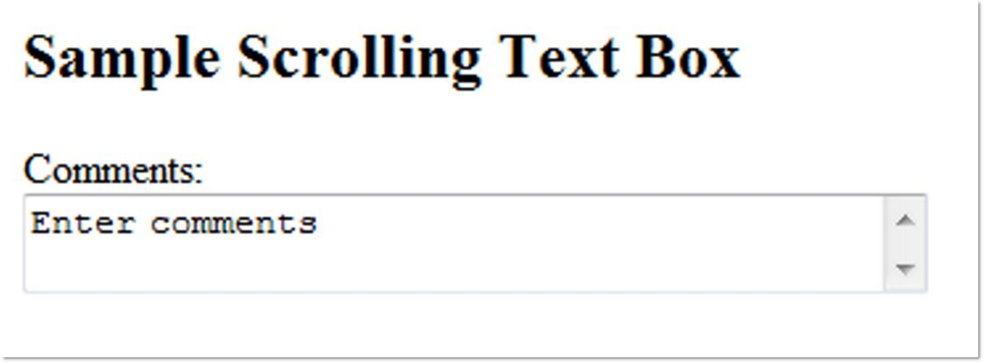
<textarea> </textarea>

Configures a scrolling text box

Container tag

Attributes:

- name
- id
- cols
- rows



Sample Scrolling Text Box

Comments:

Enter comments

Figure 9.9 Scrolling text box

Select List

<select></select>

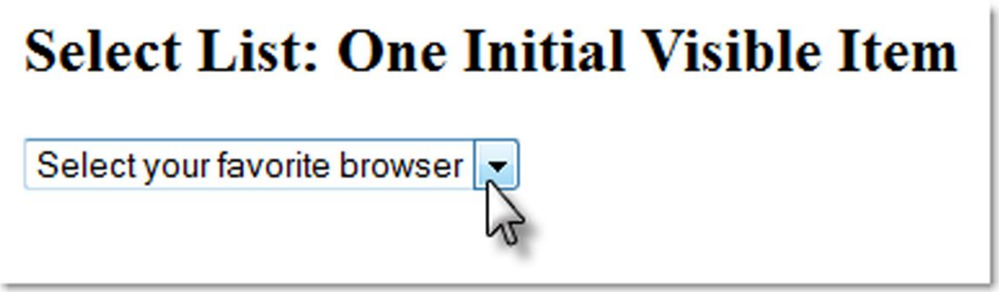
Configures a select list (along with option elements)

Also known as: Select Box, Drop-Down List, Drop-Down Box, and Option Box.

Allows the user to select one or more items from a list of predetermined choices.

Select List

- Attributes:
  - name
  - id
  - size
  - multiple



Select List: One Initial Visible Item

A select list with size set to 1 functions as a drop-down box when the arrow is clicked

```
<select size="1" name="favbrowser" id="favbrowser">
  <option>Select your favorite browser</option>
  <option value="Edge">Edge</option>
  <option value="Firefox">Firefox</option>
  <option value="Chrome">Chrome</option>
</select>
```

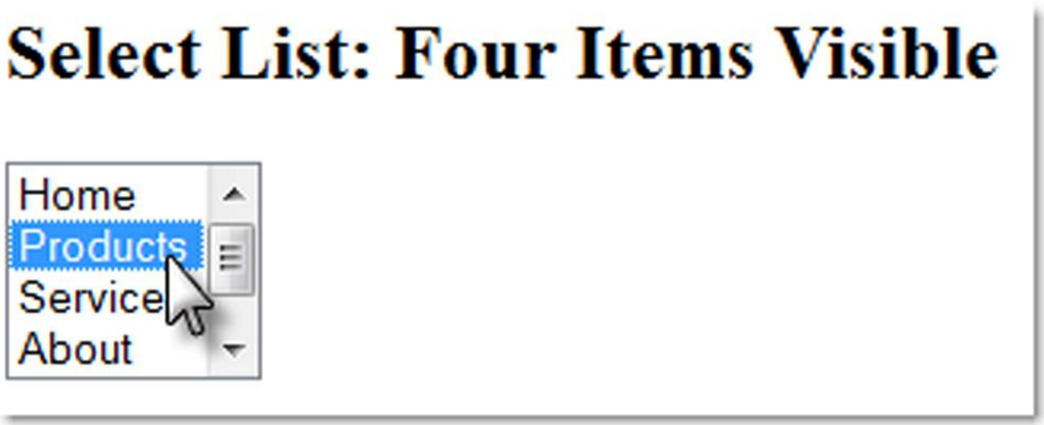
Options in a Select List

<option></option>

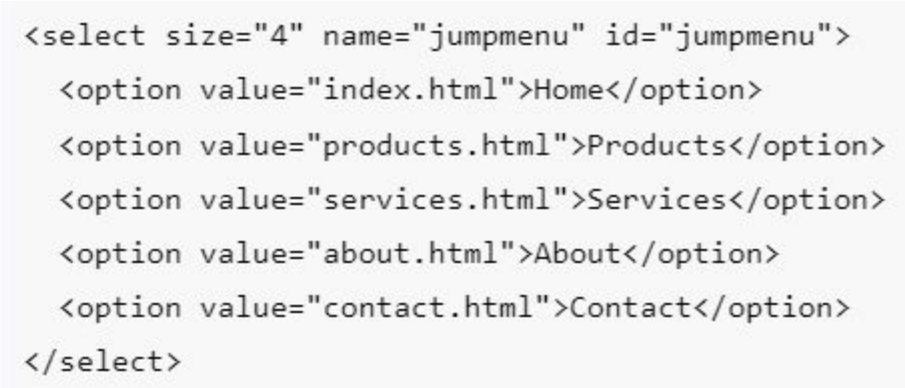
Configures the options in a Select List

Attributes:

- value
- selected



Select List: Four Items Visible



```
<select size="4" name="jumpmenu" id="jumpmenu">
  <option value="index.html">Home</option>
  <option value="products.html">Products</option>
  <option value="services.html">Services</option>
  <option value="about.html">About</option>
  <option value="contact.html">Contact</option>
</select>
```

Because there are more than four choices, the browser displays a scroll bar

Checkpoint (1 of 4)

1. You are designing a web site for a client who sells items in a retail store.

They want to create a customer list for e-mail marketing purposes. Your client sells to consumers and needs a form that accepts their customer's name and e-mail address.

Would you recommend using two input boxes
(one for the name and one for the e-mail)
or three input boxes
(for the first name, last name, and email address)?
Explain your answer.

Checkpoint (2 of 4)

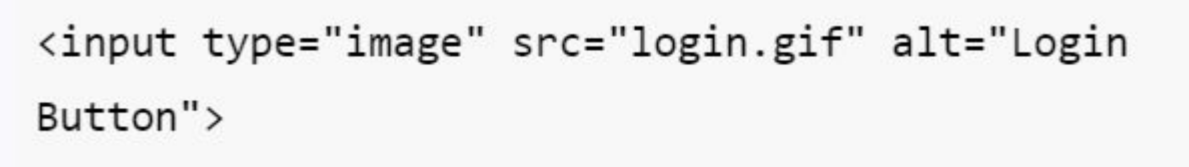
2. You are designing a survey form for a client. One of the questions has 10 possible responses.
   Only one response can be selected per question.

What type of form control would you use to configure this question on the web page?

input Image Button

<input>

Submits the form

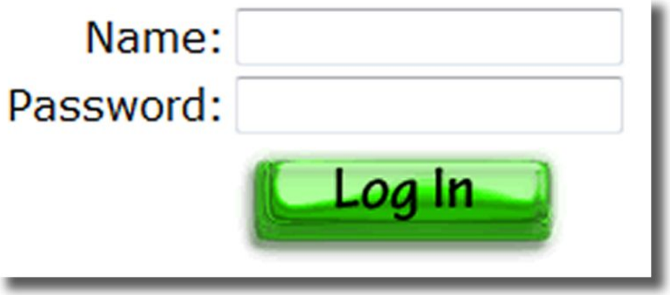


```
<input type="image" src="login.gif" alt="Login
Button">
```

When clicked:

- Triggers the **action** method on the form tag
- Sends the form data (the name=value pair for each form element) to the web server.

- Attributes:
  - type="image"
  - name
  - id
  - src



The web page visitor will select the image button to submit the form

Button Element (1 of 2)

<button></button>

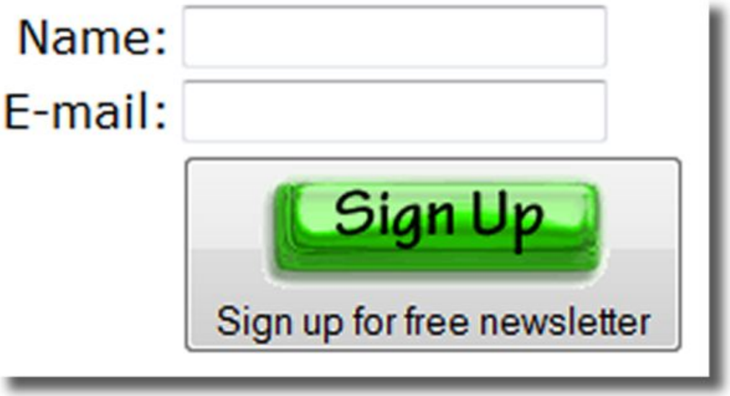
A container tag

When clicked, its function depends on the value of the type attribute.

Can contain a combination of text, images, and media

Button Element (2 of 2)

```
<button type="submit">
<img src="signup.gif" width="80" height="28" alt="Sign up for free
newsletter"><br>Sign up for free newsletter
</button>
```



The button element configured as a submit button

Accessibility & Forms

Label Element

Fieldset Element

Legend Element

Tabindex Attribute

Accesskey Attribute

Label Element

<label></label>

Associates a text label with a form control

Two Different Formats:

```
<label>Email: <input type="text" name=“email"  id
="email"></label>
```

Or

```
<label for="email">Email: </label>
<input type="text" name="CustEmail" id= "email">
```

Fieldset and Legend Elements (1 of 3)

The Fieldset Element

- Container tag
- Creates a visual group of form elements on a web page

The Legend Element

- Container tag
- Creates a text label within the fieldset

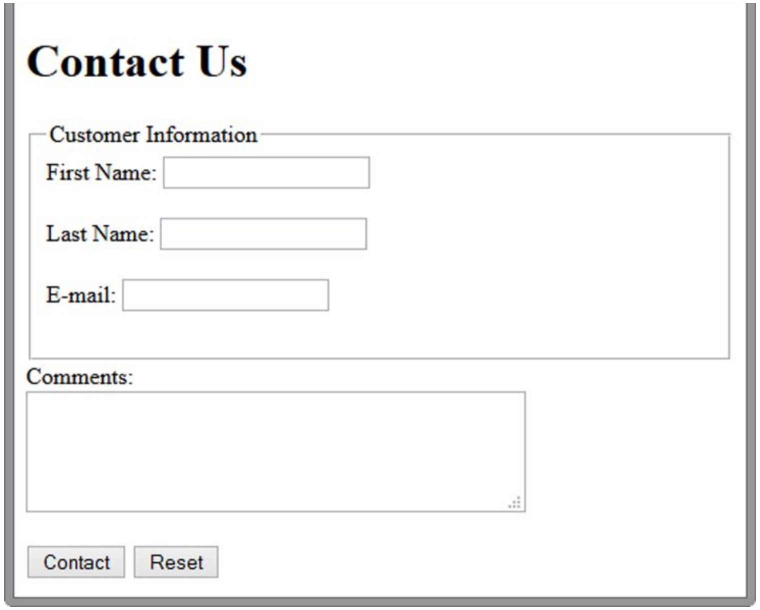
Fieldset and Legend Elements (2 of 3)

```
<fieldset><legend>Customer Information</legend>

   <label>Name:
  <input type="text" name="Name" id="Name“></label>
  <br><br>

   <label>Email:
  <input type="text" name="Email" id="Email"></label>

 </fieldset>
```



tabindex attribute

Attribute that can be used on form controls and anchor tags

Modifies the default tab order

Assign a numeric value

```
<input type="text" name="CustEmail"
id="CustEmail" tabindex="1">
```

Checkpoint (3 of 4)

1. Describe the purpose of the fieldset and legend elements.
2. Describe the purpose of the tabindex and how it supports accessibility.
3. When designing a form, should you use the standard submit button, an image button, or a button tag? Are these different in the way that they provide for accessibility? Explain your answer.
4. Taking usability into consideration, the preferred form layout structure is:

Using CSS Float to Style a Form (1 of 2)

```
form { background-color: #EAEAEA;
    font-family: Arial, sans-serif;
          width: 350px;
          padding: 10px;}
label { float: left;
          clear: left;
    display: block;
          width: 100px;
```

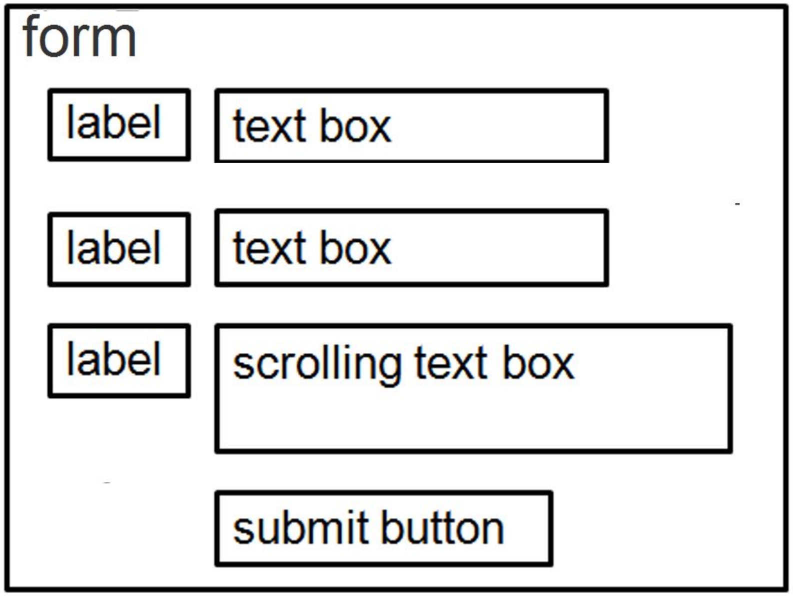


Figure 9.18 Wireframe for a form

Using CSS Float to Style a Form (2 of 2)

```
    text-align: right;
    padding-right: 10px;
    margin-top: 10px; }
input, textarea { margin-top:
10px;
    display: block;}
input[type="submit"] {
margin-left: 110px; }
```

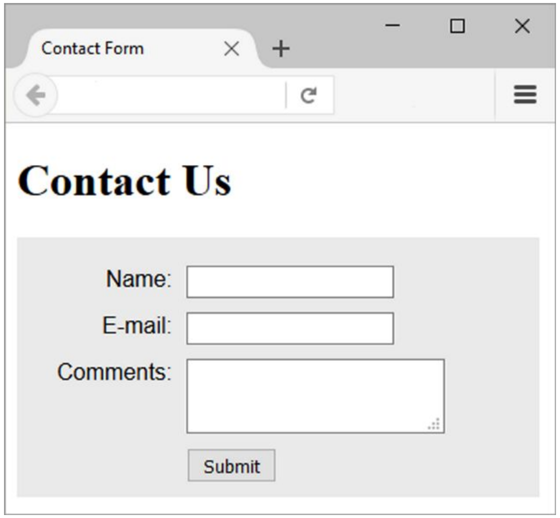


Figure 9.19 This form is configured with CSS

Using CSS Grid Layout to Style a Form

(1 of 3)

```
form { display: grid;
    grid-template-rows: auto;
    grid-template-columns: 6em 1fr;
    grid-gap: 1em; gap: 1em;
    background-color: #EAEAEA;
    font-family: Arial, sans-serif;
    min-width: 25em;
```

Using CSS Grid Layout to Style a Form

(2 of 3)

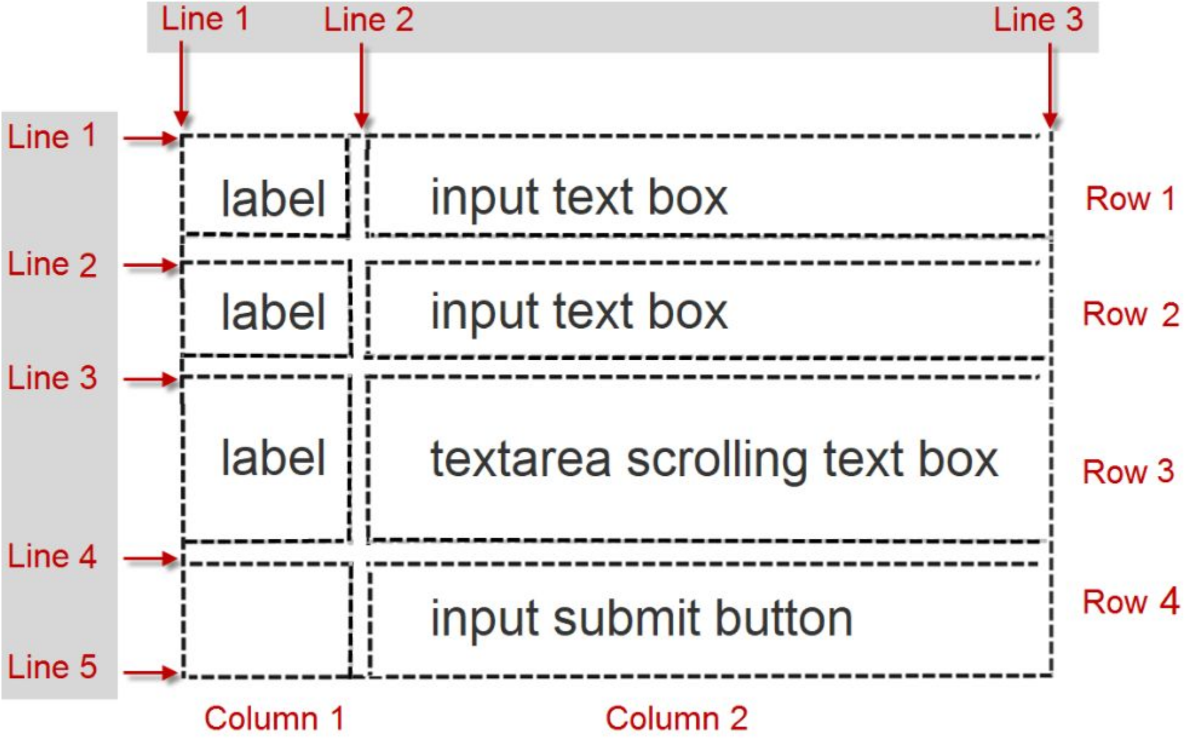
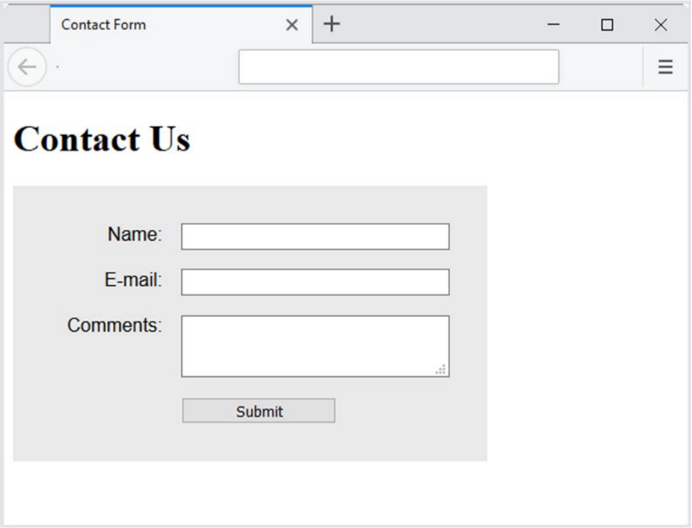


Figure 9.20 The grid for the form

Using CSS Grid Layout to Style a Form (3 of 3)

```
width: 60%;
padding: 2em; }
input[type="submit"] {
grid-column: 2 / 3;
width: 10em; }
```



Contact Us

Name:

E-mail:

Comments:

Figure 9.21 The form is styled with CSS grid layout

Server-Side Processing

- Your web browser requests web pages and their related files from a web server.
- The web server locates the files and sends them to your web browser.
- The web browser then renders the returned files and displays the requested web pages for you to use.

Server-Side Scripting

- One of many technologies in which a server-side script is embedded within a Web page document saved with a file extension such as:
  - .php (PHP)
  - .cfm (Adobe ColdFusion)
  - .jsp (Sun JavaServer Pages)
  - .aspx (ASP.Net).
- Uses direct execution — the script is run either by the web server itself or by an extension module to the web server.

Common Uses of Server-Side Scripting

Search a database

Place an order at an online store

Send a web page to a friend

Subscribe to a newsletter

Sending information to a Server-side Script

```
<form method="post"
action="https://webdevbasics.net/scripts/demo.php ">
```

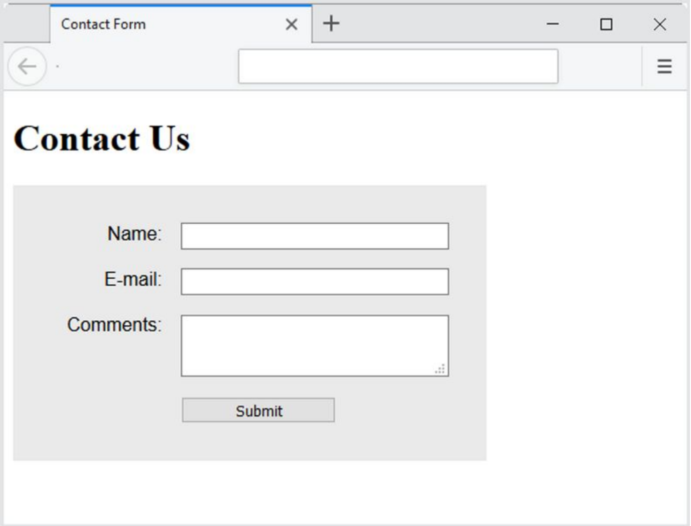
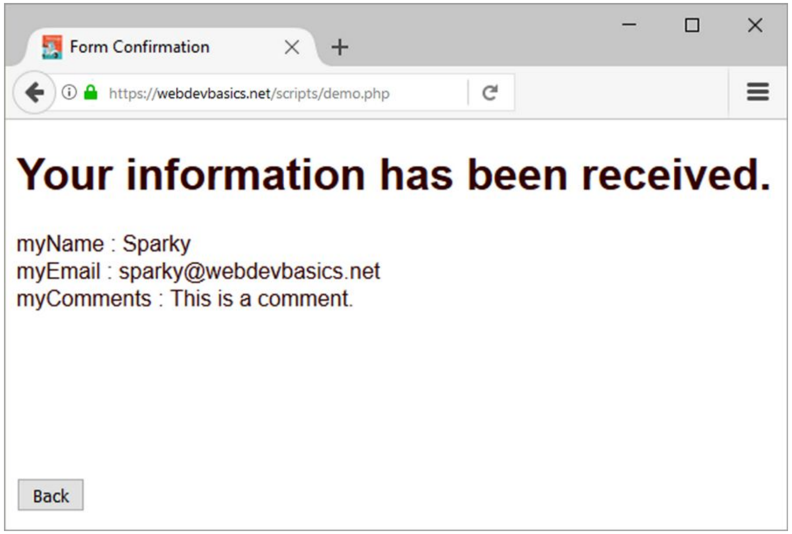


Figure 9.21 The form is styled with CSS grid layout



Your information has been received.

```
myName : Sparky
myEmail : sparky@webdevbasics.net
myComments : This is a comment.
```

Figure 9.22 The server-side script has created this web page in response to the form

Checkpoint (4 of 4)

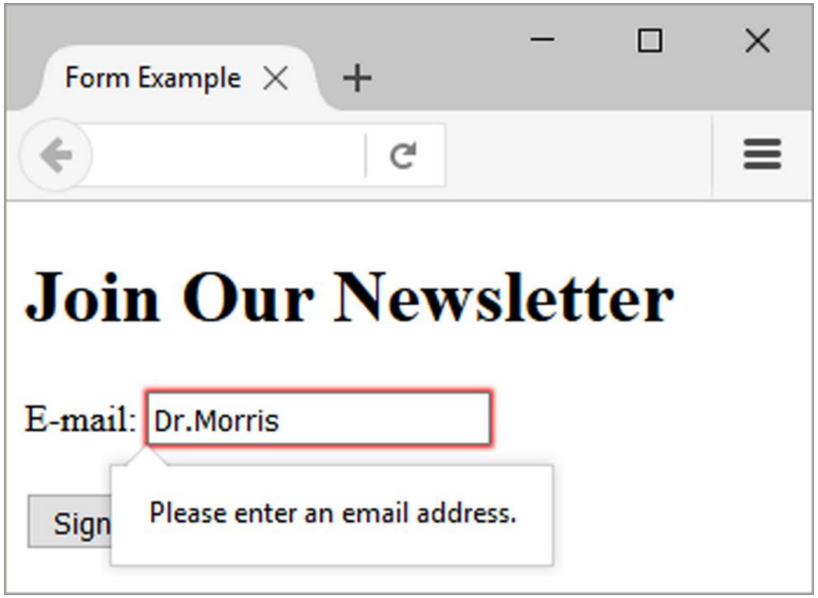
1. Describe server-side processing.
2. Describe why communication is needed between the developer of a server-side script and the web page designer.

HTML5: Email Text Box

<input>

Accepts text information in e-mail address format

- Common Attributes:
  - type="email"
  - name
  - id
  - size
  - maxlength
  - value
  - placeholder
  - required



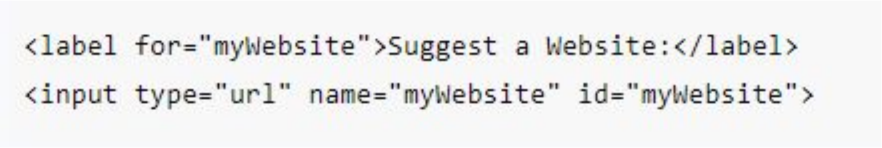
Join Our Newsletter

E-mail: Dr.Morris

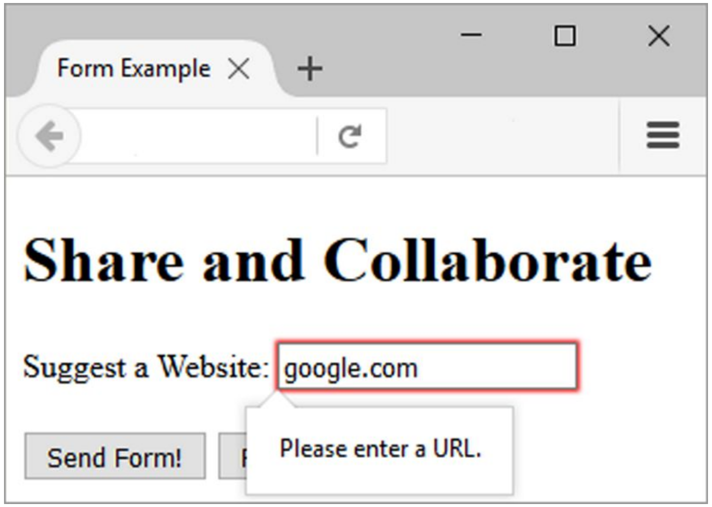
Please enter an email address.

Figure 9.23 The browser displays an error message

HTML5: URL input



```
<label for="myWebsite">Suggest a Website:</label>
<input type="url" name="myWebsite" id="myWebsite">
```



Share and Collaborate

Suggest a Website: google.com

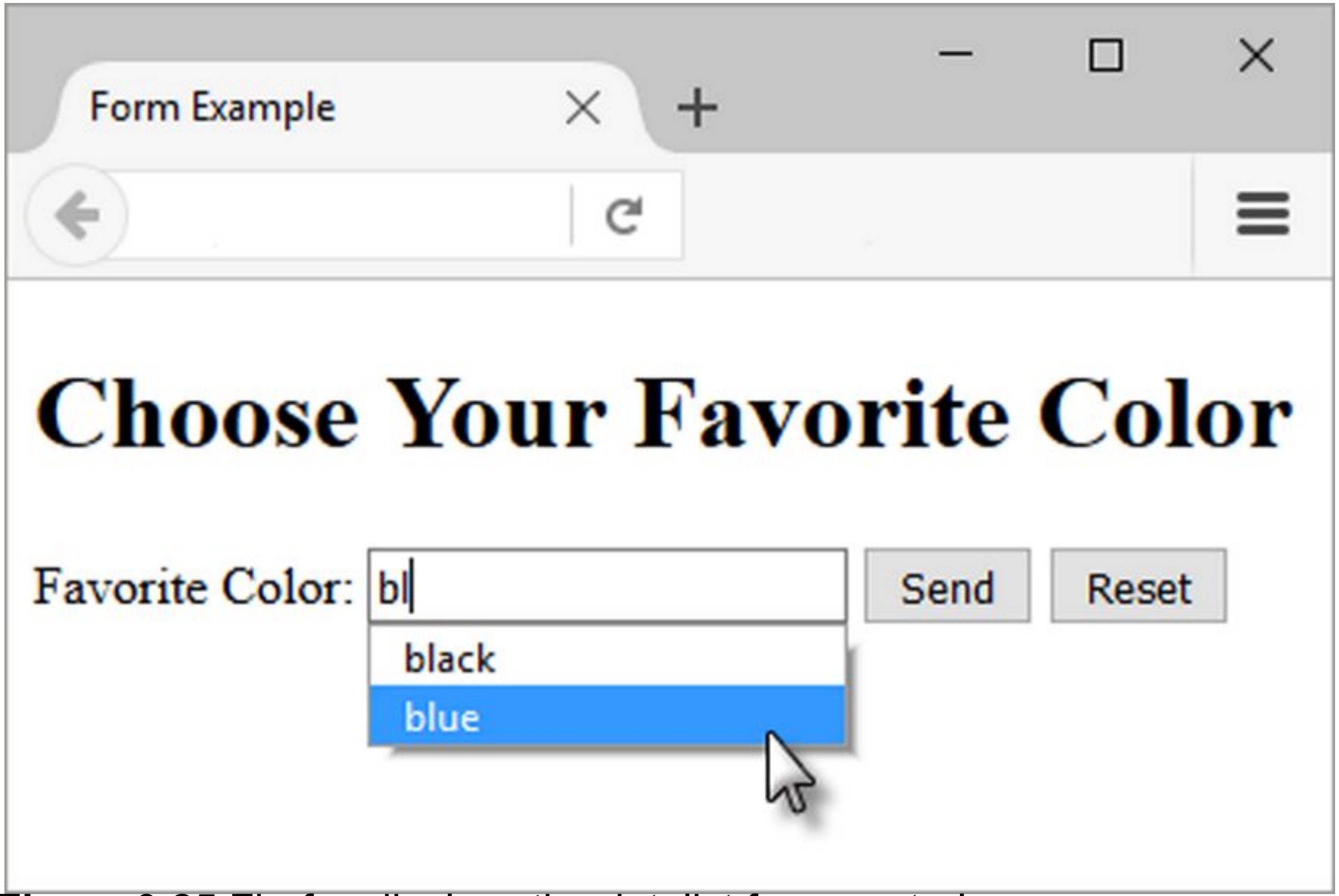
Please enter a URL.

The browser displays an error message

HTML5: Datalist Control (1 of 2)

```
<label for="color">Favorite Color:</label>
<input type="text" name="color" id="color" list="colors">
<datalist id="colors">
    <option value="red" label="Red">
    <option value="green" label="Green">
    <option value="blue" label="Blue">
    <option value="yellow" label=“Yellow">
    <option value="pink" label="Pink">
     <option value="black" label="Black">
</datalist>
```

HTML5: Datalist Control (2 of 2)



Choose Your Favorite Color

Favorite Color: bl

black

blue

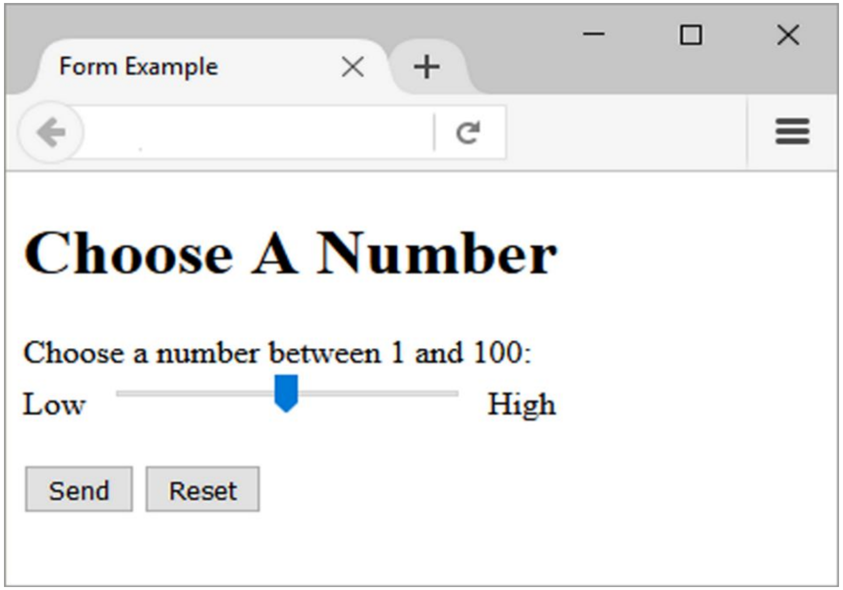
Figure 9.25 Firefox displays the datalist form control

HTML5: Slider Control

```
<label for="myChoice">

Choose a number between 1 and 100:</label><br>

Low <input type="range"
        name="myChoice"
        id="myChoice"
        min="1"
        max="100"> High
```



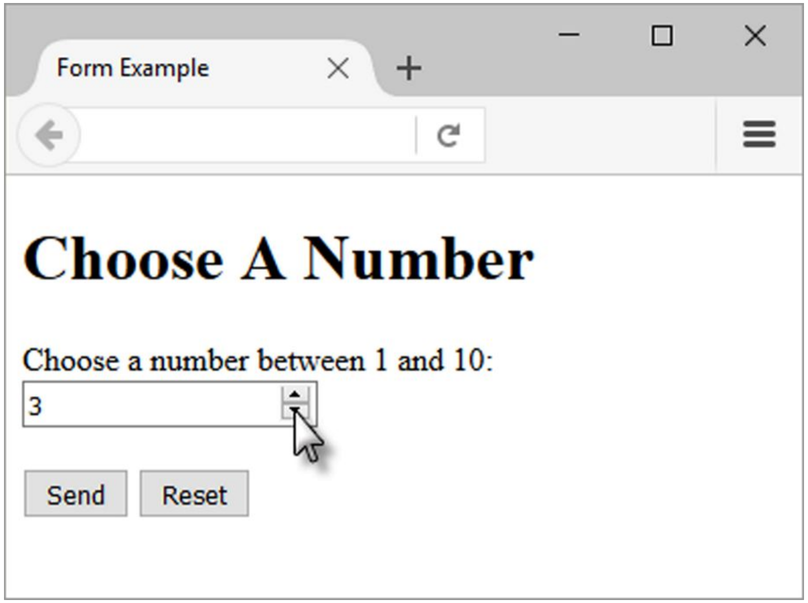
Choose A Number

Choose a number between 1 and 100:

Figure 9.27 The Firefox browser displays the range form control.

HTML5: Spinner Control

```
<label for="myChoice">Choose a number between 1
and 10:</label>
<input type="number"
        name="myChoice"
        id="myChoice“
        min="1" max="10">
```



Choose A Number

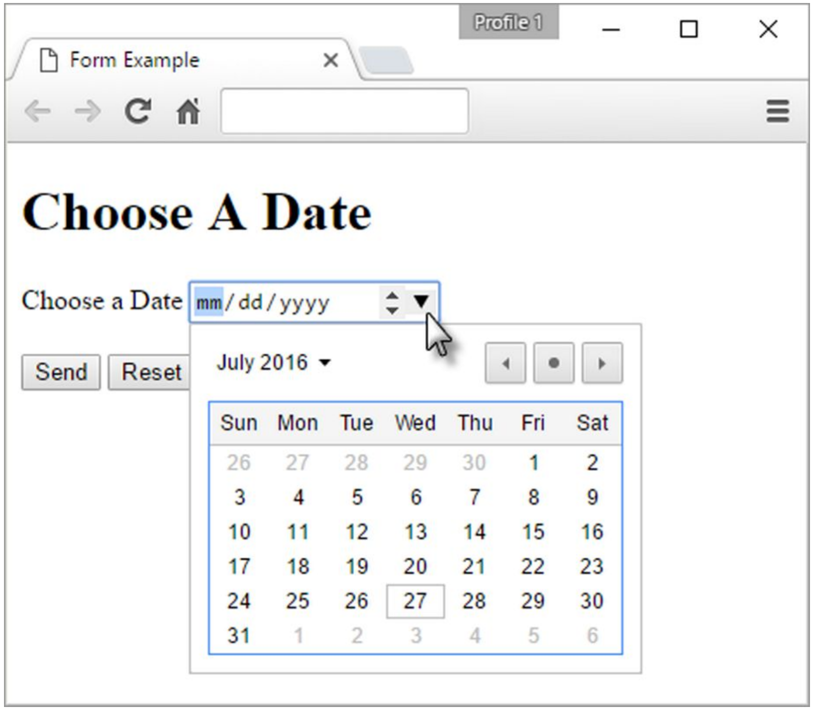
Choose a number between 1 and 10:

3

Figure 9.28 A spinner control displayed in the Firefox browser

HTML5: Calendar Date-Picker Control

```
<label for="myDate">Choose a Date</label>
<input type="date" name="myDate" id="myDate">
```



Choose A Date

Choose a Date mm / dd / yyyy

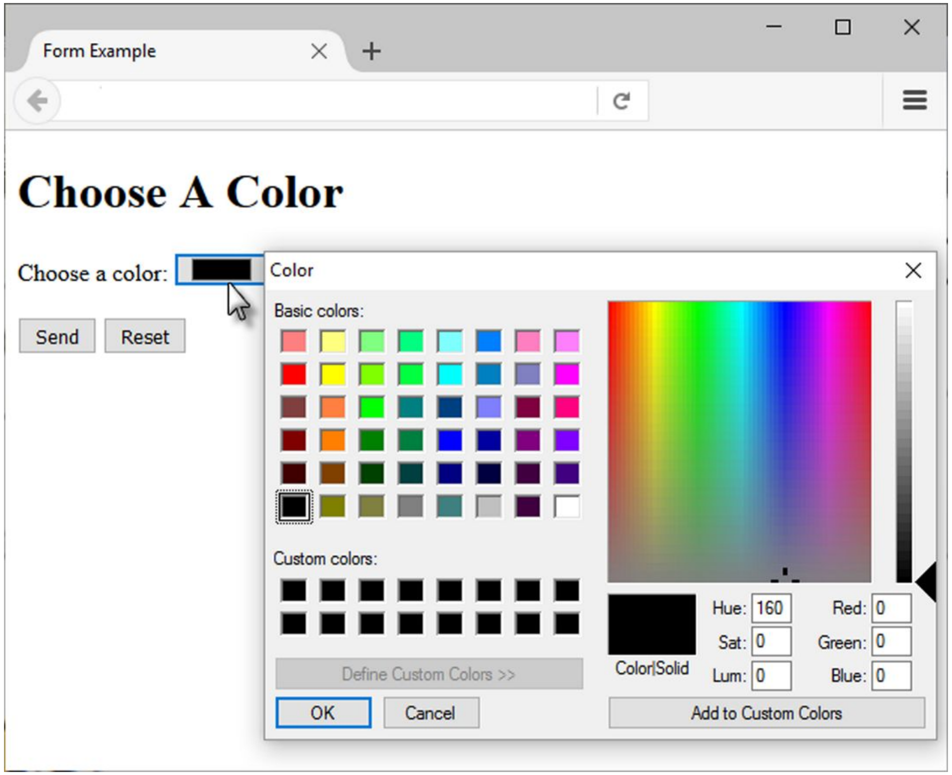
July 2016

Sun	Mon	Tue	Wed	Thu	Fri	Sat
26	27	28	29	30	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	5	6

A date form control displayed in the Firefox browser

HTML5: Color Well Control

```
<label for="myColor">Choose a color:</label>
<input type="color" name="myColor" id="myColor">
```



Choose A Color

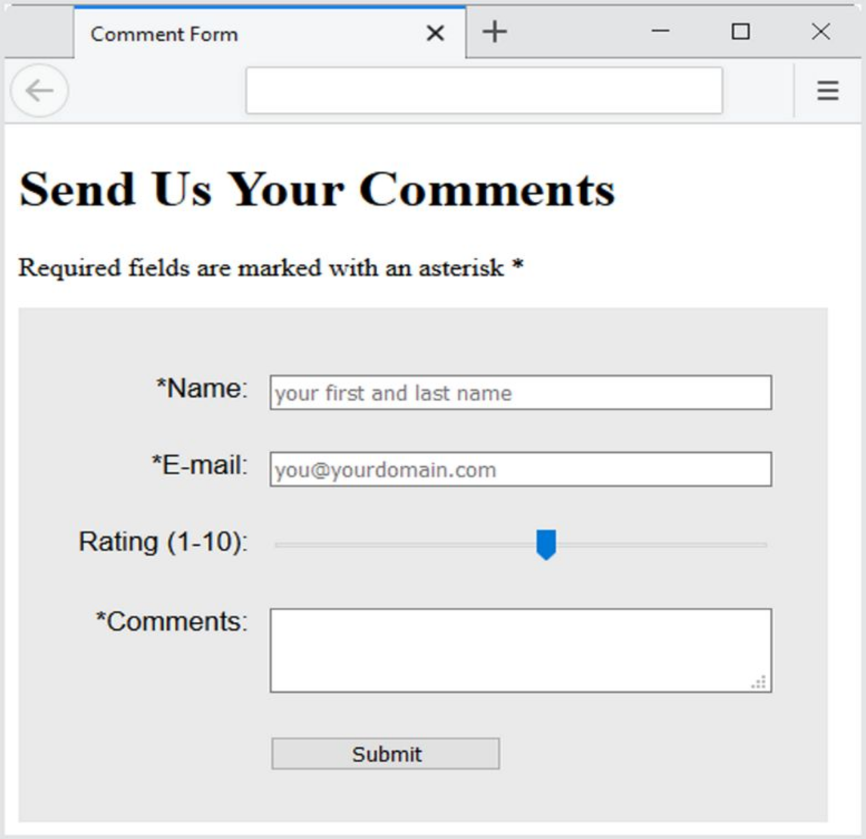
Choose a color:

Hue: 160
Sat: 0
Lum: 0
Red: 0
Green: 0
Blue: 0

The Firefox browser supports the color-well form control

Practice with an HTML5 Form

The form display and functioning varies with browser support.



Send Us Your Comments

Required fields are marked with an asterisk *

*Name:	your first and last name
*E-mail:	you@yourdomain.com
Rating (1-10):	
*Comments:	

The form displayed in the Firefox browser

Summary

This chapter introduced the use of forms on web pages.

You learned about how to configure form elements and provide for accessibility.

You also learned how to configure a form to access server-side processing.

In addition, you learned about new HTML5 form controls.