Chapter 10

How to work with forms and data validation

Objectives

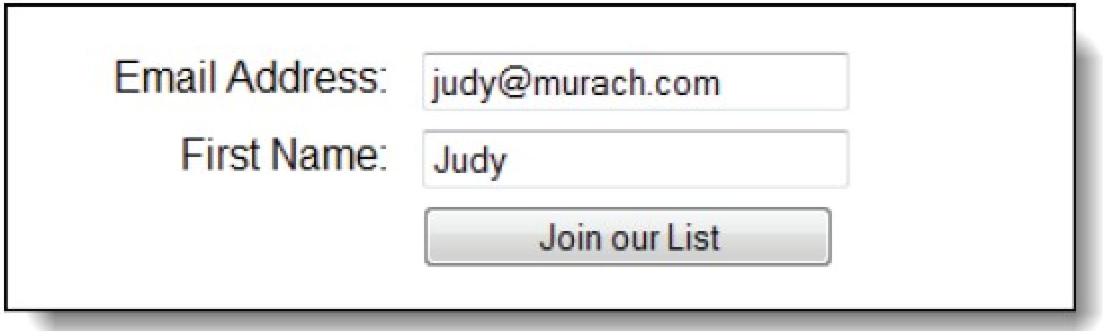
- Introduction to forms and controls
- How to use jQuery to work with forms
- A Validation application that uses JavaScript

Introduction to forms and controls

Introduction to forms

- A ***forms*** contain one or more ***controls*** like text boxes, radio buttons, list and check boxes that can receive data.

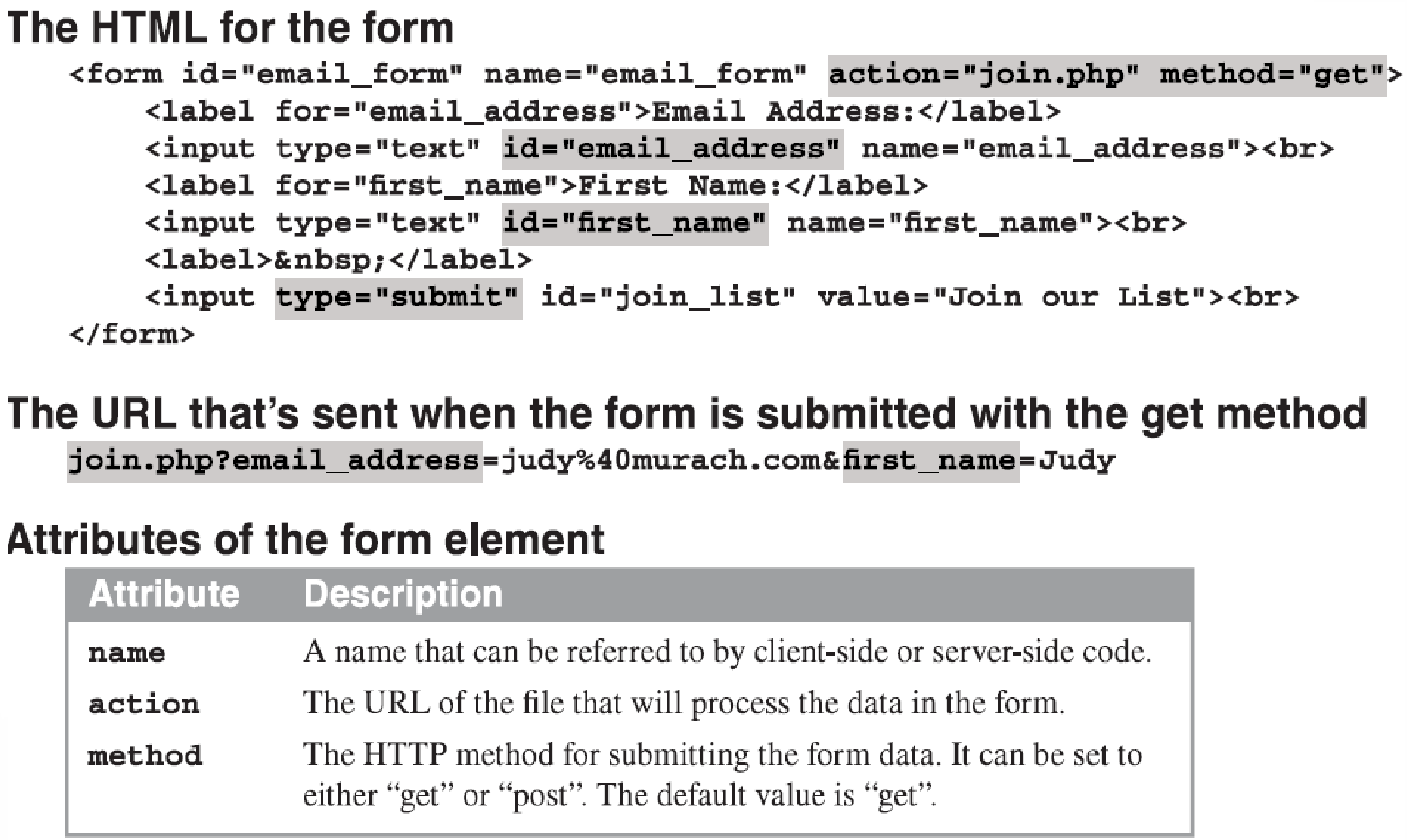
A form in a web browser



Email Address: judy@murach.com

First Name: Judy

Introduction to forms (cont.)



The HTML for the form

```
<form id="email_form" name="email_form" action="join.php" method="get">
    <label for="email_address">Email Address:</label>
    <input type="text" id="email_address" name="email_address"><br>
    <label for="first_name">First Name:</label>
    <input type="text" id="first_name" name="first_name"><br>
    <label>&nbsp;</label>
    <input type="submit" id="join_list" value="Join our List"><br>
</form>
```

The URL that's sent when the form is submitted with the get method

```
join.php?email_address=judy%40murach.com&first_name=Judy
```

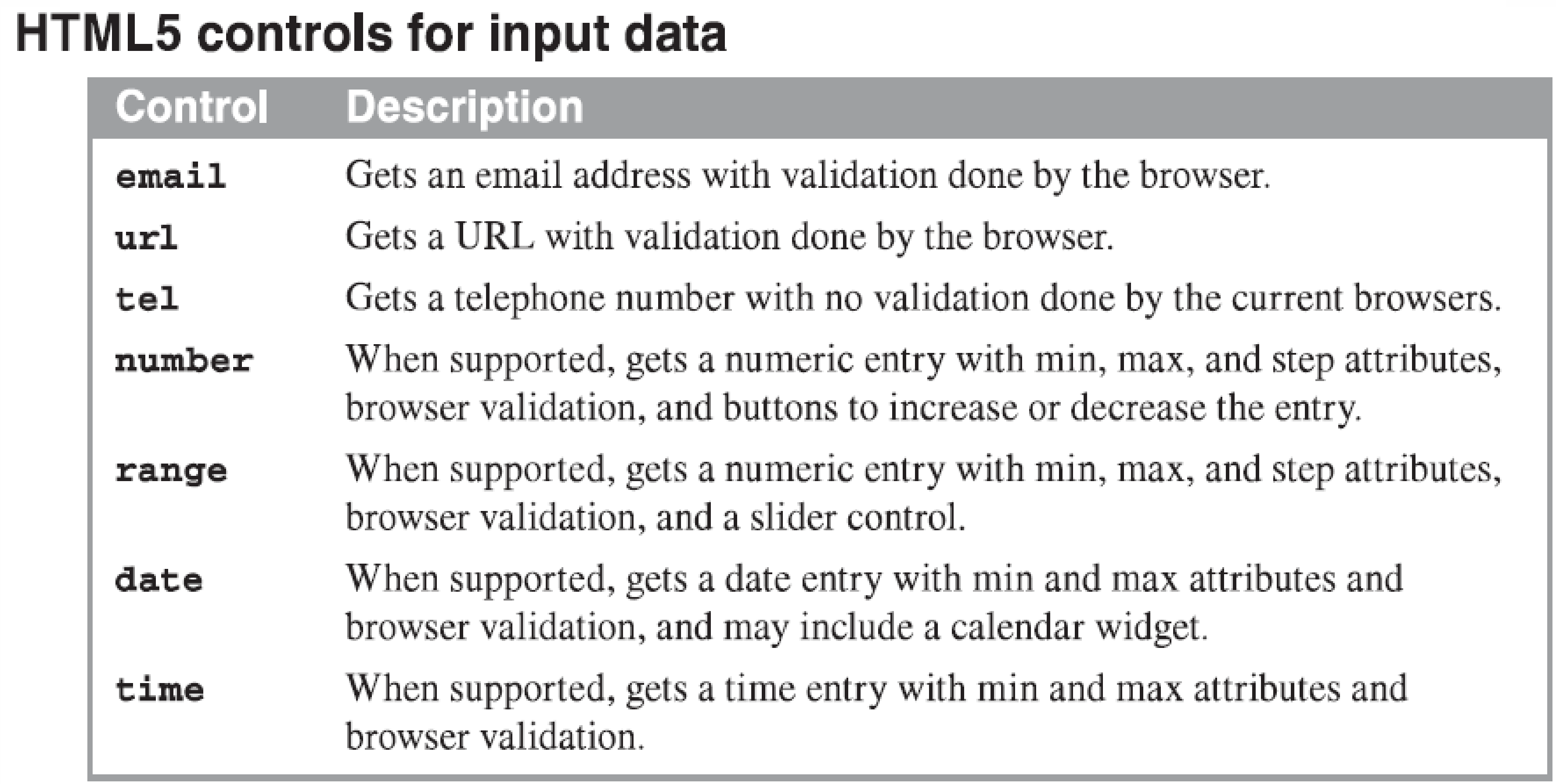
Attributes of the form element

Attribute	Description
`name`	A name that can be referred to by client-side or server-side code.
`action`	The URL of the file that will process the data in the form.
`method`	The HTTP method for submitting the form data. It can be set to either "get" or "post". The default value is "get".

Introduction to forms (cont.)

- When you click on ***summit button*** for the form the form data is sent to the server as a part of an HTML request.
- There are two method submit by data to the server are **get** and **post**.
- The **get** method will show the data on the URL, with the post method, the data is hidden.
- ***Data validation*** refers to checking the data collected by a form to make sure it is valid, and complete data validation is always done on the server.

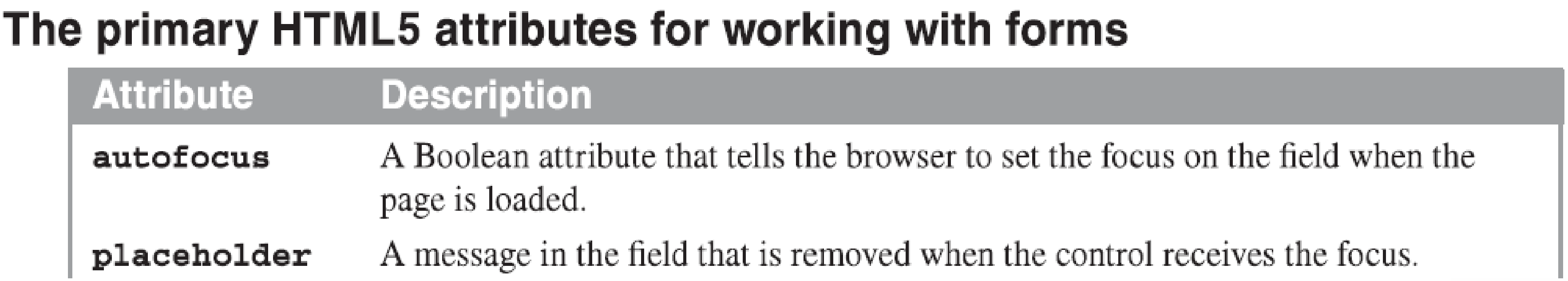
The HTML5 controls for working with forms



HTML5 controls for input data

Control	Description
`email`	Gets an email address with validation done by the browser.
`url`	Gets a URL with validation done by the browser.
`tel`	Gets a telephone number with no validation done by the current browsers.
`number`	When supported, gets a numeric entry with min, max, and step attributes, browser validation, and buttons to increase or decrease the entry.
`range`	When supported, gets a numeric entry with min, max, and step attributes, browser validation, and a slider control.
`date`	When supported, gets a date entry with min and max attributes and browser validation, and may include a calendar widget.
`time`	When supported, gets a time entry with min and max attributes and browser validation.

The HTML5 controls for working with forms (cont.)



The primary HTML5 attributes for working with forms

Attribute	Description
`autofocus`	A Boolean attribute that tells the browser to set the focus on the field when the page is loaded.
`placeholder`	A message in the field that is removed when the control receives the focus.

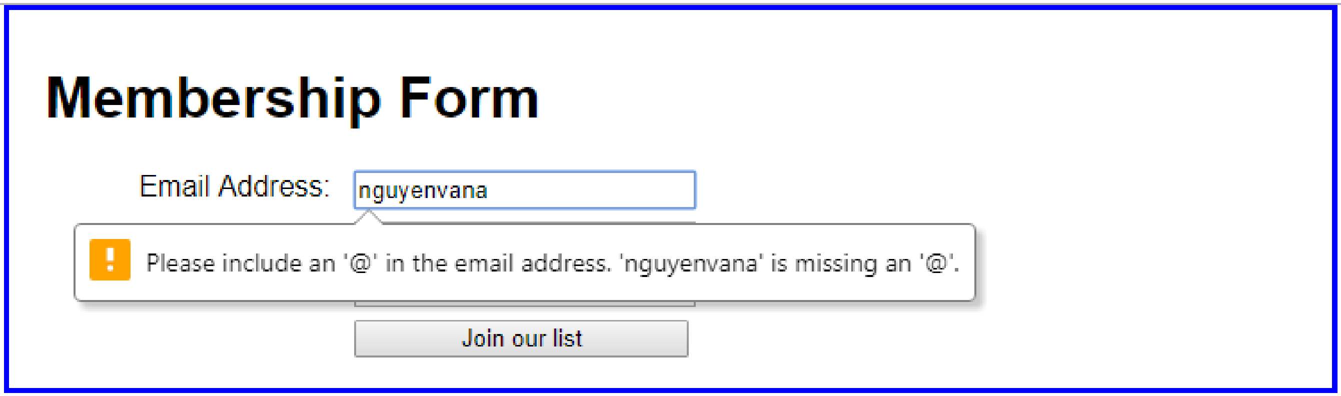
- Many of the HTML5 input controls provide for basic data validation.

The HTML5 controls for working with forms (cont.)

- The HTML for a form that uses some of these controls and attributes

```
<form action="register.html" method="get" name="email_form" id="email_form">
        <label for="email">Email Address:</label>
        <input type="email" id="email" name="email" autofocus><br>
        <label for="name">Name:</label>
        <input type="text" id="name" name="name"><br>
        <label for="phone">Phone Number:</label>
        <input type="tel" id="phone" name="phone" placeholder="999-999-9999"><br>
        <label >&nbsp;</label>
        <input type="submit" id="join_list" value="Join our list">
</form>
```

- The form with an error message for the email address



Membership Form

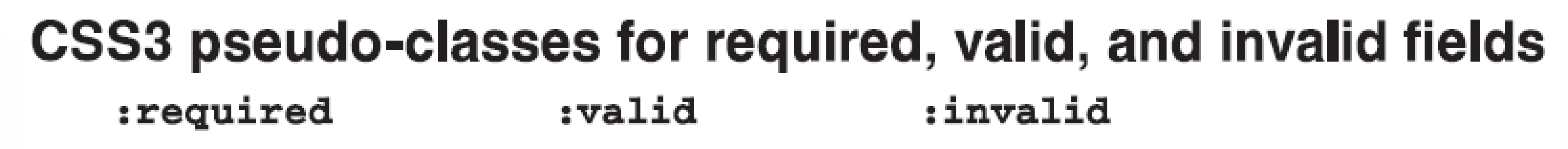
Email Address: nguyenvana

! Please include an '@' in the email address. 'nguyenvana' is missing an '@'.

The HTML5 and CSS3 features for data validation

- The HTML5 attributes for data validation

Attribute	Description
required	A Boolean attribute that indicates that a value is required for a field.
title	Text that is display in a tooltip when the mouse hover the field.
pattern	A regular expression that is used to validate the entry in a field.
novalidate	A Boolean attribute that tell the browser that it shoudn’t validate.
autocomplete	Set this attribute “off” to tell browser to disable auto-complete.



CSS3 pseudo-classes for required, valid, and invalid fields

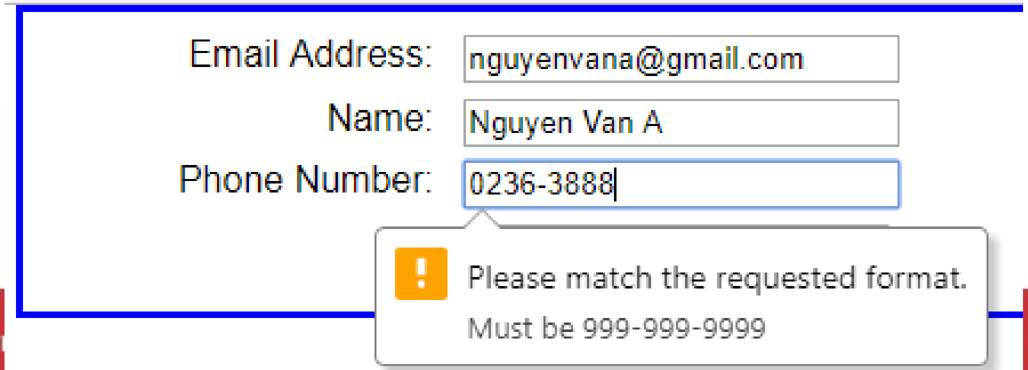
```
:required          :valid          :invalid
```

The HTML5 and CSS3 features for data validation (cont.)

- The HTML for a form that uses some of these attributes

```
<form action="register.html" method="get" name="email_form" id="email_form">
        <label for="email">Email Address:</label>
        <input type="email" id="email" name="email" autofocus
               required autocomplete="off"><br>
        <label for="name">Name:</label>
        <input type="text" id="name" name="name" required><br>
        <label for="phone">Phone Number:</label>
        <input type="tel" id="phone" name="phone" required
            pattern="\d{3}[\-]\d{3}[\-]\d{4}"
            title="Must be 999-999-9999"><br>
        <label >&nbsp;</label>
        <input type="submit" id="join_list" value="Join our list">
</form>
```

- The form with an error message for the phone field



Email Address: nguyenvana@gmail.com

Name: Nguyen Van A

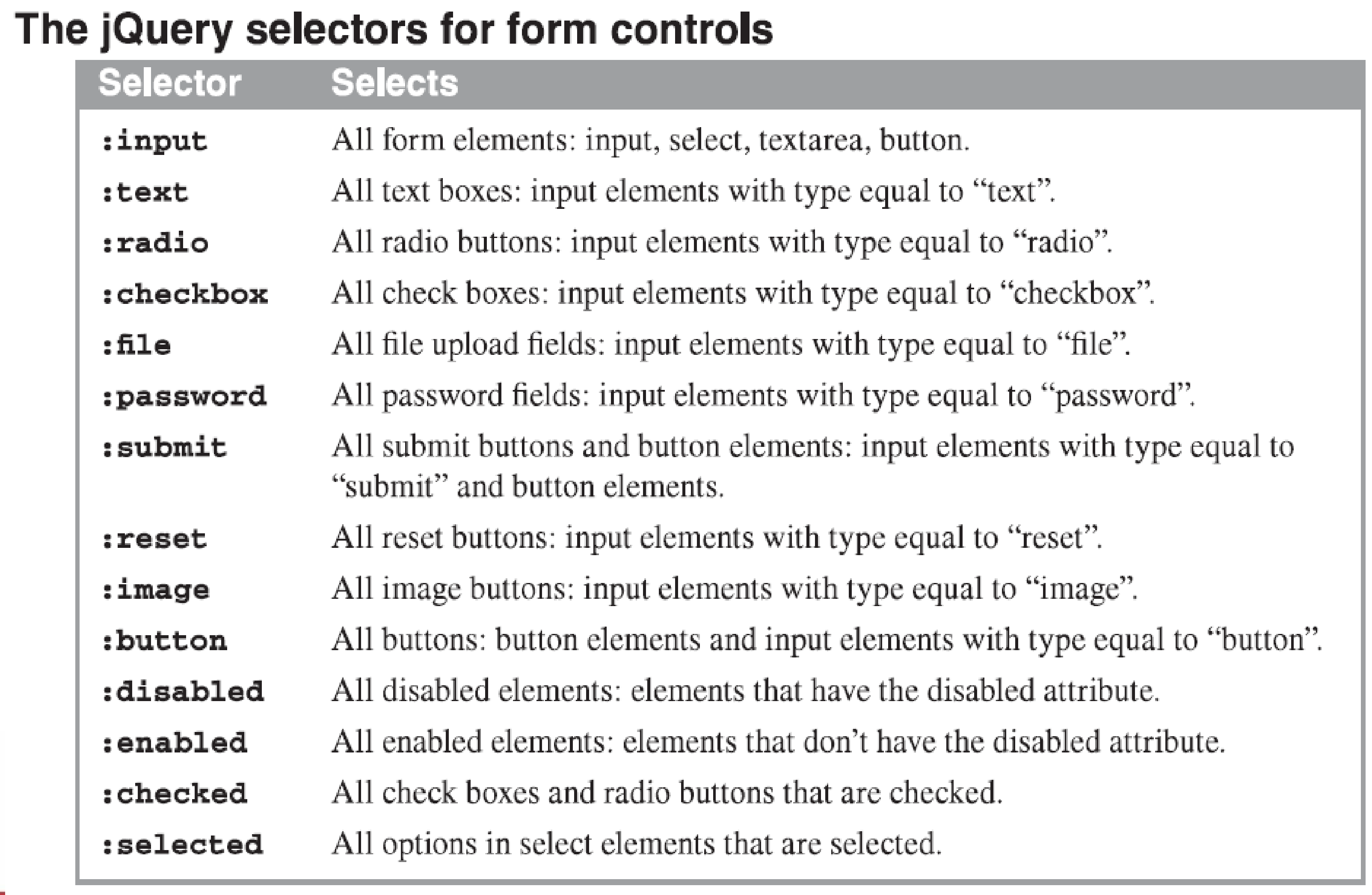
Phone Number: 0236-3888

Please match the requested format.

Must be 999-999-9999

How to use jQuery to work with forms

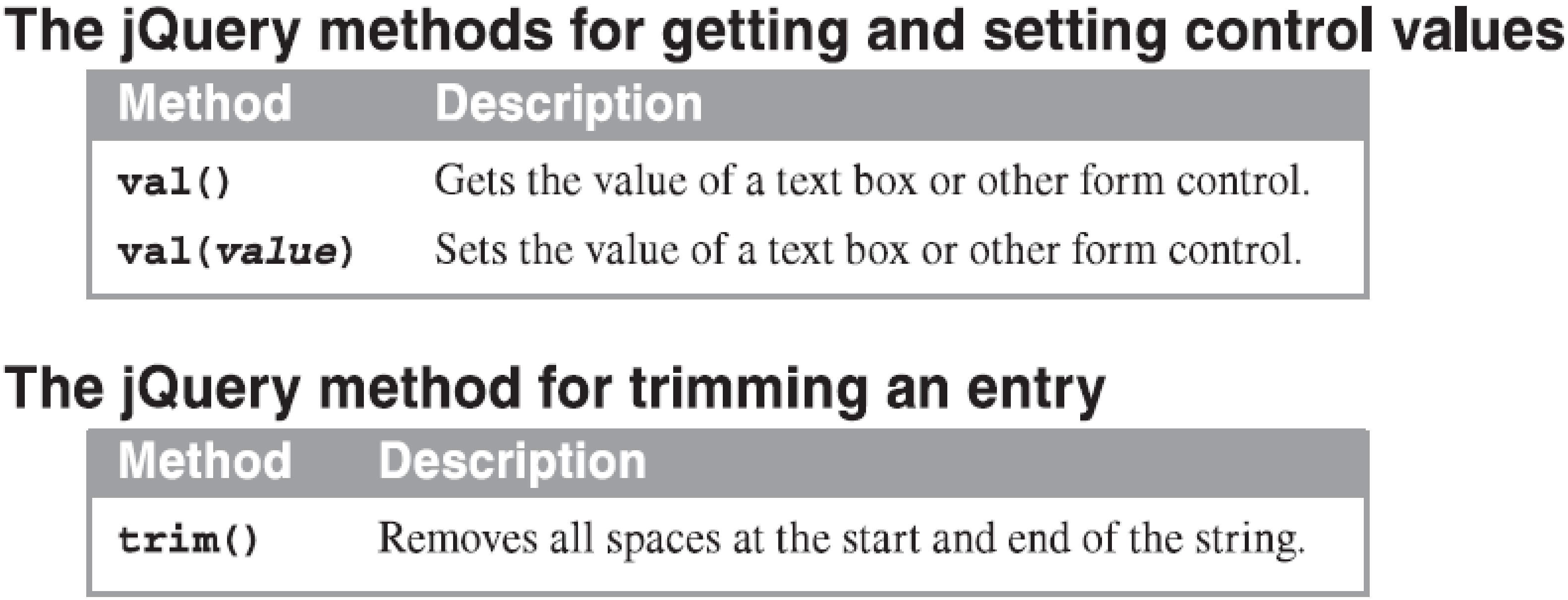
The jQuery selectors and methods for forms



The jQuery selectors for form controls

Selector	Selects
`:input`	All form elements: input, select, textarea, button.
`:text`	All text boxes: input elements with type equal to "text".
`:radio`	All radio buttons: input elements with type equal to "radio".
`:checkbox`	All check boxes: input elements with type equal to "checkbox".
`:file`	All file upload fields: input elements with type equal to "file".
`:password`	All password fields: input elements with type equal to "password".
`:submit`	All submit buttons and button elements: input elements with type equal to "submit" and button elements.
`:reset`	All reset buttons: input elements with type equal to "reset".
`:image`	All image buttons: input elements with type equal to "image".
`:button`	All buttons: button elements and input elements with type equal to "button".
`:disabled`	All disabled elements: elements that have the disabled attribute.
`:enabled`	All enabled elements: elements that don't have the disabled attribute.
`:checked`	All check boxes and radio buttons that are checked.
`:selected`	All options in select elements that are selected.

The jQuery selectors and methods for forms (cont.)



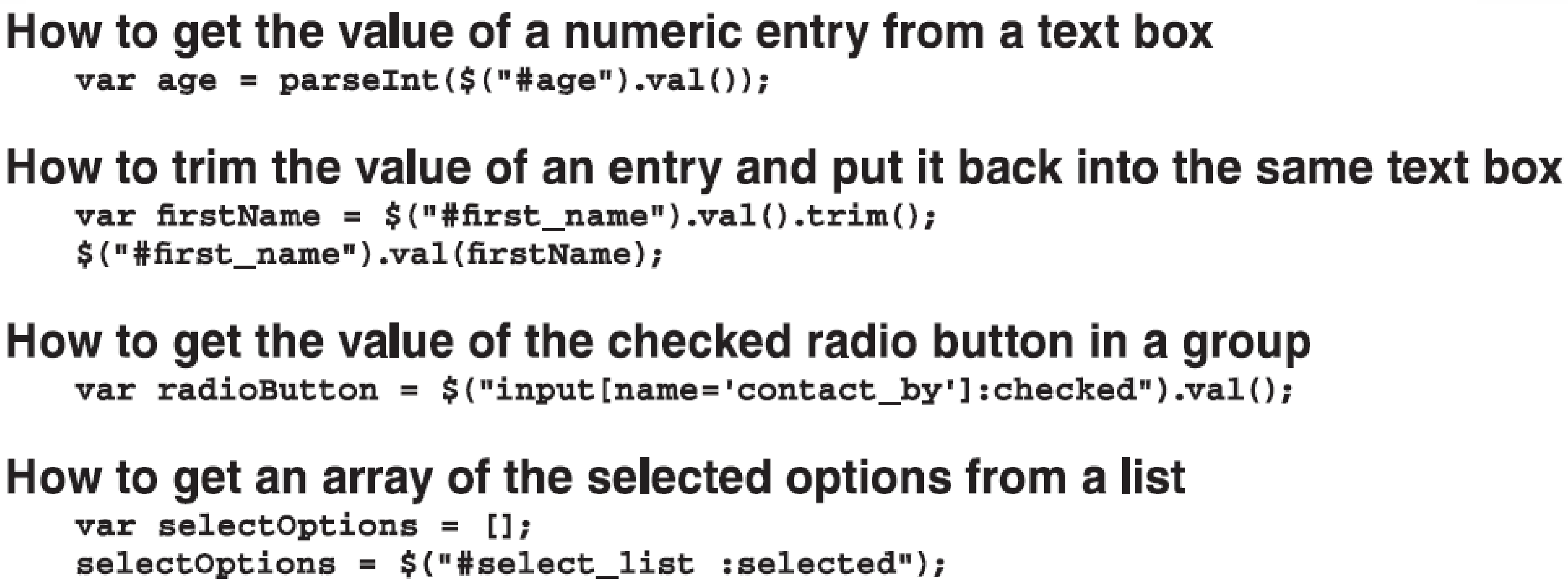
The jQuery methods for getting and setting control values

Method	Description
`val()`	Gets the value of a text box or other form control.
`val(value)`	Sets the value of a text box or other form control.

The jQuery method for trimming an entry

Method	Description
`trim()`	Removes all spaces at the start and end of the string.

The jQuery selectors and methods for forms (cont.)



How to get the value of a numeric entry from a text box

```
var age = parseInt($("#age").val());
```

How to trim the value of an entry and put it back into the same text box

```
var firstName = $("#first_name").val().trim();
$("#first_name").val(firstName);
```

How to get the value of the checked radio button in a group

```
var radioButton = $("input[name='contact_by']:checked").val();
```

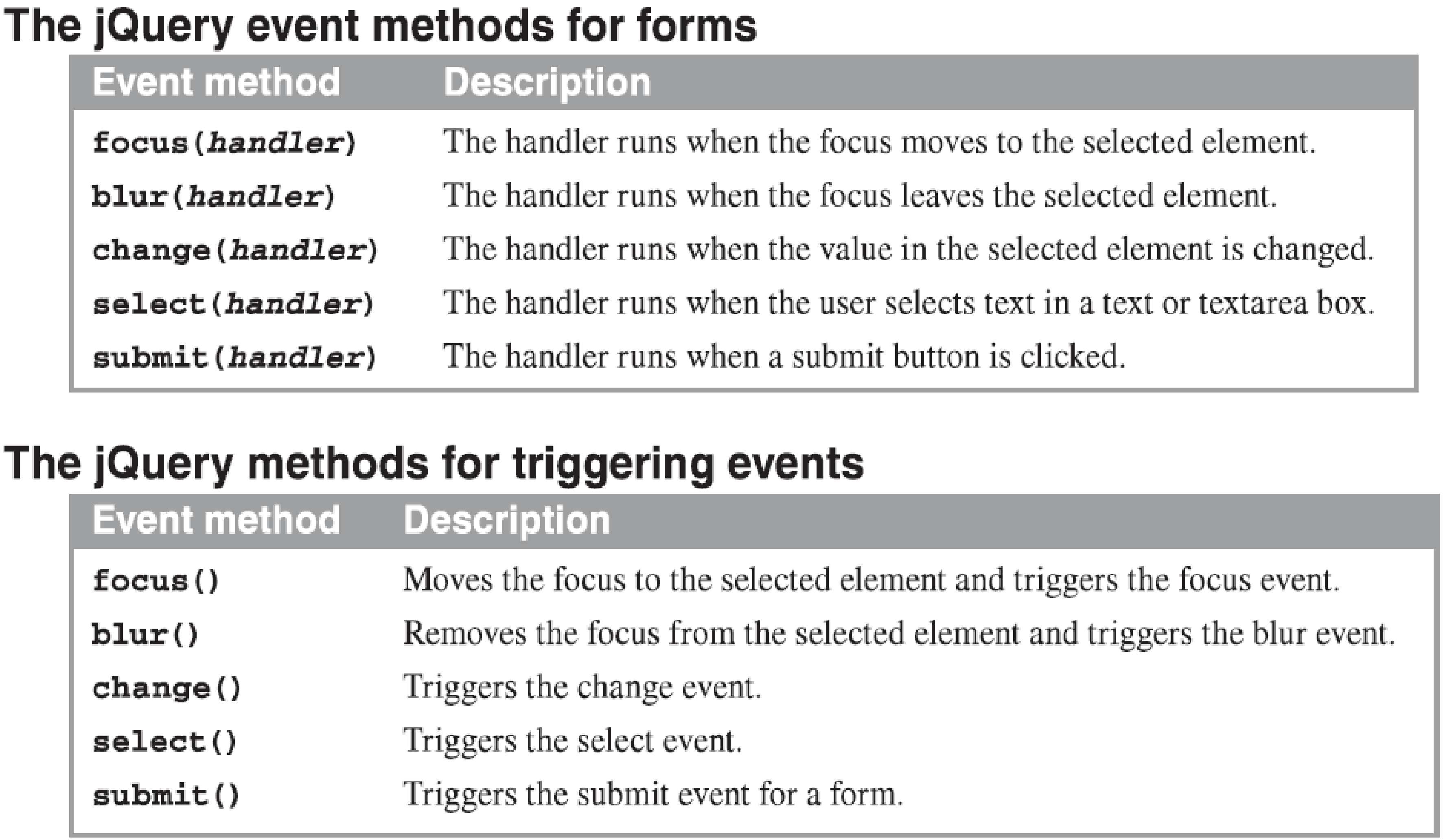
How to get an array of the selected options from a list

```
var selectOptions = [];
selectOptions = $("#select_list :selected");
```

The jQuery event methods for forms

- You can use event handlers for the focus, blur, change, and select events to process data as the user works with individual controls.
- You can use event handler for the click event of a regular button to validate data in a form.
- Also you can use an event for the submit button to validate data in a form. Then if any data is invalid, you can issue the preventDefault() method to cancel the submission of the data to the server.

The jQuery event methods for forms (cont.)



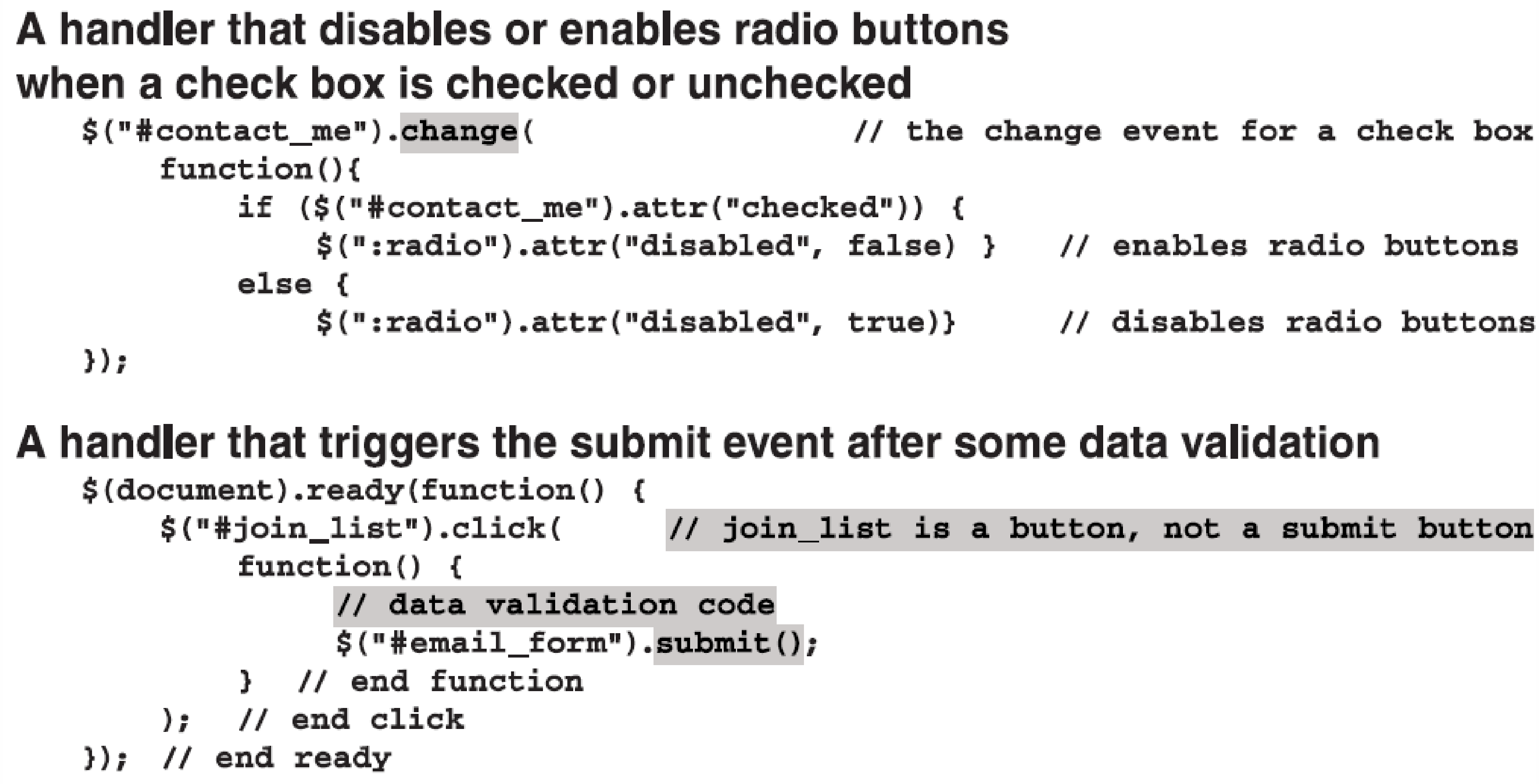
The jQuery event methods for forms

Event method	Description
`focus(handler)`	The handler runs when the focus moves to the selected element.
`blur(handler)`	The handler runs when the focus leaves the selected element.
`change(handler)`	The handler runs when the value in the selected element is changed.
`select(handler)`	The handler runs when the user selects text in a text or textarea box.
`submit(handler)`	The handler runs when a submit button is clicked.

The jQuery methods for triggering events

Event method	Description
`focus()`	Moves the focus to the selected element and triggers the focus event.
`blur()`	Removes the focus from the selected element and triggers the blur event.
`change()`	Triggers the change event.
`select()`	Triggers the select event.
`submit()`	Triggers the submit event for a form.

The jQuery event methods for forms (cont.)



A handler that disables or enables radio buttons when a check box is checked or unchecked

```
$("#contact_me").change(                    // the change event for a check box
    function(){
        if ($("#contact_me").attr("checked")) {
            $(":radio").attr("disabled", false) }   // enables radio buttons
        else {
            $(":radio").attr("disabled", true)}     // disables radio buttons
});
```

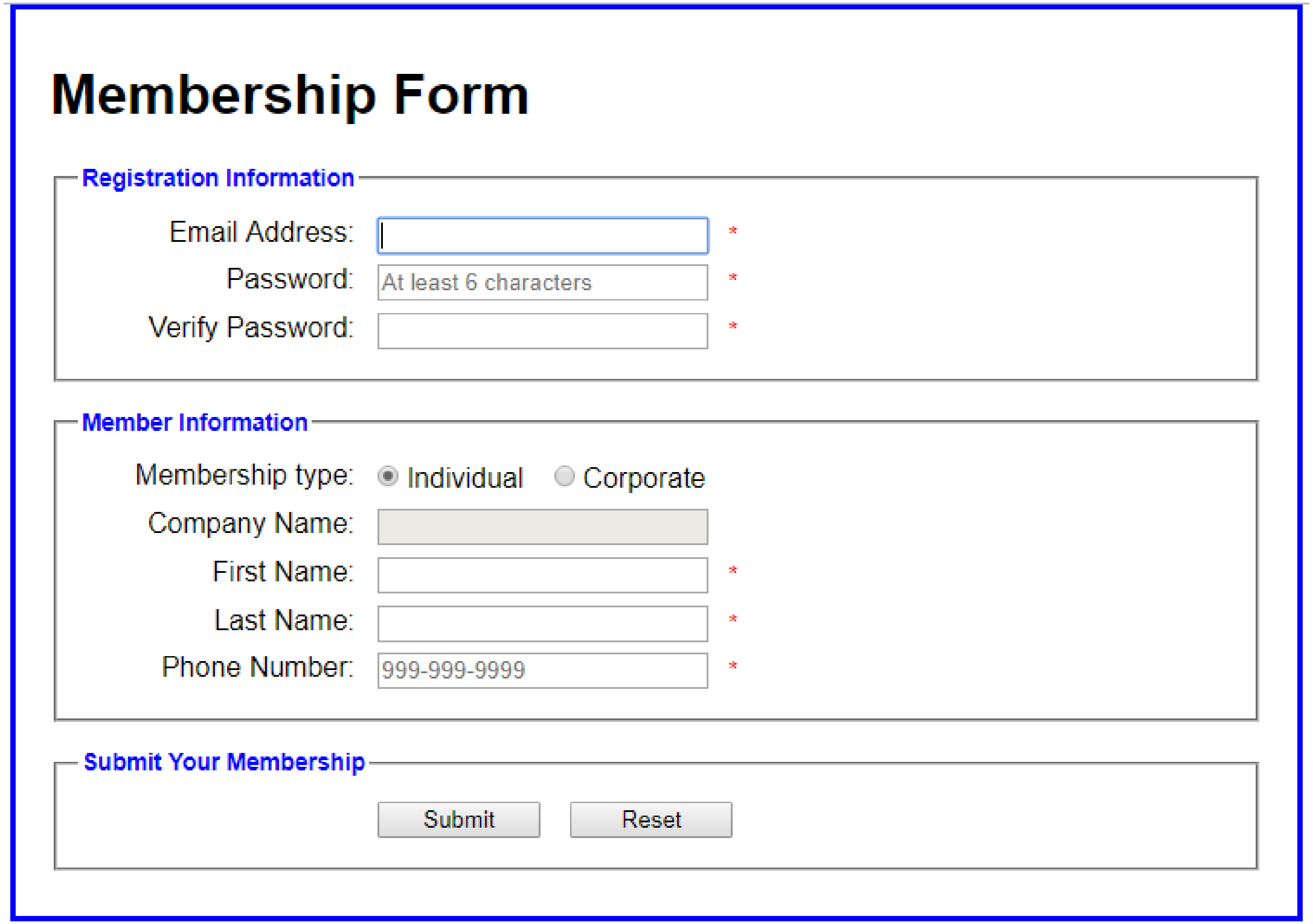
A handler that triggers the submit event after some data validation

```
$(document).ready(function() {
    $("#join_list").click(     // join_list is a button, not a submit button
        function() {
             // data validation code
             $("#email_form").submit();
        }  // end function
    );  // end click
}); // end ready
```

A Validation application that uses JavaScript

A Validation application

- The web interface



Membership Form

Registration Information

Email Address:		*
Password:	At least 6 characters	*
Verify Password:		*

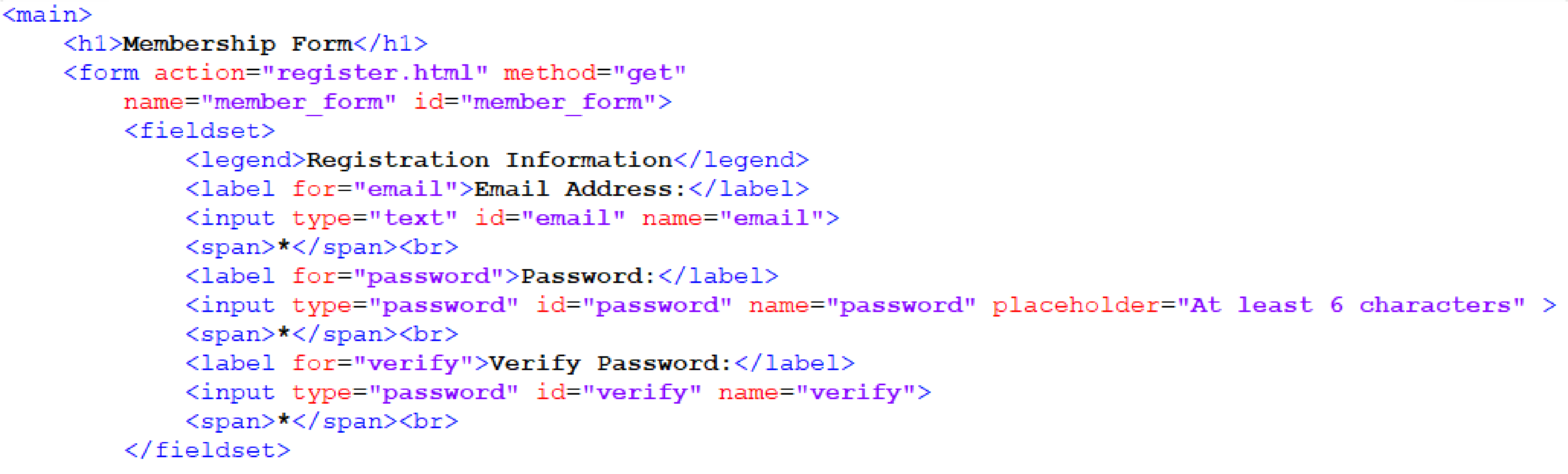
Member Information

Membership type:	● Individual ○ Corporate	
Company Name:		
First Name:		*
Last Name:		*
Phone Number:	999-999-9999	*

Submit Your Membership

A Validation application (cont.)

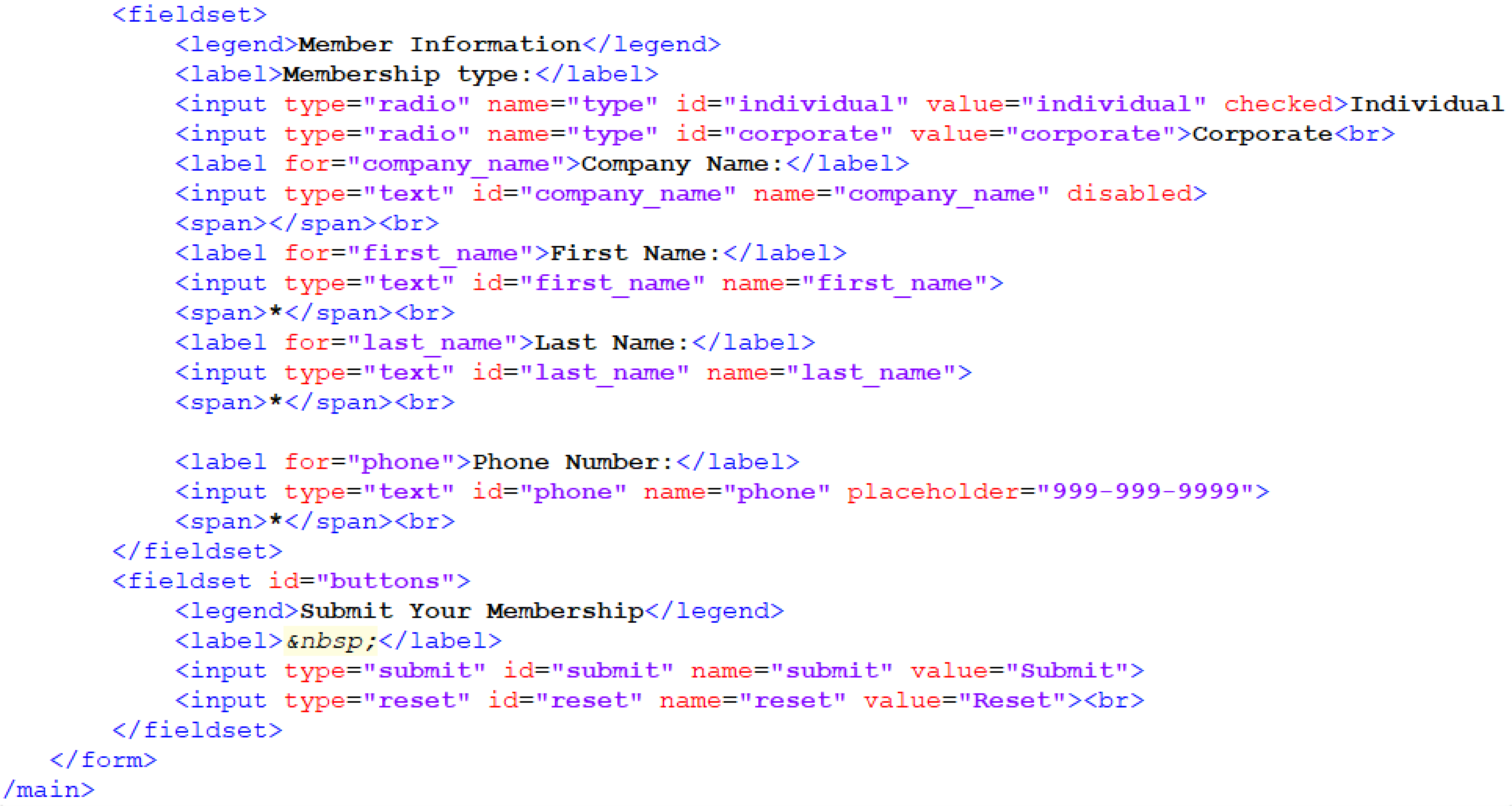
- The HTML code



```
<main>
    <h1>Membership Form</h1>
    <form action="register.html" method="get"
        name="member_form" id="member_form">
        <fieldset>
            <legend>Registration Information</legend>
            <label for="email">Email Address:</label>
            <input type="text" id="email" name="email">
            <span>*</span><br>
            <label for="password">Password:</label>
            <input type="password" id="password" name="password" placeholder="At least 6 characters" >
            <span>*</span><br>
            <label for="verify">Verify Password:</label>
            <input type="password" id="verify" name="verify">
            <span>*</span><br>
        </fieldset>
```

A Validation application (cont.)

- The HTML code

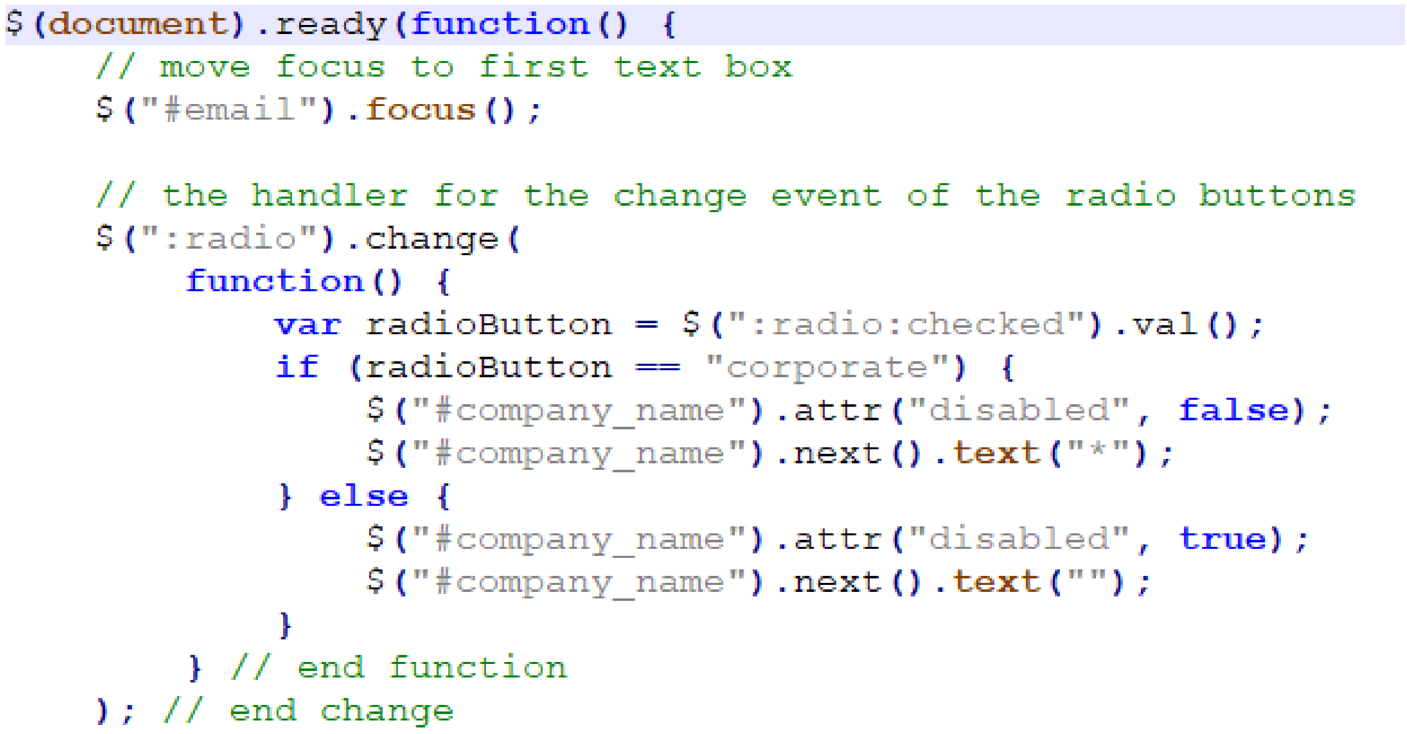


```
       <fieldset>
           <legend>Member Information</legend>
           <label>Membership type:</label>
           <input type="radio" name="type" id="individual" value="individual" checked>Individual
           <input type="radio" name="type" id="corporate" value="corporate">Corporate<br>
           <label for="company_name">Company Name:</label>
           <input type="text" id="company_name" name="company_name" disabled>
           <span></span><br>
           <label for="first_name">First Name:</label>
           <input type="text" id="first_name" name="first_name">
           <span>*</span><br>
           <label for="last_name">Last Name:</label>
           <input type="text" id="last_name" name="last_name">
           <span>*</span><br>

           <label for="phone">Phone Number:</label>
           <input type="text" id="phone" name="phone" placeholder="999-999-9999">
           <span>*</span><br>
       </fieldset>
       <fieldset id="buttons">
           <legend>Submit Your Membership</legend>
           <label>&nbsp;</label>
           <input type="submit" id="submit" name="submit" value="Submit">
           <input type="reset" id="reset" name="reset" value="Reset"><br>
       </fieldset>
   </form>
/main>
```

A Validation application (cont.)

- The JavaScript code

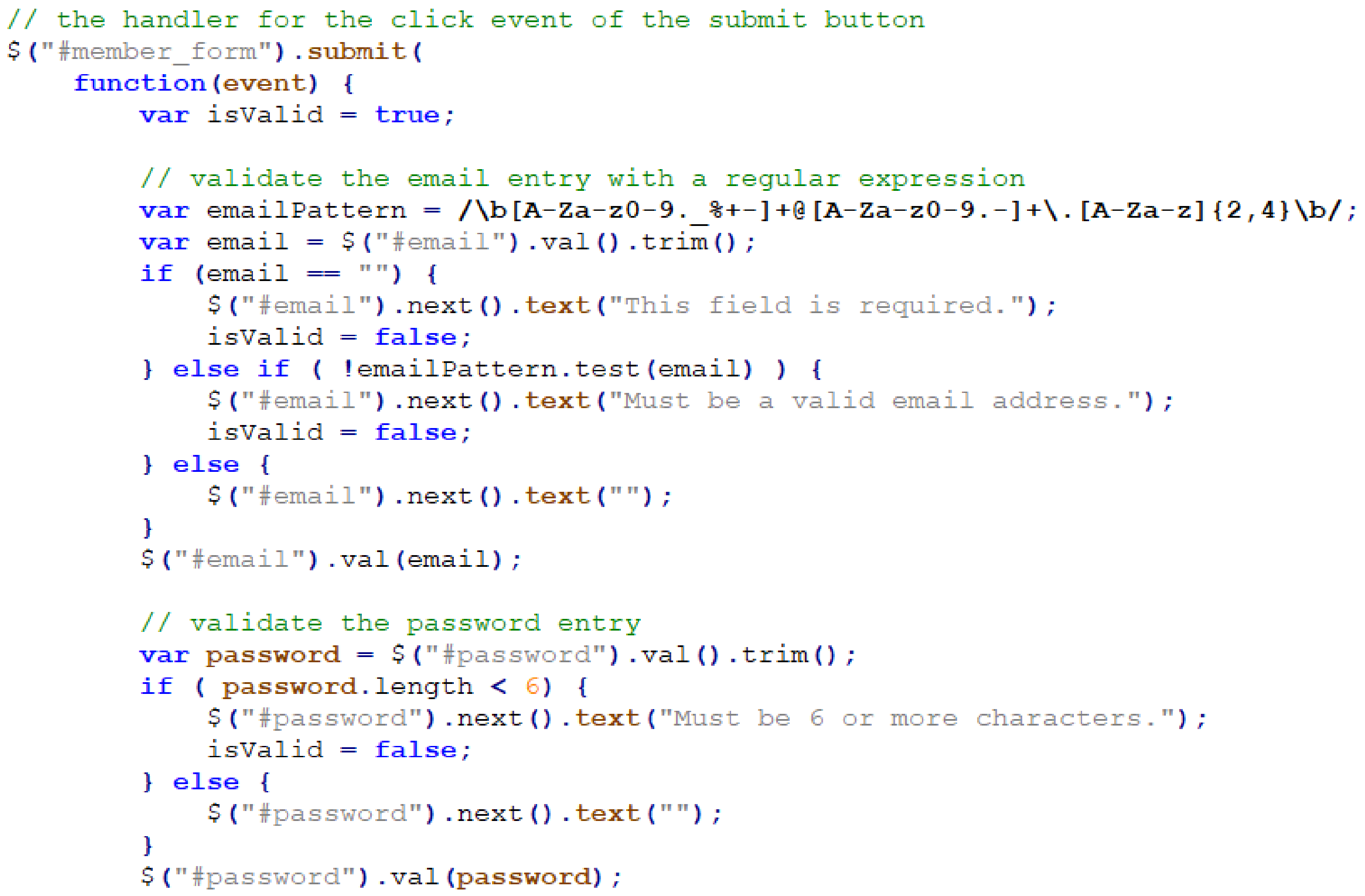


```
$(document).ready(function() {
    // move focus to first text box
    $("#email").focus();

    // the handler for the change event of the radio buttons
    $(":radio").change(
        function() {
            var radioButton = $(":radio:checked").val();
            if (radioButton == "corporate") {
                $("#company_name").attr("disabled", false);
                $("#company_name").next().text("*");
            } else {
                $("#company_name").attr("disabled", true);
                $("#company_name").next().text("");
            }
        } // end function
    ); // end change
```

A Validation application (cont.)

- The JavaScript code

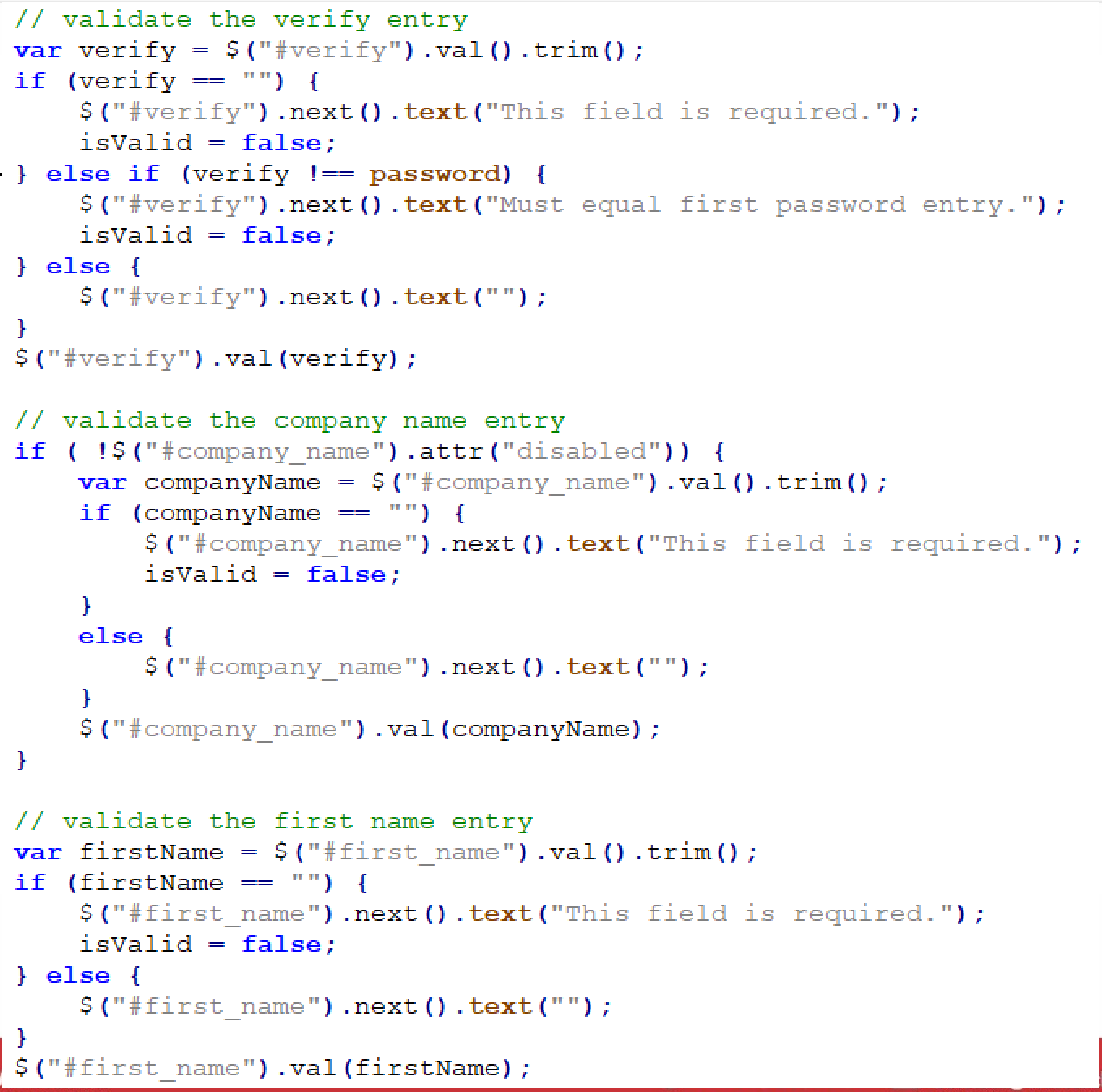


```
// the handler for the click event of the submit button
$("#member_form").submit(
    function(event) {
        var isValid = true;

        // validate the email entry with a regular expression
        var emailPattern = /\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,4}\b/;
        var email = $("#email").val().trim();
        if (email == "") {
            $("#email").next().text("This field is required.");
            isValid = false;
        } else if ( !emailPattern.test(email) ) {
            $("#email").next().text("Must be a valid email address.");
            isValid = false;
        } else {
            $("#email").next().text("");
        }
        $("#email").val(email);

        // validate the password entry
        var password = $("#password").val().trim();
        if ( password.length < 6) {
            $("#password").next().text("Must be 6 or more characters.");
            isValid = false;
        } else {
            $("#password").next().text("");
        }
        $("#password").val(password);
```

- T



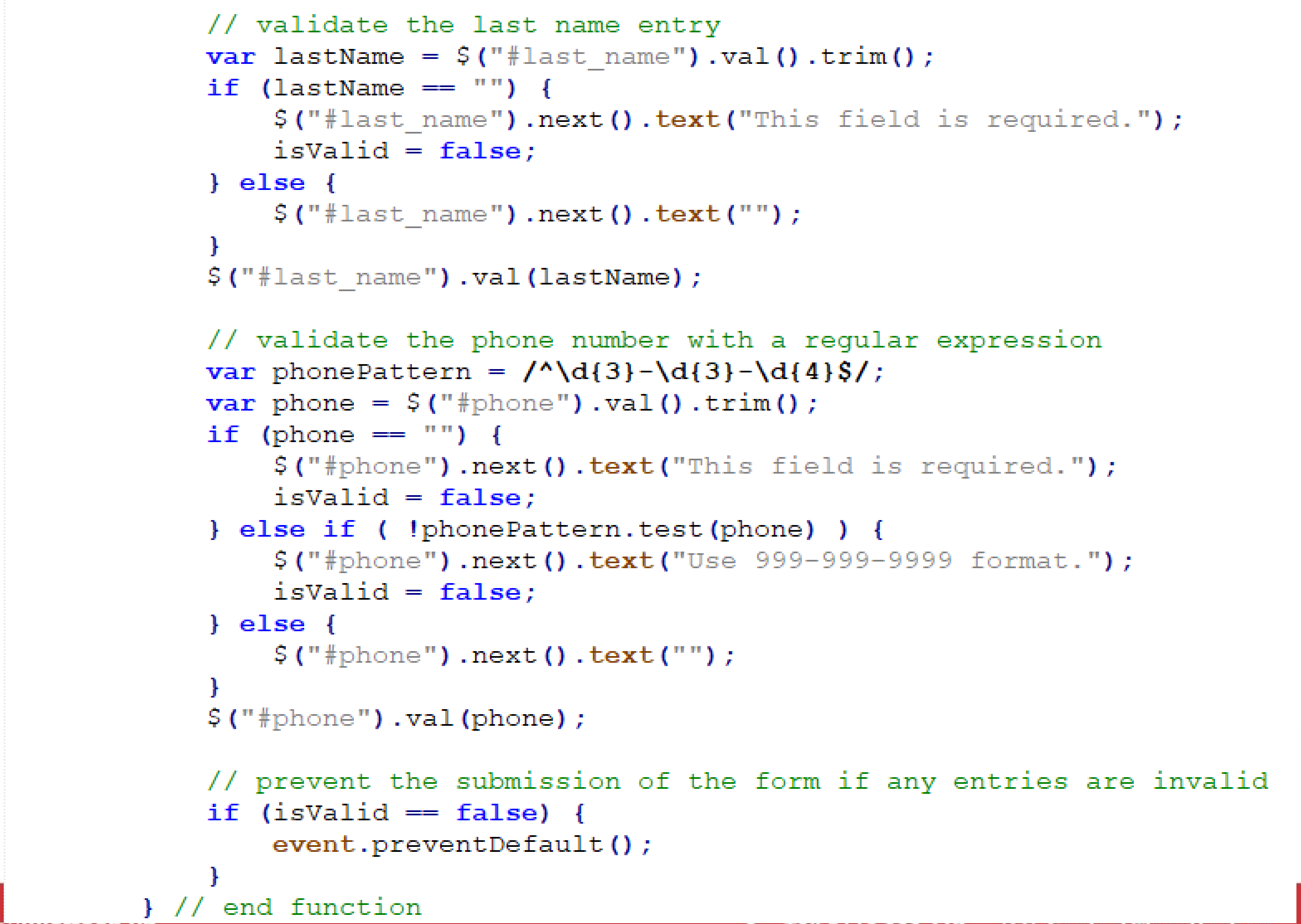
```
// validate the verify entry
var verify = $("#verify").val().trim();
if (verify == "") {
    $("#verify").next().text("This field is required.");
    isValid = false;
} else if (verify !== password) {
    $("#verify").next().text("Must equal first password entry.");
    isValid = false;
} else {
    $("#verify").next().text("");
}
$("#verify").val(verify);

// validate the company name entry
if ( !$("#company_name").attr("disabled")) {
    var companyName = $("#company_name").val().trim();
    if (companyName == "") {
        $("#company_name").next().text("This field is required.");
        isValid = false;
    }
    else {
        $("#company_name").next().text("");
    }
    $("#company_name").val(companyName);
}

// validate the first name entry
var firstName = $("#first_name").val().trim();
if (firstName == "") {
    $("#first_name").next().text("This field is required.");
    isValid = false;
} else {
    $("#first_name").next().text("");
}
$("#first_name").val(firstName);
```

A Validation application (cont.)

- The JavaScript code



```
        // validate the last name entry
        var lastName = $("#last_name").val().trim();
        if (lastName == "") {
            $("#last_name").next().text("This field is required.");
            isValid = false;
        } else {
            $("#last_name").next().text("");
        }
        $("#last_name").val(lastName);

        // validate the phone number with a regular expression
        var phonePattern = /^\d{3}-\d{3}-\d{4}$/;
        var phone = $("#phone").val().trim();
        if (phone == "") {
            $("#phone").next().text("This field is required.");
            isValid = false;
        } else if ( !phonePattern.test(phone) ) {
            $("#phone").next().text("Use 999-999-9999 format.");
            isValid = false;
        } else {
            $("#phone").next().text("");
        }
        $("#phone").val(phone);

        // prevent the submission of the form if any entries are invalid
        if (isValid == false) {
            event.preventDefault();
        }
    } // end function
```

Summary

- A ***forms*** contain one or more ***controls*** like text boxes, radio buttons, list and check boxes that can receive data.
- When you click on ***summit button*** for the form the form data is sent to the server as a part of an HTML request.
- The **get** method will show the data on the URL, with the **post** method, the data is hidden.
- ***Data validation*** refers to checking the data collected by a form to make sure it is valid, and complete data validation is always done on the server.

Summary (cont.)

- jQuery provides special selectors for selecting the controls on the form, the val() method for getting and setting the value in a control, and a trim() method for trimming an entry.
- You can use event handlers for the focus, blur, change, and select events to process data as the user works with individual controls.
- You can use event handler for the click event of a regular button to validate data in a form.

The End.