JavaScript, Sixth Edition

Chapter 6 Solutions

Short Quiz 1

1. Describe two common uses of JavaScript with forms.

JavaScript is commonly used with forms for two reasons: to add functionality that makes forms easier for users to fill out, and to validate or process the data that a user enters before that data is submitted to a server-side script.

1. Name two advantages of identifying form elements using the getElementById() method rather than by specifying their index numbers within the form.

One advantage of referencing form objects using methods such as getElementById() rather than the forms collection is that you don’t need to switch between using one syntax for referencing form objects, and another syntax for non-form objects. In addition, using Document object methods makes your code more flexible when you need the same code to be able to refer to both form and nonform elements. Finally, use of the name attribute is not allowed in the strict DTD for XHTML, so if you’re writing JavaScript to work with XHTML documents, you cannot use the forms array or other browser arrays.

Short Quiz 2

1. If you were designing a form with a question that asked users if they owned a bicycle, would you expect more accurate input using a text input box or a check box? Why?

A check box would provide more accurate input. It would allow users to respond only yes (checked) or no (unchecked). If users had to respond by typing in a text input box, they might misspell their input, or enter text that’s not easily interpretable as yes or no.

1. What property do you use to change the selected option element in a selection list? What value do you use to specify that no option element is selected?

selectedIndex

-1

Short Quiz 3

1. List two attributes that specify parameters for browser-based validation in child elements of a form, and describe what each does.

See Table 6-12

1. List a value for the type attribute of the input element that triggers browser-based validation, and describe what content browsers check for.

See Table 6-13

1. What is the constraint validation API?

A set of properties and methods that enable you to customize many aspects of the way browsers present browser-based validation feedback.

1. What is one drawback of using browser-based validation?

No aspect of bubble appearance aside from text is customizable, meaning that the color, shape, and size varies among browsers.

Browser-based validation doesn’t allow you to set multiple validation messages.

Short Quiz 4

1. Which event fires when a form is submitted?

submit

1. What method do you use to disable the default behavior associated with an event?

preventDefault()

1. What property do you check to determine if a text input box contains user input?

value

1. What built-in JavaScript function do you use to test whether a value is not a number?

isNaN()

# Review Questions

1. Objects representing each of the controls in a form are stored in the \_\_\_\_\_ collection.
   1. forms
   2. controls
   3. inputs
   4. elements
2. Which of the following type values for the input element does *not* enable you to provide users with a limited set of choices?
   1. radio
   2. email
   3. checkbox
   4. range
3. What value of the selectedIndex property of a select object corresponds to no selection?
   1. -1
   2. 0
   3. 1
   4. false
4. To simulate the behavior of placeholder text in older browsers, you can instead set the value of the \_\_\_\_\_\_\_ property.
   1. src
   2. alt
   3. title
   4. value
5. Which event do you use to call a function when a user selects a field or moves the insertion point into a field?
   1. blur
   2. focus
   3. input
   4. forminput
6. Which event do you use to call a function when a field is no longer selected, or a user moves the insertion point to a different field?
   1. blur
   2. focus
   3. click
   4. forminput
7. Which of the following attributes determines whether a check box or option button is selected?
   1. checked
   2. defaultChecked
   3. selected
   4. focus
8. What do you assign to the value property of a text input box to remove its content?
   1. false
   2. true
   3. ""
   4. null
9. Which of the following attributes triggers browser-based validation in modern browsers?
   1. max
   2. title
   3. alt
   4. src
10. Which of the following input type values triggers browser-based validation in modern browsers?
    1. password
    2. text
    3. radio
    4. number
11. Which of the following properties has a value of true when a user has left a required field blank?
    1. required
    2. valueMissing
    3. patternMismatch
    4. typeMismatch
12. Which of the following attributes for form child elements would you use for a field that must have a value before the form can be submitted?
    1. novalidate
    2. min
    3. required
    4. max
13. What method do you use to disable the default behavior for an event?
    1. preventDefault()
    2. checkValidity()
    3. select()
    4. getElementById()
14. Which statement moves the browser to the top of the page?
    1. scroll(top)
    2. scroll(0,0)
    3. move(top)
    4. move(0,0)
15. For any fields that require numeric values, you can use JavaScript’s built-in \_\_\_\_\_\_\_\_ function to determine whether the user actually entered a number.
    1. value()
    2. integer()
    3. isNumber()
    4. isNaN()
16. Explain how to transfer the contents of one field to another field.

You look up the value property of the source field and assign it to the value property for the target field.

1. What is the purpose of the novalidate attribute?

The novalidate attribute toggles off validation of a form when added to the opening <form> tag.

1. Explain how the validity object of the constraint validation API is used for checking the validity of form data.

The validity object contains several properties; if all of these properties have a value of false, then the value of the validity object is true.

1. Explain how to check if any option button in a set is selected.

To check if an option button is selected, you access the value of its checked property. To check if none of the option buttons in a set are selected, you can create a conditional statement using And (&&) operators. This code could check if the first button is not selected (using the ! operator), and (&&) if the second button is not selected, etc. If all of those conditions are true, the if statement is true, meaning that no button is checked.

1. Explain how to check if a user’s entry is a number.

For any fields that require numeric values, you can use JavaScript’s built-in isNaN() function to determine whether the user actually entered a number. Recall from Chapter 2 that the isNaN() function determines whether a value is the special value NaN (not a number). The isNaN() function returns a value of true if it is passed a value that is not a number; if passed a value that is a number, the function returns a value of false.

# Case Projects

## Individual Case Project

Add validation the code for one of the forms on your individual website. First, ensure that your form uses at least three of the following field types: check boxes, text boxes, option buttons, selection lists, and text areas. Then, program validation for your questionnaire, ensuring that users enter values or make selections in all fields, and verifying at least one other aspect of at least one of the fields. Provide appropriate feedback to users when the form fails validation. Test your completed program until all validation works reliably with different combinations of valid and erroneous data.

Grading rubric: Students should submit a web page that contains a form that incorporates as least three of the following field types: check boxes, text boxes, option buttons, selection lists, and text areas. The page should incorporate validation code checking that a user has completed every field. The validation should also check an additional aspect of at least one of the fields, such as checking a field value based on the value of a different field, or checking for appropriate content type. The form should provide appropriate feedback to the user when a validation error is identified.

## Team Case Project

Add validation code to one of the forms on your team website. First, ensure that your form uses at least three of the following field types: check boxes, text boxes, option buttons, selection lists, and text areas. Next, as a team, plan validation for each field in the form. Your validation should require a value in each field, and should verify at least one other aspect of at least one field. Divide your team into two groups—one that will write code to verify that all fields have values, and the other to write code to verify another aspect of the entered data. Each group’s code should also incorporate appropriate feedback to users when it encounters validation errors. When both groups are done, work as a team to integrate the code into the document. Strategize as a team about how to test for all possible validation scenarios. Test and debug the code until your completed program until all validation works reliably with different combinations of valid and erroneous data.

Grading rubric: Each team should submit a web page that contains a form that incorporates as least three of the following field types: check boxes, text boxes, option buttons, selection lists, and text areas. The page should incorporate validation code checking that a user has completed every field. The validation should also check an additional aspect of at least one of the fields, such as checking a field value based on the value of a different field, or checking for appropriate content type. The form should provide appropriate feedback to the user when a validation error is identified.