CHAPTER 16

Make Web Pages Do Something Using JavaScript

JavaScript is the language of the browser. It is an object-oriented programming language. Although JavaScript looks much like Java (because both were based on C and C++), it is not at all related. JavaScript is often used to programmatically interact with an HTML page. It does this by interacting with the DOM (document object model. JavaScript is supported by all major browsers).

Include JavaScript in your web page by putting the script between `<script>` and `</script>` tags. You can place the script tags in the head section or at the bottom of page. If you place the JavaScript on top of your page or between the `<head>` tags, the user may see a blank page for a few seconds. However, once the page is loaded, everything will be fully functional from the first second. If you place the JavaScript at the bottom of the page, the page will seem to load faster, but the JavaScript will not run until the page (and script) is fully loaded. JavaScript may also be saved in a text file (no script tags are needed in this case) and referenced in the head section of your page. This allows you to reuse the same JavaScript on multiple pages. Using linked files is better from a maintenance perspective since all the JavaScript resides in only one location, making updates easy.

D. Wolf and A.J. Henley, *Java EE Web Application Primer*,
https://doi.org/10.1007/978-1-4842-3195-1_16

Because many developers find JavaScript challenging to work with, various libraries have been developed over the years to simplify the routine tasks of working with JavaScript. JQuery (`http://www.jquery.com`) works across all browsers that support JavaScript and makes working with JavaScript much more consistent. JQuery is used extensively in BootStrap. We will look at BootStrap shortly; it makes working with JQuery, HTML, and CSS even more fun.

Note Manipulating the DOM is one of JavaScript's more powerful uses. With DOM, you can navigate through and modify an entire page, ranging from simply adding an element to rearranging several areas on the page. DOM breaks up a document into a *tree* of *nodes*.

Validate a Form Using JavaScript

The form in Listing 16-1 is used to submit a post to Bullhorn. The post should be validated before the user attempts to submit it. JavaScript allows us to do this at the browser. The JavaScript method to validate the form is shown in the listing and contains one method, `validate()`. The JavaScript should be placed between `<script>...</script>` tags at the bottom of the page just before the closing body tag, `</body>`. Placing the script after the elements it references ensures the elements have been created by the DOM before the script is executed. The `validate` method looks at the element with an ID of `posttext` and returns false if the length of this text box is 0 (empty post). A false return will prevent the form from being submitted.

Listing 16-1. A JavaScript Function to Validate the Form Can Go Between Script Tags at the Bottom of the Web Page, Just Before the Closing Body Tag

```
function validate() {
        valid = true;
        if ($('#posttext').val().length==0){
        alert("You may not submit an empty post.");
        valid = false;
        }
return valid;
}
```

The HTML form that will use the preceding validation script goes on your web page within the `<body>...</body>` tags and before your JavaScript. This form will call the script when the Submit button is clicked. If the `validate` method returns false then the form will not be submitted. See Listing 16-2. The `onsubmit` attribute of the `form` tag calls the JavaScript function to validate the form.

Listing 16-2. Form for Submitting a Post

```
<form role="form"
                action="PostServ" method="post"
onsubmit="return validate();">
<label for="post">Create New Post (141 char):</label>
<textarea name="posttext" id="posttext"
                Maxlength="141"></textarea>

<div id="textarea_feedback"></div>


<input type="submit" value="Submit" id="submit"/>
<input type="reset" value="Clear"/>
</form>
```

Display Number of Characters in Text Box

We can also use JavaScript to count the number of characters remaining and update the web page dynamically as the user types. This is an excellent example of the power of JavaScript. It shows that it can be used to manipulate the web page at the browser. This JavaScript function will load when the document is ready. The document is ready after it has been fully rendered and all the DOM has been downloaded from the web server to the browser. Then the function will be created. This function will set the HTML property of the element with the ID of `textarea_feedback` to "XX characters remaining," where XX is the number of remaining characters from the max length of 141. Within `document.ready`, the keyup event of the element with an ID of `posttext` is modified to include another function that counts the number of remaining characters and displays them in the `textarea_feedback` element. See Listing 16-3.

Listing 16-3. JavaScript to Return the Number of Characters Remaining in the Text Box

```
$(document).ready(function() {
var text_max = 141;
$('#textarea_feedback').html
     (text_max + ' characters remaining');
  $('#posttext').keyup(function() {
  var text_length = $('#posttext').val().length;

    var text_remaining = text_max - text_length;
      $('#textarea_feedback').html(text_remaining + '
        characters remaining');
    });
});
```