

trivia

JavaScript and browser compatibility

Thanks to the efforts of the W3C, HTML and CSS are now run in very similar ways across all of the major browsers. JavaScript, however, still has very bad cross-browser compatibility.

Writing JavaScript that would work in every browser used to require a lot of programming to account for the peculiarities of different browsers.

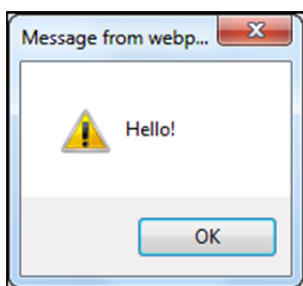
Fortunately, JQuery does all of that for you.

JQuery is a JavaScript library with a lot of useful functions. Most importantly, all of its functions are cross-browser compatible.

If you look in the *scripts* folder of this lesson's project, you'll see the JQuery files.

ASP.NET doesn't automatically generate JQuery code, although it does include the JQuery files with every new Web Application project.

Because ASP.NET doesn't use JQuery you won't cover it in this book, but it is a wonderful resource if you work with JavaScript in the future.



Lesson 2-11: Work with JavaScript

As well as HTML and CSS the remaining language recognized by a web browser is JavaScript. JavaScript is code that can be used to add interactive features to web pages without having to obtain information from the server. For this reason it is called client-side code (in IT terminology a browser is referred to as a "client").

Although you won't need to write any JavaScript in order to create basic ASP.NET sites, it's important that you can recognize it when you see it as ASP.NET will automatically add JavaScript to a lot of pages.

- 1 Open *HTMLTest* from your sample files folder.
- 2 Open *scripttest.aspx* in *Source* view.

You'll see that a link to *scripttest.js* has already been added to this page.

```
<script type="text/javascript" src="/scripts/scripttest.js"></script>
```

- 3 Add inline JavaScript.

In the same way as CSS, you can add JavaScript directly to the page instead of having it in a separate file. But just like with CSS, it is best practice to keep your JavaScript code in separate files.

1. Add the following code to the *head* tag, under the existing *script* tag pair:

```
<script type="text/javascript">
    alert("Hello!");
</script>
```

```
<head runat="server">
    <title></title>
    <link rel="stylesheet" href="/s
    <script type="text/javascript" s
    <script type="text/javascript">
        alert("Hello!");
    </script>
</head>
```

2. View *scripttest.aspx* in your browser (*Design* view isn't capable of executing JavaScript).

You should see your message pop up on the screen. This is one of the most basic JavaScript functions.

3. Close your browser and remove the JavaScript code you just added.

- 4 Assign a JavaScript function to a page element's onclick event.

1. Return to *scripttest.aspx* in *Source* view.
2. Change the code of `<div id="Button">` to:

```
<div id="Button" onclick="ClickMessage()">
```

```
    Click Me!
</div>
```

note

ASP.NET and JavaScript

Many of ASP.NET's special functions are accomplished using JavaScript, but you might never know this as it's all generated automatically behind the scenes.

Although there's a lot more to know about JavaScript, the basic understanding provided by this lesson will enable you to complete and understand all of the lessons in this book.

note

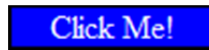
Client-Side and Server-Side

JavaScript is *Client-Side* code. This means it runs on the computer of the person visiting your web site, not on the web server that is hosting your web site.

Because JavaScript is client-side, you can never rely on it running: the visitor might have disabled JavaScript in their browser settings.

The C# code you'll be learning later on is *Server-Side* and can't be directly interfered with by visitors to your site.

- Open the page in your browser.



- Click the *Click Me!* Button.



You'll see a message pop up as before. If you look in *scripttest.js* in the *Scripts* folder, you'll see the JavaScript *function* that caused this to happen.

By adding the function's name to the div's *onclick* property, you've made it run when the div is clicked.

5 Change colors with JavaScript.

- Close your browser and open the *scripttest.js* file. You'll find it in the *scripts* folder of the project.
- Replace the line `alert("Clicked!");` with:

```
document.getElementById("Button")
.setAttribute("style", "background-color: red;");
```

```
function ClickMessage() {
    document.getElementById("Button")
    .setAttribute("style", "background-color: red;");
}
```

This might seem like a complicated piece of code, but if you break it down into its components it's not so difficult.

```
document.getElementById("Button")
```

This tells JavaScript to search the page for an element with an ID of *Button*. The blue *div* on the page is called *Button*, so it will find that.

```
setAttribute("style", "background-color: red;");
```

This tells JavaScript to set the *style* property of the tag to `background-color: red;`. You used the *style* property in *Lesson 2-8: Work with CSS*.

You've already set the *ClickMessage()* function to run when the *div* is clicked. When the user clicks on the *div*, the JavaScript will change the div's HTML code to the following:

```
<div id="Button" style="background-color: red;"
onclick="ClickMessage()">Click Me!</div>
```

- View *scripttest.aspx* in your browser.
- Click on the *Click Me!* div.

The color of the div changes when it is clicked.



6 Save your changes and close Visual Studio.