Section I
JavaScript Basic

JAVASCRIPT HOW TO AND WHERE TO

In This Chapter

- ❑ How To Put a JavaScript into an HTML Page
- ❑ How To Handle Simple Browsers
- ❑ Where To Put the JavaScript
- ❑ Using an External JavaScript
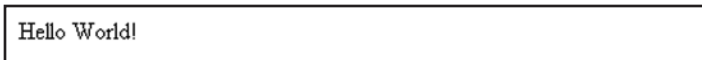
The HTML `<script>` tag is used to insert a JavaScript into an HTML page.

How To Put a JavaScript into an HTML Page

The following example shows how to use JavaScript to write text on a Web page.

The result of this script is shown in Figure 1.1.

```
<html>
<body>
<script type="text/javascript">
document.write("Hello World!");
</script>
</body>
</html>
```

Hello World!

Figure 1.1

Here's your first opportunity to personalize JavaScript. Change the "Hello World" text to "Happy, Happy, Joy, Joy!" and see what happens. The result of your changes is shown in Figure 1.2.

Try it yourself >>

```html
<html>
<body>
<script type="text/javascript">
document.write("Happy, Happy, Joy, Joy!");
</script>
</body>
</html>
```

Happy, Happy, Joy, Joy!

Figure 1.2

The following example shows how to add HTML tags to the JavaScript. The result of this code is shown in Figure 1.3.

```html
<html>
<body>
<script type="text/javascript">
document.write("<h1>Hello World!</h1>");
</script>
</body>
</html>
```

Hello World!

Figure 1.3

Continuing with our happier version of the code, change the "Hello World!" text to "Happy, Happy, Joy, Joy!" and see what happens. The result of your changes is shown in Figure 1.4.

Try it yourself >>

```
<html>
<body>
<script type="text/javascript">
document.write("<h1>Happy, Happy, Joy, Joy!</h1>");
</script>
</body>
</html>
```

Happy, Happy, Joy, Joy!

Figure 1.4

To insert a JavaScript into an HTML page, we use the `<script>` tag. Inside the `<script>` tag we use the type attribute to define the scripting language.

So, `<script type="text/javascript">` and `</script>` tell where the Java-Script starts and ends:

```
<html>
<body>
<script type="text/javascript">
...
</script>
</body>
</html>
```

The `document.write` command is a standard JavaScript command for writing output to a page.

When you type the `document.write` command between the `<script>` and `</script>` tags, the browser will recognize it as a JavaScript command and execute the code line. In this case, the browser writes Hello World! to the page:

```
<html>
<body>
<script type="text/javascript">
document.write("Hello World!");
```

(continued)

(continued)
```
      </script>
      </body>
      </html>
```

> **N O T E** If we had not typed the `<script>` tag, the browser would have treated the `document.write(“Hello World!”)` command as pure text and would just write the entire line on the page, as shown in Figure 1.5.

```
document.write("Hello World!");
```

Figure 1.5

How to Handle Simple Browsers

Browsers that do not support JavaScript will display JavaScript as page content.

To prevent them from doing this and as a part of the JavaScript standard, the HTML comment tag should be used to "hide" the JavaScript.

Just add an HTML comment tag `<!--` before the first JavaScript statement, and an end-of–comment tag `-->` after the last JavaScript statement, like this:

```
<html>
<body>
<script type="text/javascript">
<!--
document.write("Hello World!");
//-->
</script>
</body>
</html>
```

The two forward slashes at the end of comment line (`//`) comprise the JavaScript comment symbol. This prevents JavaScript from executing the `-->` tag.

Where to Put the JavaScript

JavaScripts in a page will be executed immediately while the page loads into the browser. This is not always what we want. Sometimes we want to execute a script when a page loads, or at a later event, such as when a user clicks a button. When this is the case we put the script inside a function. You will learn about functions in Chapter 10, "JavaScript Functions."

Scripts in <head>

Scripts to be executed when they are called, or when an event is triggered, are placed in functions.

Put your functions in the head section. This way they are all in one place, and they do not interfere with page content. The resulting alert box is shown in Figure 1.6.

Try it yourself >>

```
<html>
<head>
<script type="text/javascript">
function message()
{
alert("This alert box was called with the onload event");
}
</script>
</head>

<body onload="message()">
</body>
</html>
```


Figure 1.6

Scripts in <body>

If you don't want your script to be placed inside a function, or if your script should write page content, it should be placed in the body section. Figure 1.7 shows the result.

Try it yourself >>

```
<html>
<head>
</head>

<body>
<script type="text/javascript">
document.write("This message is written by JavaScript");
</script>
</body>

</html>
```

This message is written by JavaScript

Figure 1.7

Scripts in <head> and <body>

You can place an unlimited number of scripts in your document, so you can have scripts in both the body and the head section.

```
<html>
<head>
<script type="text/javascript">
function message()
{
alert("This alert box was called with the onload event");
}
</script></head>
<body onload="message()">
<script type="text/javascript">
document.write("This message is written by JavaScript");
</script>
</body>

</html>
```

Using an External JavaScript

If you want to run the same JavaScript on several pages without having to write the same script on every page, you can write a JavaScript in an external file.

Save the external JavaScript file with a `.js` file extension. Your results are shown in Figure 1.8.

N O T E The external script cannot contain the `<script>` tag!

Try it yourself >>

To use the external script, point to the `.js` file in the `src` attribute of the `<script>` tag as shown:

```
<html>
<head>
<script type="text/javascript" src="xxx.js"></script>
</head>
<body>
</body>
</html>
```

This text was written by an external script!

Figure 1.8

N O T E Remember to place the script exactly where you normally would write the script!