Lesson 1: Writing Your First JavaScript

OBJECTIVES: In this lesson you will be taught how to

- Use the <script> </script> tag
- Insert JavaScript code in a Web page
- Hide your JavaScript code from older browsers
- Use document.write to create output
- Become familiar with built in objects and object notation
- Use embedded HTML to format output
- Display the date and time using the Date object
- Use comments to document code

Preparing to Program

JavaScript code is most often embedded in HTML documents. Since JavaScript code requires the browser to perform special processing, the code is “fenced off” so to speak from the HTML with the <**script**> tag. JavaScript code is placed between <**script**> and </**script**>. JavaScript programs look something like this:

Example

```
<script language="JavaScript">

   JavaScript code goes here

</script>
```

Browsers that recognize JavaScript also recognize the special <script> ... </script> tags. Since there is more than one scripting language, the script tag has a language attribute that you set to “JavaScript.” The script tag can be inserted in the body of your HTML document or the head of your document. If your JavaScript code will write something directly on the page, it is normally placed in the body section of the document where you want the results to appear. Code that does not write to the page directly, such as code that changes elements or defines variables, is normally placed in the head section. For most of the programs in this course you will insert your code in the body of the document.

Hiding JavaScript From Older Browsers

While this has become less important, very old browsers do not understand JavaScript and will not recognize the script tag. The JavaScript code that is typed between the script tags are just lines of text to those old browsers and will be displayed on the Web page.

To prevent this from happening, a **programming convention** is used that hides the JavaScript inside **comments**. It prevents the JavaScript from being displayed. Of course, it does not solve the original problem, that there are browsers unable to process JavaScript. So this fix will hide your code for those browsers; note that they will be unable to execute it.

It looks something like this:

```
<script LANGUAGE=”JavaScript”>

<!--  hide (first comment line)

JavaScript code goes here

// finish hiding (last comment line)-->

</script>
```

A programming convention is a common practice used by most professional programmers. Although it is usually a good idea to comply with programming conventions, you are not required to do so by the rules of the language. Comments are parts of programs that document the code but are not executed. They are preceded by a special symbol that tells the broswer not to interpret the text that follows. To hide your code from old browsers, you insert the opening HTML comment code (<!--) just after the opening script tag. Just before the closing script tag, you insert the closing HTML comment preceded by two forward slashes(// -->). If you use 1st Page 2000 to write your JavaScript code, select the Scripting sub-menu, then select Define Scripting Language and choose JavaScript. The opening and closing script tags along with the needed comment lines will be generated for you by 1st page 2000 (See Appendix B for more detailed instructions on using 1st Page 2000 to write JavaScript code).

Using document.write to Display Text

The document.write method is your first JavaScript statement. It writes a line of text to the document. The format for document.write is as follows:

Syntax:

document.write(“*text to be displayed within quotation marks”*)

document.write will display what it finds between the opening parenthesis and closing parenthesis. The characters surrounded by quotation marks are known as a **string**. When the string is displayed, the quote marks are omitted. The purpose of the quote marks is to serve as **delimiters**: symbols that mark the beginning and end of the string. You may also use single quote marks to delimit a string.

As was discussed in the Introduction, document is a built in object in JavaScript. When using objects, you must use **dot (.) notation**. The dot means “belongs to.” So document.write(“some text”) means “use the write method that belongs to the object known as document to display the text found inside the parentheses.”

Adding Comments to Your Code

You often want to include information about a JavaScript in the file that contains the code. You may want to include your name for the purpose of handing in an assignment. You sometimes want to explain how a particular section of the code works.

Text that is inside a code file but is not intended to be executed is called a comment. In JavaScript there are two ways to indicate a comment. You can create a single line comment by using //, or create a multi-line comment by using /* and */. Here are some examples:

```
// everything on this line after the two slashes is a comment
```

or

```
/*
   everything,
   including what is on this line and below
   is a comment until you encounter the
   closing symbol, which is
*/
```

These comments only work inside the script tags. If you include them in the HTML section they will be displayed on the page.

In the Lab

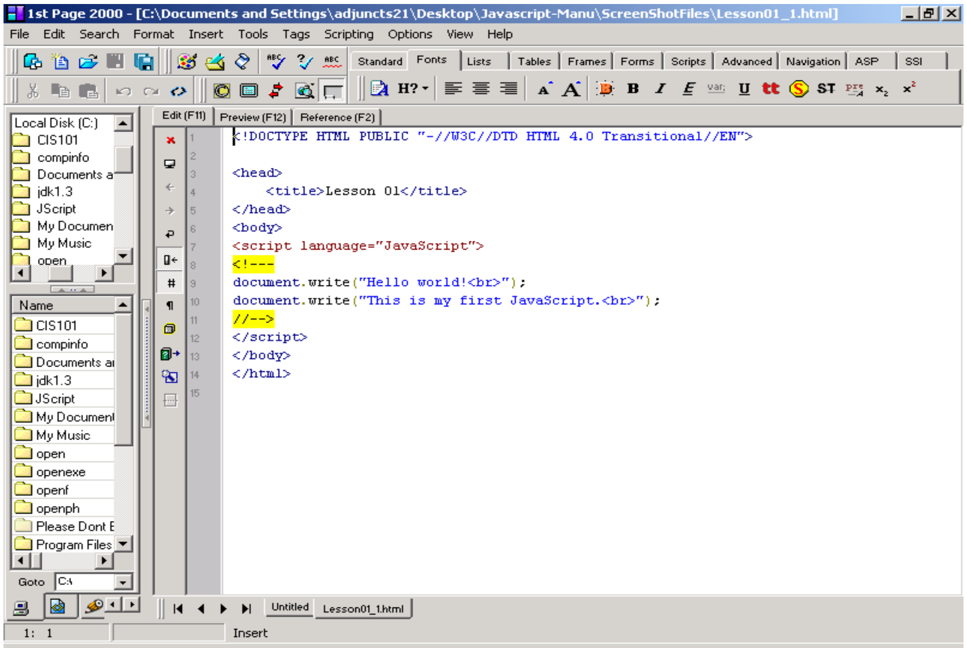
Each week in the lab you will enter JavaScript code and run it using 1st Page 2000. After correctly entering the program and running it without any mistakes, you will start making changes and additions to the code. This is a technique programmers use to learn a new language. Programmers key in a program they know is working, then try to make small adjustments and alterations until they learn enough about the language to write an entire program. Even after learning a language, programmers hardly every start with a blank page. They usually begin with some code sample close to what they want to accomplish, then make alterations.

Start 1st page 2000 and begin a new HTML document. Save it giving it the name lesson0101.html. Following the instructions outlined in Appendix B, place your cursor between the <body> … </body> tags, and insert the script tags and hiding comments by using the Scripting menu.

Now type in ***exactly*** the following code:

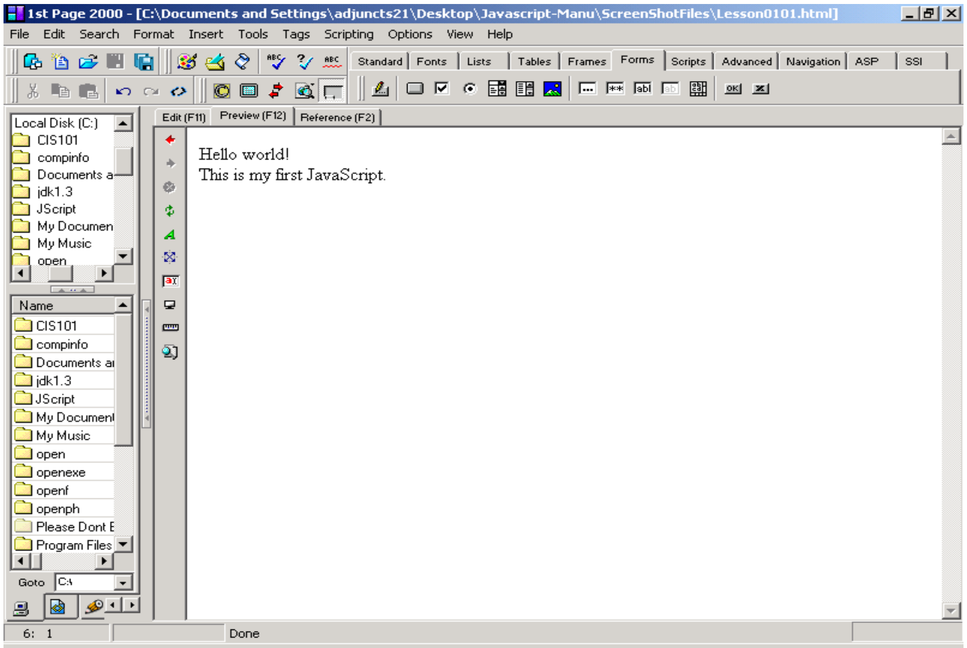
```
<html>

<head>
   <title>Lesson 01</title>
</head>

<body>

<script language="Javascript">

<!–

document.write("Hello world!<br>");

document.write("This is my first JavaScript.<br>");

//-->

</script>

</body>

</html>
```

Please note: if you use 1st Page 2000, it will generate an HTML template plus the script tags and the hiding comments. You actually only have to key in the Title, and lines 10 and 11 (the two document.write statements).


1st Page 2000 - [C:\Documents and Settings\adjuncts21\Desktop\Javascript-Manu\ScreenShotFiles\Lesson01_1.html]
File Edit Search Format Insert Tools Tags Scripting Options View Help
Standard Fonts Lists Tables Frames Forms Scripts Advanced Navigation ASP SSI
Edit (F11) Preview (F12) Reference (F2)
Untitled Lesson01_1.html
Insert

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<head>
    <title>Lesson 01</title>
</head>
<body>
<script language="JavaScript">
<!---
document.write("Hello world!<br>");
document.write("This is my first JavaScript.<br>");
//-->
</script>
</body>
</html>
```

Now click the Preview tab, and you should see the following output:


1st Page 2000 - [C:\Documents and Settings\adjuncts21\Desktop\Javascript-Manu\ScreenShotFiles\Lesson0101.html]
File Edit Search Format Insert Tools Tags Scripting Options View Help
Standard Fonts Lists Tables Frames Forms Scripts Advanced Navigation ASP SSI
Edit (F11) Preview (F12) Reference (F2)
Done

```
Hello world!
This is my first JavaScript.
```

Notice that both lines with document.write end with a semi-colon (;). The semi-colon at the end of a statement is optional. You may also end it with the new line character (by hitting enter). Since other programming and scripting languages end statements with a semicolon, JavaScript also allows you to put a semicolon at the end of your statements. You do not need semicolons at the end of statements, and you will see many examples around the Web of JavaScript code without semi-colons. All you need to do is to start the next statement on a new line.

What to do if things go wrong

If you do not see the exact same output shown above, then you have made a mistake entering the code. An error in code is called a bug. Removing errors is called debugging. When you first learn to program, most of the mistakes you make will be keyboarding errors. You might have forgotten the closing quotes on the message or misspelled document.write. Such errors in the format of statements, be it HTML or JavaScript, are known as syntax errors. For the most part, HTML syntax errors are easy to spot and correct: usually the browser just ignores malformed tags and continues on with the page. In contrast, syntax errors in JavaScript code can sometimes be difficult to identify. JavaScript is case sensitive, so "Big" and "big" will not be treated as the same word. This is NOT the case with HTML, so the combination of HTML and JavaScript in one file, with one being case sensitive and the other not is a large source of confusion.

Most Web browsers attempt to assist you in identifying and correcting JavaScript syntax errors. If you try to run your JavaScript in Internet Explorer and it has a syntax error, a window will automatically pop up identifying the error. Unfortunately, the error messages provided by Internet Explorer are not very descriptive, sometimes providing only the line number where the error occurred.

If you use Netscape, a short error message will appear in the status bar (the bottom panel) on the browser window, although that message may be short-lived and easy to overlook. With Netscape, you can generate error messages by entering "javascript:" (the word javascript followed by a colon) in the Location box. This command will invoke the JavaScript Console, which opens in a separate window. The JavaScript Console will identify the line in the HTML document where the syntax error is believed to occur along with a descriptive error message. Usually, this error message will be very helpful in fixing the underlying error. For more detailed instructions on debugging see Appendix D, "Debugging JavaScript Code."

Building on Your First JavaScript

If you are at this point in the lab, it means you have typed in lesson0101.html, and it is working properly. Do not continue with this section until that is the case.

Let's take a closer look at the first document.write statement:

```
document.write("Hello world!<br>");
```

Notice the <br> tag at the end of the string. If you remember from HTML, <br> forces a new line. It is the equivalent of hitting the enter key. You should find this interesting. It means that not only can you embed JavaScript into HTML, but you can also embed HTML into JavaScript!

Besides <br>, you can use all the other HTML text formatting tags, like headings, the font tags, bold, italic, etc. One important requirement is that the tags must be part of the string. In other words, they must be nested inside the opening and closing quotes that define the string.

Let’s start adding more lines.

After line 11 (after the second document.write statement), add the following (if your text is longer than one line you can break it up as indicated below with the backslash):

```
document.write(“<h1>This first JavaScript was written by \
(your name)</h1>”);
```

You can write in different colors. The next line write text in maroon. Be careful with the quotes within quotes. The outer quotes are double quotes, the inner quotes (around the word maroon) must be single quotes.

```
document.write(“<font color=’maroon’>Have a great day!</font><br>”);
```

Try writing extra lines using different colors. For more colors to use, see Appendix C, “Named Colors.”

Using the Date Object

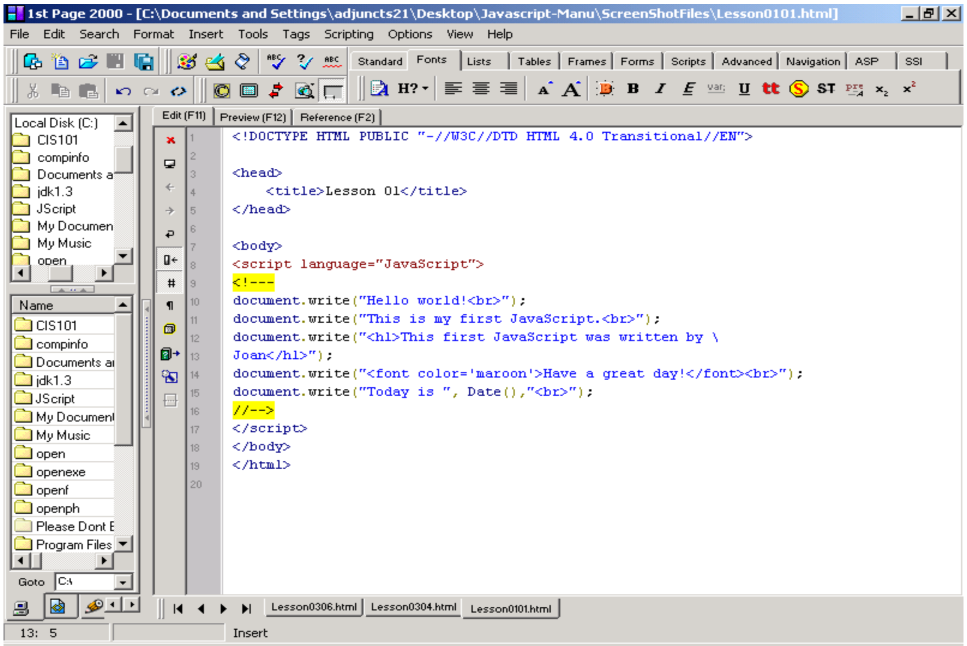
So far we haven’t done anything you couldn’t already do with HTML. We just used a different way to display text. Now it is time to spread our wings and to start seeing a little bit of what JavaScript and programming can provide to you. You are probably aware that whenever you turn on your computer, a little clock pops up in the lower right corner showing the current time. If you place your mouse arrow over the time, today’s date displays. All computers have an internal clock for the day and time. It is needed by the computer for many purposes, including executing code and saving files. With JavaScript, you can capture the current date and time and display it on your page.

Add the following code:

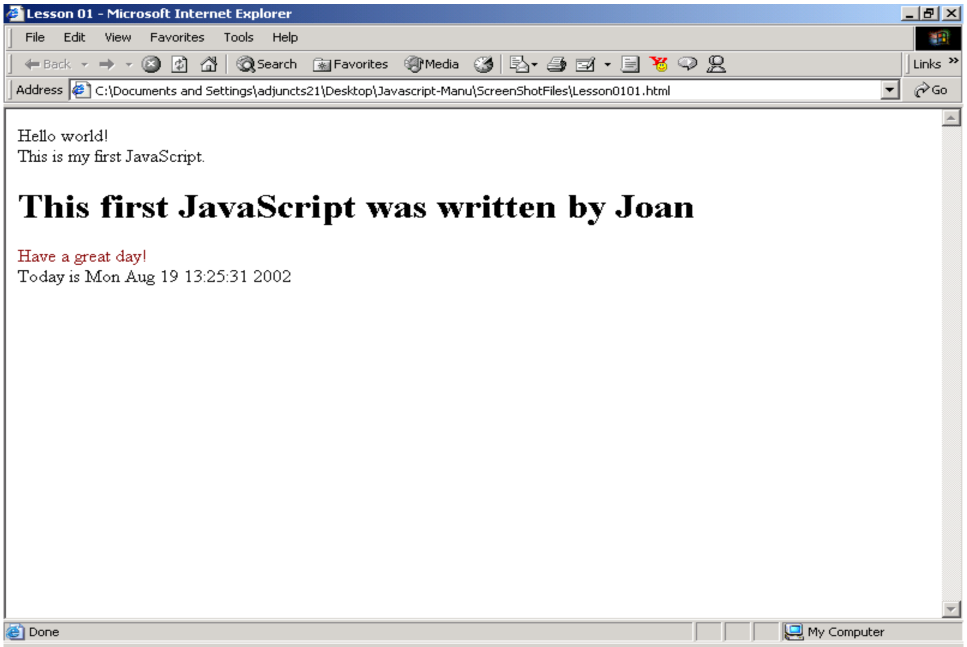
```
document.write("Today is ", Date(),"<br>");
```

Be careful with the syntax of this statement. Be careful, Date() needs to be capitalized. It will not work if you type it all lower case.

Notice the output has three parts, separated by commas. Type it in exactly as you see it and run it. Your code should display something like this:


1st Page 2000 - [C:\Documents and Settings\adjuncts21\Desktop\Javascript-Manu\ScreenShotFiles\Lesson0101.html]
File Edit Search Format Insert Tools Tags Scripting Options View Help
Standard Fonts Lists Tables Frames Forms Scripts Advanced Navigation ASP SSI
Edit (F11) Preview (F12) Reference (F2)
Lesson0306.html Lesson0304.html Lesson0101.html
13: 5 Insert

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<head>
    <title>Lesson 01</title>
</head>

<body>
<script language="JavaScript">
<!---
document.write("Hello world!<br>");
document.write("This is my first JavaScript.<br>");
document.write("<h1>This first JavaScript was written by \
Joan</h1>");
document.write("<font color='maroon'>Have a great day!</font><br>");
document.write("Today is ", Date(),"<br>");
//-->
</script>
</body>
</html>
```


Lesson 01 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media Links
Address C:\Documents and Settings\adjuncts21\Desktop\Javascript-Manu\ScreenShotFiles\Lesson0101.html Go
Done My Computer

```
Hello world!
This is my first JavaScript.

This first JavaScript was written by Joan

Have a great day!
Today is Mon Aug 19 13:25:31 2002
```

Hit the reload button of your browser or hit the edit then preview button in 1st Page 2000, and notice that the time is updated. **Date** is an object that gets the current time and date from your computer's system clock. So each time you reload the page and re-execute that line of code, a new Date object is created.

The syntax of document.write allows you to display a list of things, as long as they are separated by commas. You can keep adding to the document.write statement, as long as each part is separated by a comma. With the comma, the parts are displayed next to each other. If you remove the comma, you will get an error.

Another interesting fact is that the Date object appears differently depending on whether you use Netscape or Internet Explorer. Execute this code using both browsers and notice the differences between the two. This is just a minor example of the compatibility conflicts that exist between Netscape and Internet Explorer. Until this issue is resolved, there will continue to be examples of JavaScript code that do not run the same way with Netscape and Internet Explorer.

Student Modifications

You now have a working JavaScript that uses document.write to display text, uses embedded HTML to format text, and displays the current time and date. Add some more modifications:

- Output a line of text telling us your favorite singer or band
- Output your email address in your favorite color
- Output your favorite movie
- Add a comment within the script tags by using either // or /* and */

Key Terms and Definitions

- **script tag** – Set of tags (<script> and </script> that enclose JavaScript code.
- **programming conventions** – standards and practices that are followed by professional programmers but are not required by the syntax rules of a language.
- **comments** – text that is part of a program file but is not interpreted by the browser. Its purpose is to provide documentation for the code. Comments are indicated in JavaScript by either // or /* and */.
- **document.write** – JavaScript statement that writes text to a page.
- **string** – a word, sentence, or other set of characters, such as letters, numbers, and puctuation marks, surrounded by quote marks.
- **dot notation** - syntax required when referring to objects. The dot (.) indicates that what follows is part of the object. So document.write means that write is a method that belongs to the object named document.
- **delimiters** - Symbols that act as boundaries for parts of a program. Single and double quotes are both used as delimiters for strings.
- **debugging** – the process of elimating errors from a program.
- **Date object** - an object available to a JavaScript that contains a specifc time and date.

Lesson Summary

In Lesson 1 you learned to write your first JavaScript. You used the script tag to designate JavaScript code within an HTML document, and learned how to hide JavaScript from old browsers. You used document.write to display text. You also used HTML embedded within JavaScript to format text for display. You learned how to use comments to document your code. Finally, you added the Date object to your page to display the current date and time.

Lesson 01 Exercises

1_1. Write a JavaScript that displays the following double spaced lines:

Hello again!

Your class is CIS101.

What is your name?

1_2. Write a JavaScript that uses asterisks to display your first initial in block form. For example, the program that Fred writes would output the following:

```
*******
*
*
****
*
*
*
```

1_3. Write a JavaScript program that lists all the courses you are taking this semester. Use a new line and a new color for each course.