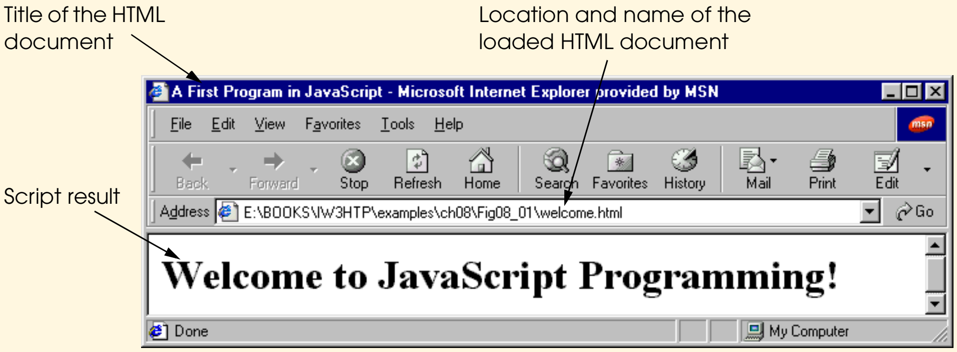
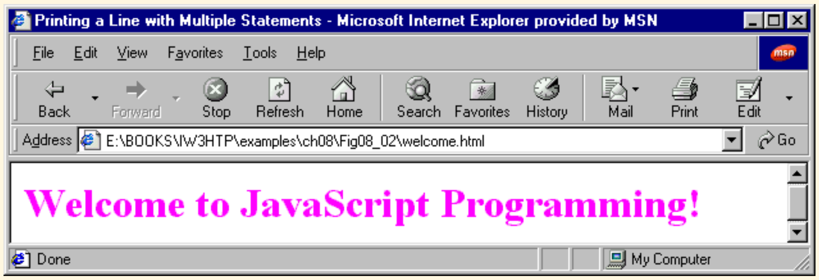
```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<!-- Fig. 8.1: welcome.html -->

<HTML>
<HEAD>
<TITLE>A First Program in JavaScript</TITLE>

<SCRIPT LANGUAGE = "JavaScript">
   document.writeln(
      "<H1>Welcome to JavaScript Programming!</H1>" );
</SCRIPT>

</HEAD><BODY></BODY>
</HTML>
```


Title of the HTML document
Location and name of the loaded HTML document
Script result
A First Program in JavaScript - Microsoft Internet Explorer provided by MSN
File
Edit
View
Favorites
Tools
Help
msn
Back
Forward
Stop
Refresh
Home
Search
Favorites
History
Mail
Print
Edit
Address
E:\BOOKS\IW3HTP\examples\ch08\Fig08_01\welcome.html
Go
Done
My Computer

Welcome to JavaScript Programming!

Fig. 8.1 A first program in JavaScript.

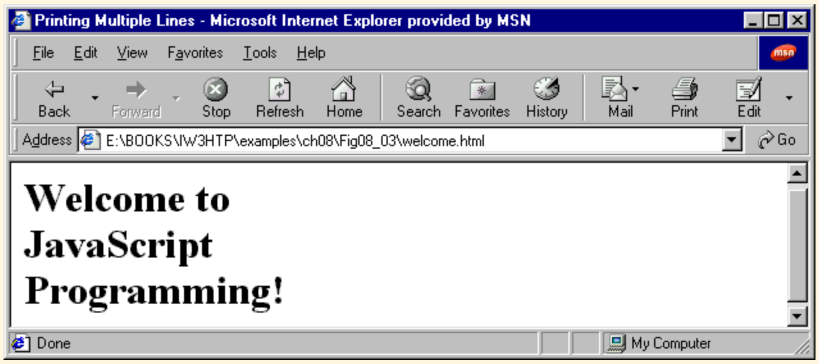
```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<!-- Fig. 8.2: welcome.html -->

<HEAD>
<TITLE>Printing a Line with Multiple Statements</TITLE>

<SCRIPT LANGUAGE = "JavaScript">
   document.write( "<FONT COLOR='magenta'><H1>Welcome to " );
   document.writeln( "JavaScript Programming!</H1></FONT>" );
</SCRIPT>

</HEAD><BODY></BODY>
</HTML>
```


Printing a Line with Multiple Statements - Microsoft Internet Explorer provided by MSN
File Edit View Favorites Tools Help
Back
Forward
Stop
Refresh
Home
Search
Favorites
History
Mail
Print
Edit
Address
E:\BOOKS\IW3HTP\examples\ch08\Fig08_02\welcome.html
Go
Done
My Computer

Welcome to JavaScript Programming!

Fig. 8.2 Printing on one line with separate statements.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<!-- Fig. 8.3: welcome.html -->

<HEAD><TITLE>Printing Multiple Lines</TITLE>

<SCRIPT LANGUAGE = "JavaScript">
   document.writeln(
      "<H1>Welcome to<BR>JavaScript<BR>Programming!</H1>" );
</SCRIPT>

</HEAD><BODY></BODY>
</HTML>
```



Printing Multiple Lines - Microsoft Internet Explorer provided by MSN

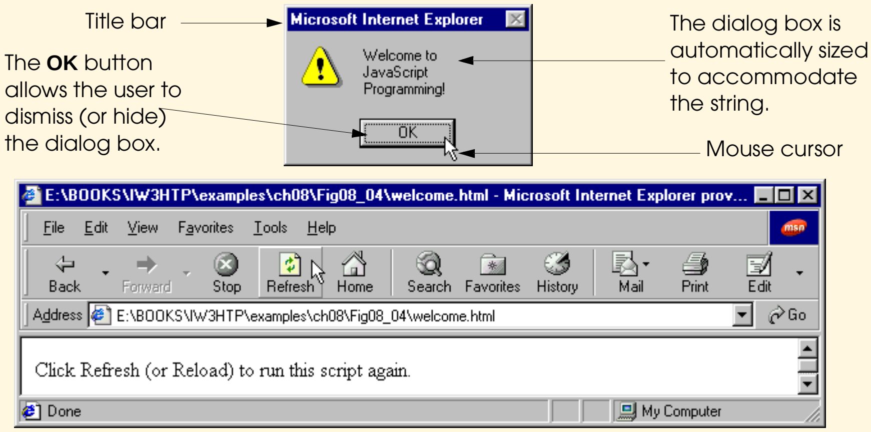
Address E:\BOOKS\IW3HTP\examples\ch08\Fig08_03\welcome.html

Welcome to
JavaScript
Programming!

Done

Fig. 8.3 Printing on multiple lines with a single statement.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<!-- Fig. 8.4: welcome.html -->
<!-- Printing multiple lines in a dialog box -->

<HEAD>

<SCRIPT LANGUAGE = "JavaScript">
   window.alert( "Welcome to\nJavaScript\nProgramming!" );
</SCRIPT>

</HEAD>

<BODY>
<P>Click Refresh (or Reload) to run this script again.</P>
</BODY>
</HTML>
```


Title bar
Microsoft Internet Explorer
The **OK** button allows the user to dismiss (or hide) the dialog box.
OK
The dialog box is automatically sized to accommodate the string.
Mouse cursor
E:\BOOKS\IW3HTP\examples\ch08\Fig08_04\welcome.html - Microsoft Internet Explorer prov...
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit
Address E:\BOOKS\IW3HTP\examples\ch08\Fig08_04\welcome.html
Go
Done
My Computer

Welcome to
JavaScript
Programming!

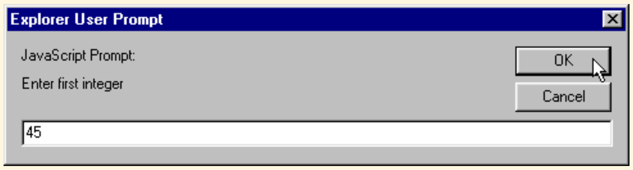
Click Refresh (or Reload) to run this script again.

Fig. 8.4 Displaying multiple lines in a dialog box.

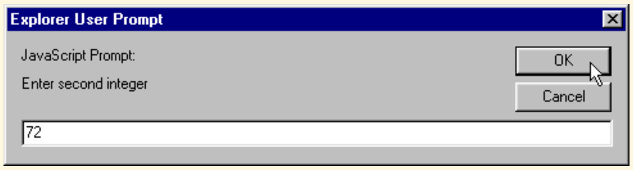
Escape sequence	Description
`\n`	Newline. Position the screen cursor to the beginning of the next line.
`\t`	Horizontal tab. Move the screen cursor to the next tab stop.
`\r`	Carriage return. Position the screen cursor to the beginning of the current line; do not advance to the next line. Any characters output after the carriage return overwrite the previous characters output on that line.
`\\`	Backslash. Used to represent a backslash character in a string.
`\"`	Double quote. Used to represent a double quote character in a string contained in double quotes. For example, `window.alert( "\"in quotes\"" );` displays **`"in quotes"`** in an **`alert`** dialog.
`\’`	Single quote. Used to represent a single quote character in a string. For example, `window.alert( ’\’in quotes\’’ );` displays **`’in quotes’`** in an **`alert`** dialog.

Fig. 8.5 Some common escape sequences.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<!-- Fig. 8.6: Addition.html -->

<HEAD>
<TITLE>An Addition Program</TITLE>

<SCRIPT LANGUAGE = "JavaScript">
   var firstNumber,   // first string entered by user
       secondNumber,  // second string entered by user
       number1,       // first number to add
       number2,       // second number to add
       sum;           // sum of number1 and number2

   // read in first number from user as a string
   firstNumber = window.prompt( "Enter first integer", "0" );

   // read in second number from user as a string
   secondNumber = window.prompt( "Enter second integer", "0" );

   // convert numbers from strings to integers
   number1 = parseInt( firstNumber );
   number2 = parseInt( secondNumber );

   // add the numbers
   sum = number1 + number2;

   // display the results
   document.writeln( "<H1>The sum is " + sum + "</H1>" );
</SCRIPT>

</HEAD>
<BODY>
<P>Click Refresh (or Reload) to run the script again</P>
</BODY>
</HTML>
```

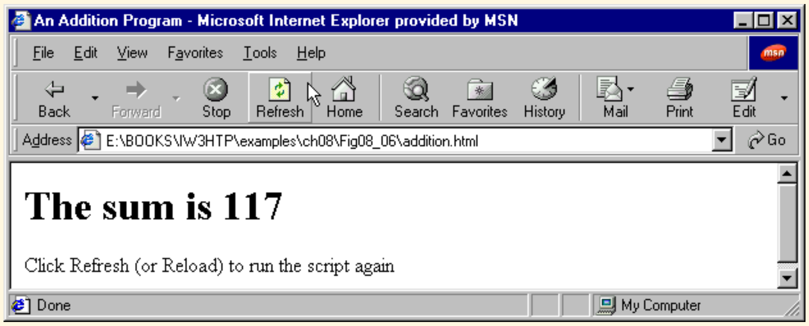


JavaScript Prompt:
Enter first integer
45



JavaScript Prompt:
Enter second integer
72

Fig. 8.6 An addition script "in action" (part 1 of 2).


An Addition Program - Microsoft Internet Explorer provided by MSN
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit
Address E:\BOOKS\IW3HTP\examples\ch08\Fig08_06\addition.html
Go
Done
My Computer

The sum is 117

Click Refresh (or Reload) to run the script again

Fig. 8.6 An addition script "in action" (part 2 of 2).

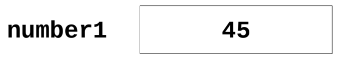


Fig. 8.7 Memory location showing the name and value of variable **`number1`**.

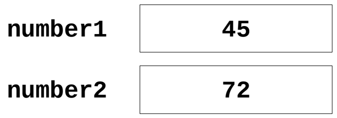


Fig. 8.8 Memory locations after values for variables **`number1`** and **`number2`** have been input.

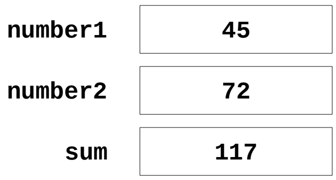


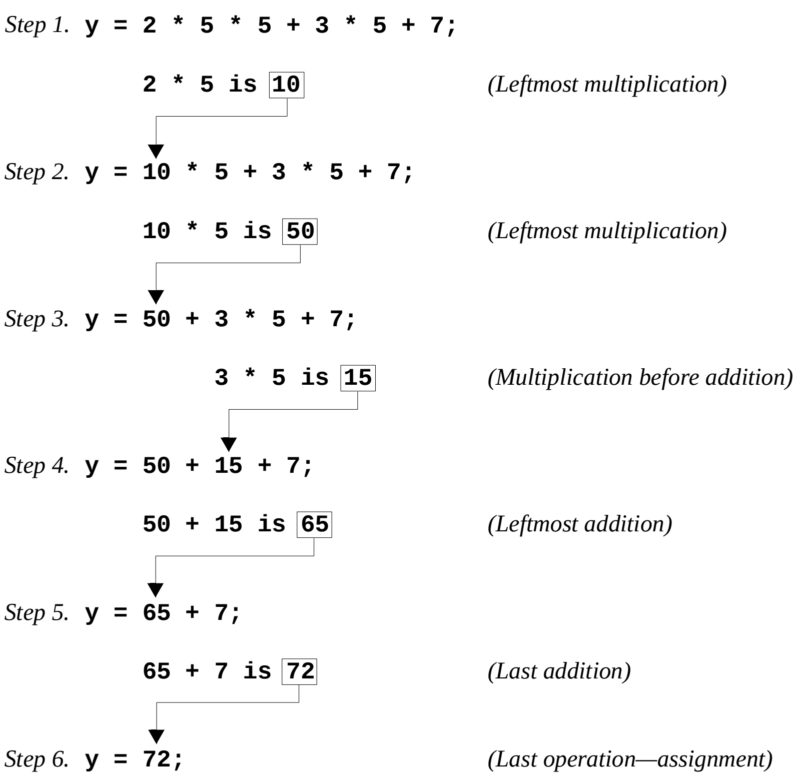
Fig. 8.9 Memory locations after a calculation.

JavaScript operation	Arithmetic operator	Algebraic expression	JavaScript expression
Addition	`+`	$f + 7$	`f + 7`
Subtraction	`-`	$p - c$	`p - c`
Multiplication	`*`	bm	`b * m`
Division	`/`	x / y or $\frac{x}{y}$ or $x \div y$	`x / y`
Modulus	`%`	$r \bmod s$	`r % s`

Fig. 8.10 Arithmetic operators.

Operator(s)	Operation(s)	Order of evaluation (precedence)
( )	Parentheses	Evaluated first. If the parentheses are nested, the expression in the innermost pair is evaluated first. If there are several pairs of parentheses "on the same level" (not nested), they are evaluated left to right.
*****, **/** or **%**	Multiplication Division Modulus	Evaluated second. If there are several, they are evaluated left to right.
+ or **-**	Addition Subtraction	Evaluated last. If there are several, they are evaluated left to right.

Fig. 8.11 Precedence of arithmetic operators.



Step 1. `y = 2 * 5 * 5 + 3 * 5 + 7;`

`2 * 5 is 10` *(Leftmost multiplication)*

Step 2. `y = 10 * 5 + 3 * 5 + 7;`

`10 * 5 is 50` *(Leftmost multiplication)*

Step 3. `y = 50 + 3 * 5 + 7;`

`3 * 5 is 15` *(Multiplication before addition)*

Step 4. `y = 50 + 15 + 7;`

`50 + 15 is 65` *(Leftmost addition)*

Step 5. `y = 65 + 7;`

`65 + 7 is 72` *(Last addition)*

Step 6. `y = 72;` *(Last operation—assignment)*

Fig. 8.12 Order in which a second-degree polynomial is evaluated.

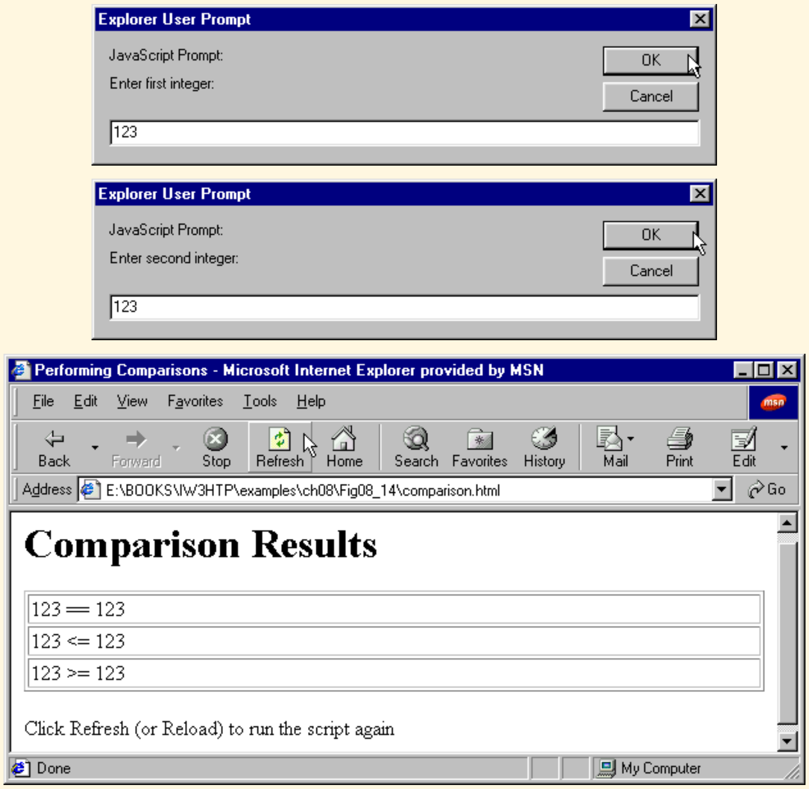
Standard algebraic equality operator or relational operator	JavaScript equality or relational operator	Sample JavaScript condition	Meaning of JavaScript condition
Equality operators			
=	`==`	`x == y`	`x` is equal to `y`
≠	`!=`	`x != y`	`x` is not equal to `y`
Relational operators			
>	`>`	`x > y`	`x` is greater than `y`
<	`<`	`x < y`	`x` is less than `y`
≥	`>=`	`x >= y`	`x` is greater than or equal to `y`
≤	`<=`	`x <= y`	`x` is less than or equal to `y`

Fig. 8.13 Equality and relational operators.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<!-- Fig. 8.14: comparison.html -->
<!-- Using if statements, relational operators, -->
<!-- and equality operators -->

<HEAD>
<TITLE>Performing Comparisons</TITLE>

<SCRIPT LANGUAGE = "JavaScript">
   var first,   // first string entered by user
       second;  // second string entered by user

   // read first number from user as a string
   first = window.prompt( "Enter first integer:", "0" );

   // read second number from user as a string
   second = window.prompt( "Enter second integer:", "0" );

   document.writeln( "<H1>Comparison Results</H1>" );
   document.writeln( "<TABLE BORDER = '1' WIDTH = '100%'>" );

   if ( first == second )
      document.writeln( "<TR><TD>" + first + " == " + second +
                        "</TD></TR>" );

   if ( first != second )
      document.writeln( "<TR><TD>" + first + " != " + second +
                        "</TD></TR>" );

   if ( first < second )
      document.writeln( "<TR><TD>" + first + " < " + second +
                        "</TD></TR>" );

   if ( first > second )
      document.writeln( "<TR><TD>" + first + " > " + second +
                        "</TD></TR>" );

   if ( first <= second )
      document.writeln( "<TR><TD>" + first + " <= " + second +
                        "</TD></TR>" );

   if ( first >= second )
      document.writeln( "<TR><TD>" + first + " >= " + second +
                        "</TD></TR>" );

   // Display results
   document.writeln( "</TABLE>" );
</SCRIPT>

</HEAD>
```

Fig. 8.14 Using equality and relational operators.

```
<BODY>
<P>Click Refresh (or Reload) to run the script again</P>
</BODY>
</HTML>
```



Explorer User Prompt

JavaScript Prompt:

Enter first integer:

123

Explorer User Prompt

JavaScript Prompt:

Enter second integer:

123

Performing Comparisons - Microsoft Internet Explorer provided by MSN

Address E:\BOOKS\IW3HTP\examples\ch08\Fig08_14\comparison.html

Comparison Results

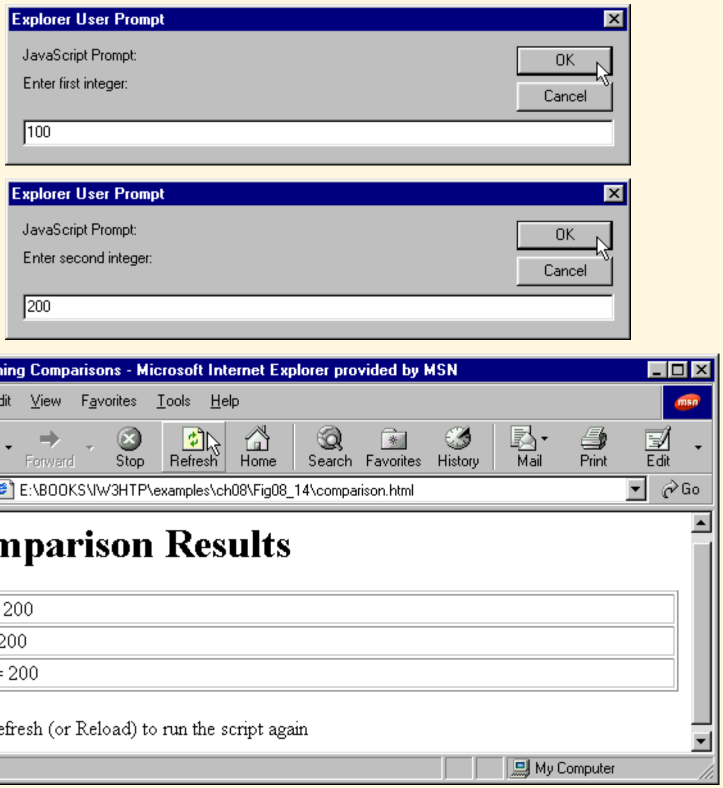
123 == 123
123 <= 123
123 >= 123

Click Refresh (or Reload) to run the script again

Done

My Computer

Fig. 8.14 Using equality and relational operators.


Explorer User Prompt
OK
Cancel
Explorer User Prompt
OK
Cancel
Performing Comparisons - Microsoft Internet Explorer provided by MSN
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit
Address
Go
Done
My Computer

JavaScript Prompt:
Enter first integer:

100

JavaScript Prompt:
Enter second integer:

200

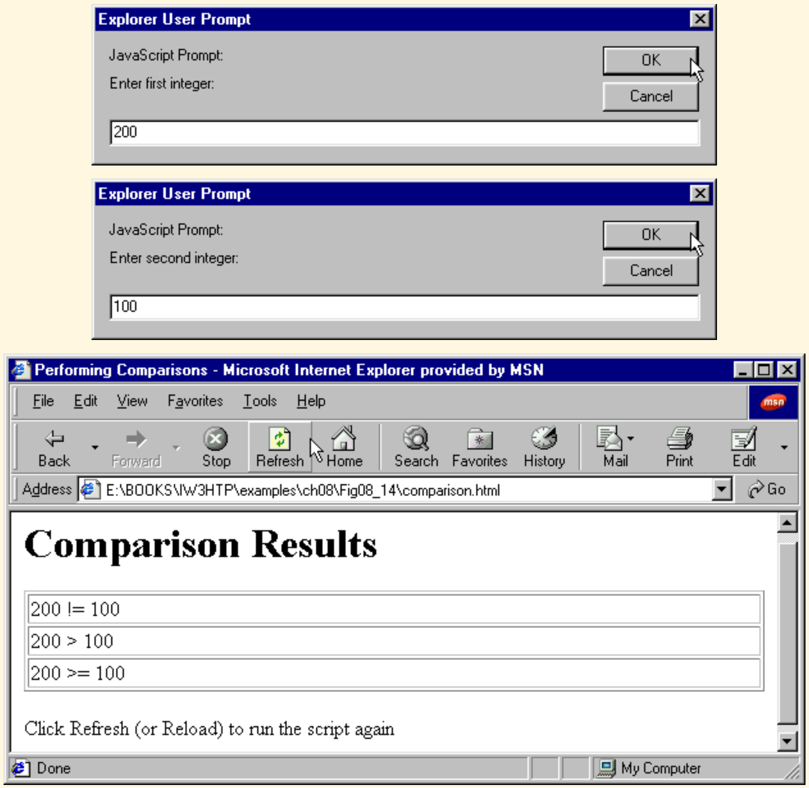
E:\BOOKS\IW3HTP\examples\ch08\Fig08_14\comparison.html

Comparison Results

100 != 200
100 < 200
100 <= 200

Click Refresh (or Reload) to run the script again

Fig. 8.14 Using equality and relational operators.



Explorer User Prompt

JavaScript Prompt:

Enter first integer:

200

Explorer User Prompt

JavaScript Prompt:

Enter second integer:

100

Performing Comparisons - Microsoft Internet Explorer provided by MSN

Address: E:\BOOKS\IW3HTP\examples\ch08\Fig08_14\comparison.html

Comparison Results

200 != 100
200 > 100
200 >= 100

Click Refresh (or Reload) to run the script again

Fig. 8.14 Using equality and relational operators.

Operators	Associativity	Type
`()`	left to right	parentheses
`* / %`	left to right	multiplicative
`+ -`	left to right	additive
`< <= > >=`	left to right	relational
`== !=`	left to right	equality
`=`	right to left	assignment

Fig. 8.15 Precedence and associativity of the operators discussed so far.