

Chapter 7 - JavaScript: Introduction to Scripting

1

Outline

- 7.1 Introduction
- 7.2 Simple Program: Printing a Line of Text in a Web Page
- 7.3 Another JavaScript Program: Adding Integers
- 7.4 Memory Concepts
- 7.5 Arithmetic
- 7.6 Decision Making: Equality and Relational Operators
- 7.7 JavaScript Internet and World Wide Web Resources

© 2001 Prentice Hall, Inc. All rights reserved.



2

Outline

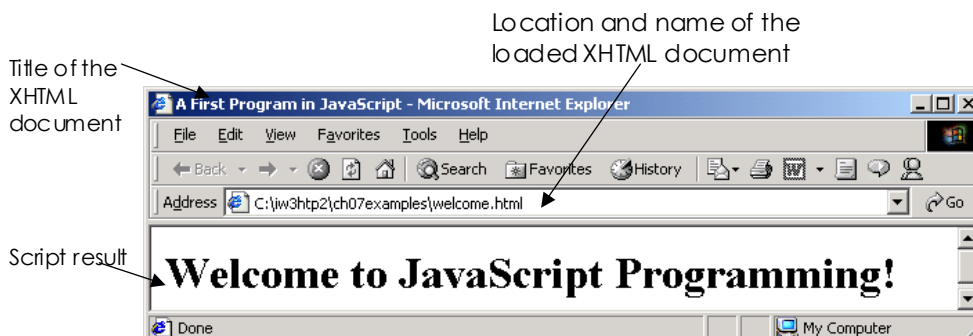


welcome.html

```
1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5  <!-- Fig. 7.1: welcome
6  <!-- Displaying a line
7
8  <html xmlns = "http://www.w3.org/
9    <head>
10     <title>A First Program in JavaScript</title>
11
12     <script type = "text/javascript">
13       <!--
14       document.writeln(
15         "<h1>Welcome to JavaScript Programming!</h1>" );
16       // -->
17     </script>
18
19   </head><body></body>
20 </html>
```

The **script** tag indicates to the browser that the text which follows is part of a script.

The **document** object's **writeln** method writes a line of XHTML markup in the XHTML document.



Program Output

© 2001 Prentice Hall, Inc.
All rights reserved.



Outline

Welcome2.html

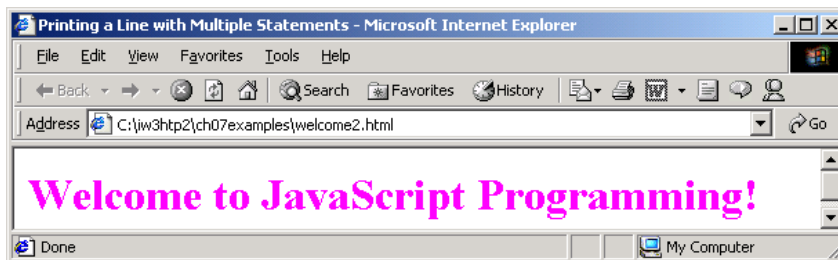
```

1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5  <!-- Fig. 7.2: welcome2.html -->
6  <!-- Printing a Line with Multiple Statements -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml" -->
9    <head>
10     <title>Printing a Line with Multiple Statements</title>
11
12     <script type = "text/javascript">
13       <!--
14       document.write( "<h1 style = \"color: magenta\">" );
15       document.write( "Welcome to JavaScript " +
16         "Programming!</h1>" );
17       // -->
18     </script>
19
20   </head><body></body>
21 </html>

```

Using the **style** attribute, the color of the text is changed from black to magenta.

The escape sequence `\` places a quote in the string and is not displayed in the browser.



Program Output

© 2001 Prentice Hall, Inc.
All rights reserved.



Outline

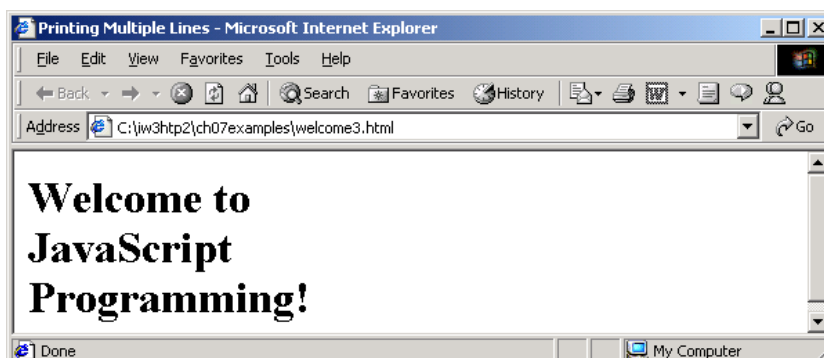
Welcome3.html

```

1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5  <!-- Fig. 7.3: welcome3.html -->
6  <!-- Printing Multiple Lines -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml" -->
9    <head><title>Printing Multiple Lines</title>
10
11     <script type = "text/javascript">
12       <!--
13       document.writeln( "<h1>Welcome to<br />JavaScript" +
14         "<br />Programming!</h1>" );
15       // -->
16     </script>
17
18   </head><body></body>
19 </html>

```

Using **break** tags, the text is displayed as three lines.



Program Output

© 2001 Prentice Hall, Inc.
All rights reserved.



```

1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5  <!-- Fig. 7.4: welcome4.html -->
6  <!-- Printing multiple lines in a dialog box -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9    <head><title>Printing Multiple Lines in a Dialog Box</title>
10
11    <script type = "text/javascript">
12      <!--
13        window.alert( "Welcome to\nJavaScript\nProgramming!" );
14      // -->
15    </script>
16
17  </head>
18
19  <body>
20    <p>Click Refresh</p>
21  </body>
22 </html>

```

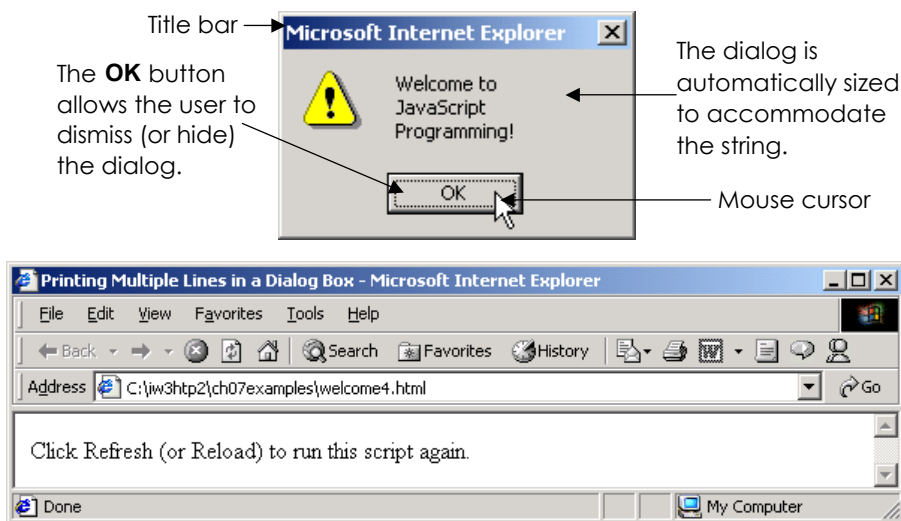
The window method `alert` displays an alert dialog to the user.

When the alert dialog displays, the string passed as its one argument is displayed.

The escape sequence `\n` is the newline character that places all remaining text on the next line.



Program Output



7.2 Simple Program: Printing a Line of Text in a Web Page

Escape sequence	Description
<code>\n</code>	Newline. Position the screen cursor at the beginning of the next line.
<code>\t</code>	Horizontal tab. Move the screen cursor to the next tab stop.
<code>\r</code>	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 characters previously output on that line.
<code>\\</code>	Backslash. Used to represent a backslash character in a string.
<code>\"</code>	Double quote. Used to represent a double quote character in a string contained in double quotes. For example, <code>window.alert("\"in quotes\"");</code> displays "in quotes" in an alert dialog.
<code>\'</code>	Single quote. Used to represent a single quote character in a string. For example, <code>window.alert('\''in quotes'\'');</code> displays 'in quotes' in an alert dialog.

Fig. 7.5 Some common escape sequences.



```

1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5  <!-- Fig. 7.6: Addition.html -->
6  <!-- Addition Program -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9    <head>
10     <title>An Addition Program</title>
11
12     <script type = "text/javascript">
13       <!--
14       var firstNumber,    // first s
15         secondNumber,    // second
16         number1,         // first
17         number2,         // second
18         sum;             // sum of
19
20       // read in first number from user as a string
21       firstNumber =
22         window.prompt( "Enter first integer", "0" );
23
24       // read in second number from user as a string
25       secondNumber =
26         window.prompt( "Enter second integer", "0" );
27
28       // convert numbers from strings to integers
29       number1 = parseInt( firstNumber );
30       number2 = parseInt( secondNumber );
31
32       // add the numbers
33       sum = number1 + number2;
34

```



Outline

Addition.html

The **window** method **prompt** displays a prompt dialog in the browser with a message and a text field for input.

The first argument passed to method

prompt is the message.

The second argument is the default value for the text field.

Function **parseInt** converts its string argument to an integer.

The **+** operator adds the two numbers input by the user.

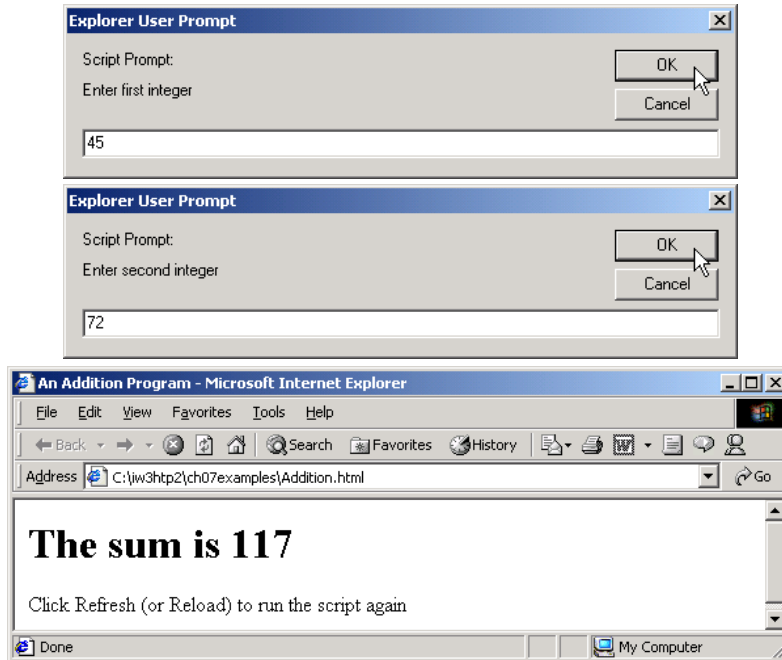


```

35      // display the results
36      document.writeln( "<h1>The sum is " + sum + "</h1>" );
37      // -->
38      </script>
39
40  </head>
41  <body>
42      <p>Click Refresh (or Reload) to run the script again</p>
43  </body>
44  </html>

```

Program Output



© 2001 Prentice Hall, Inc.
All rights reserved.

7.3 Another JavaScript Program: Adding Integers

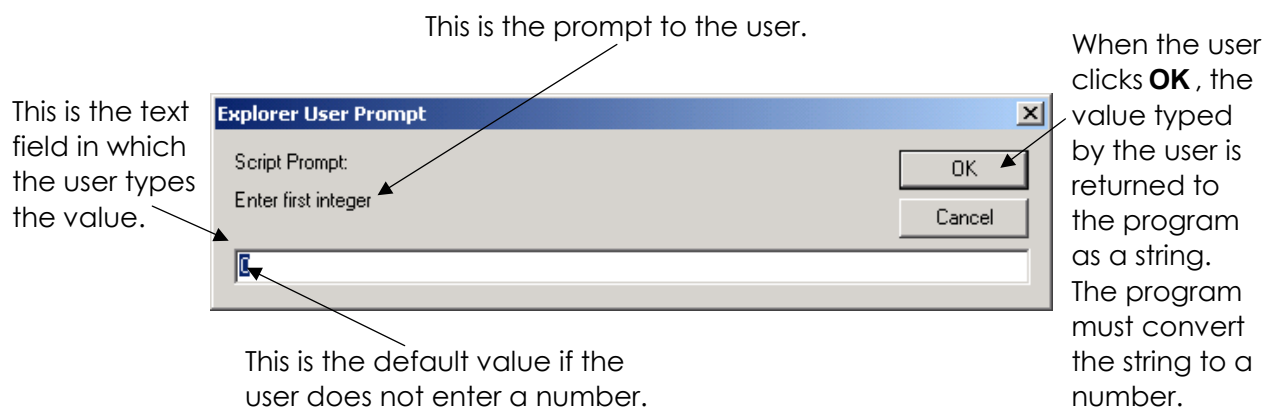


Fig. 7.7 Prompt dialog displayed by the window object's prompt method.



7.4 Memory Concepts

number1

45

Fig. 7.8 Memory location showing the name and value of variable number1.

number1

45

number2

72

Fig. 7.9 Memory locations after values for variables number1 and number2 have been input.



7.5 Arithmetic

number1

45

number2

72

sum

117

Fig. 7.10 Memory locations after calculating the sum of number1 and number2.



7.5 Arithmetic

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 $\langle \text{Anchor9} \rangle$ or $x \ y$	x / y
Modulus	%	$r \bmod s$	$r \% s$

Fig. 7.11 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 (i.e., not nested), they are evaluated from left to right.
*, / or %	Multiplication Division Modulus	Evaluated second. If there are several such operations, they are evaluated from left to right.
+ or -	Addition Subtraction	Evaluated last. If there are several such operations, they are evaluated from left to right.

Fig. 7.12 Precedence of arithmetic operators.



7.6 Decision Making: Equality and Relational Operators

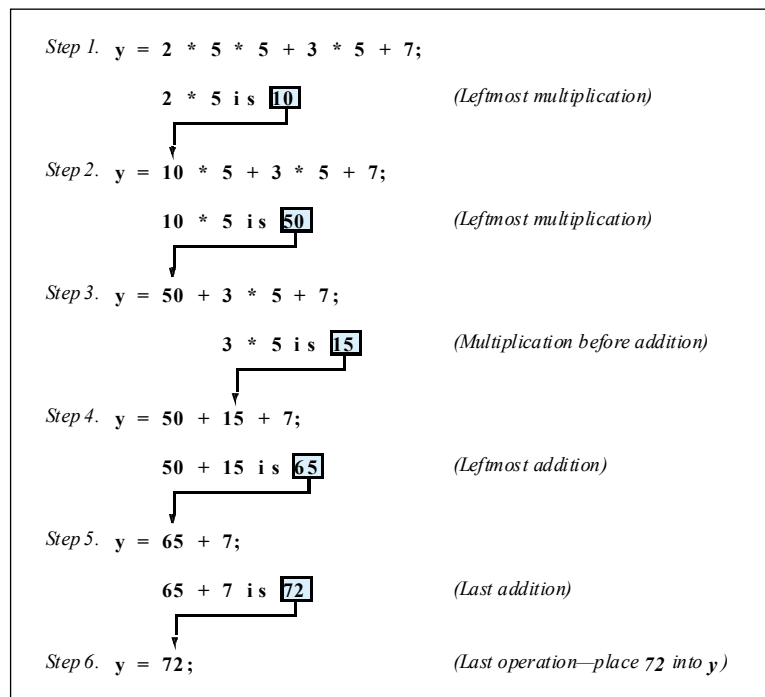


Fig. 7.13 Order in which a second-degree polynomial is evaluated.



7.6 Decision Making: Equality and Relational Operators

Standard algebraic equality operator or relational operator	JavaScript equality or relational operator	Sample JavaScript condition	Meaning of JavaScript condition
<i>Equality operators</i>			
=	==	x == y	x is equal to y
	!=	x != y	x is not equal to y
<i>Relational operators</i>			
>	>	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. 7.14 Equality and relational operators.



```

1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5  <!-- Fig. 7.14: comparison.html          -->
6  <!-- Using if statements, relational operators -->
7  <!-- and equality operators             -->
8
9  <html xmlns = "http://www.w3.org/1999/xhtml">
10   <head>
11     <title>Performing Comparisons</title>
12
13     <script type = "text/javascript">
14       <!--
15       var first,    // first string entered by user
16           second,  // second string entered by user
17           number1, // first number entered by user
18           number2; // second number entered by user
19
20       // read first number from user as a string
21       first = window.prompt( "Enter first integer:", "0" );
22
23       // read second number from user as a string
24       second = window.prompt( "Enter second integer:", "0" );
25
26       // convert numbers from strings to integers
27       number1 = parseInt( first );
28       number2 = parseInt( second );
29
30       document.writeln( "<h1>Comparison Results</h1>" );
31       document.writeln(
32         "<table border = \"1\" width = \"100%\">" );
33

```

Two prompt dialogs retrieve user input.



```
34     if ( number1 == number2 )
35         document.writeln( "<tr><td>" + number1 + " == " +
36             number2 + "</td></tr>" );
37
38     if ( number1 != number2 )
39         document.writeln( "<tr><td>" + number1 + " != " +
40             number2 + "</td></tr>" );
41
42     if ( number1 < number2 )
43         document.writeln( "<tr><td>" + number1 + " < " +
44             number2 + "</td></tr>" );
45
46     if ( number1 > number2 )
47         document.writeln( "<tr><td>" + number1 + " > " +
48             number2 + "</td></tr>" );
49
50     if ( number1 <= number2 )
51         document.writeln( "<tr><td>" + number1 + " <= " +
52             number2 + "</td></tr>" );
53
54     if ( number1 >= number2 )
55         document.writeln( "<tr><td>" + number1 + " >= " +
56             number2 + "</td></tr>" );
57
58     // Display results
59     document.writeln( "</table>" );
60     // -->
61 </script>
62
63 </head>
64 <body>
65     <p>Click Refresh (or Reload) to run the script again</p>
66 </body>
67 </html>
```

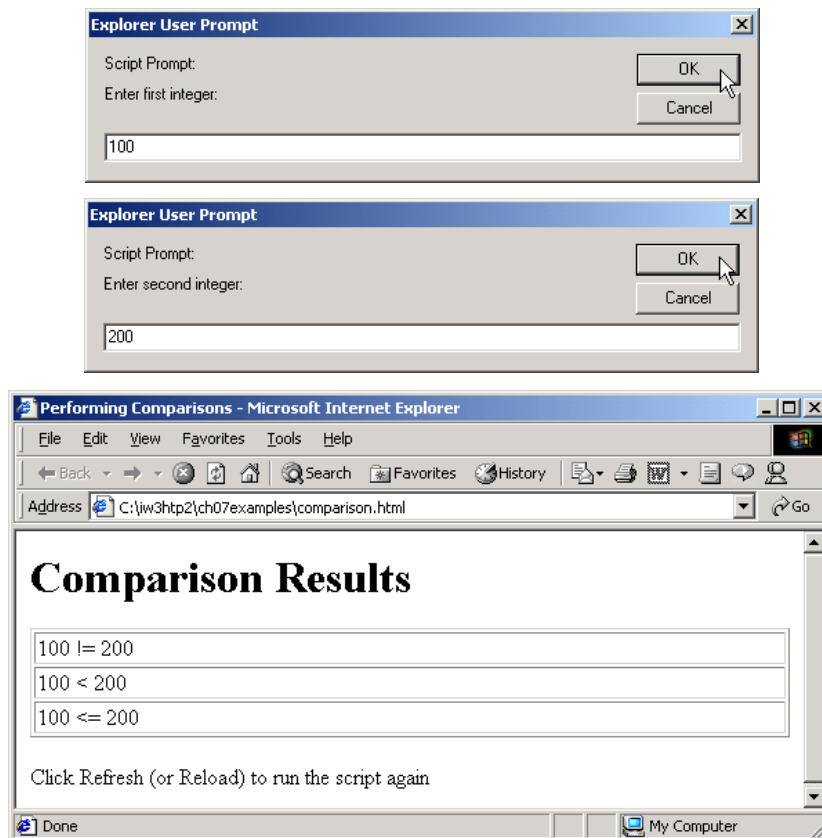
Each **if** statement uses the equality operators to determine the relationship of the two integers input by the user.



The screenshot shows the program's execution. It starts with two 'Explorer User Prompt' dialog boxes, both asking for the 'first integer' and showing the value '123'. Below these, a Microsoft Internet Explorer window titled 'Performing Comparisons - Microsoft Internet Explorer' displays the 'Comparison Results'. The results are shown in a table with three rows: '123 == 123', '123 <= 123', and '123 >= 123'. At the bottom of the browser window, a message says 'Click Refresh (or Reload) to run the script again'.



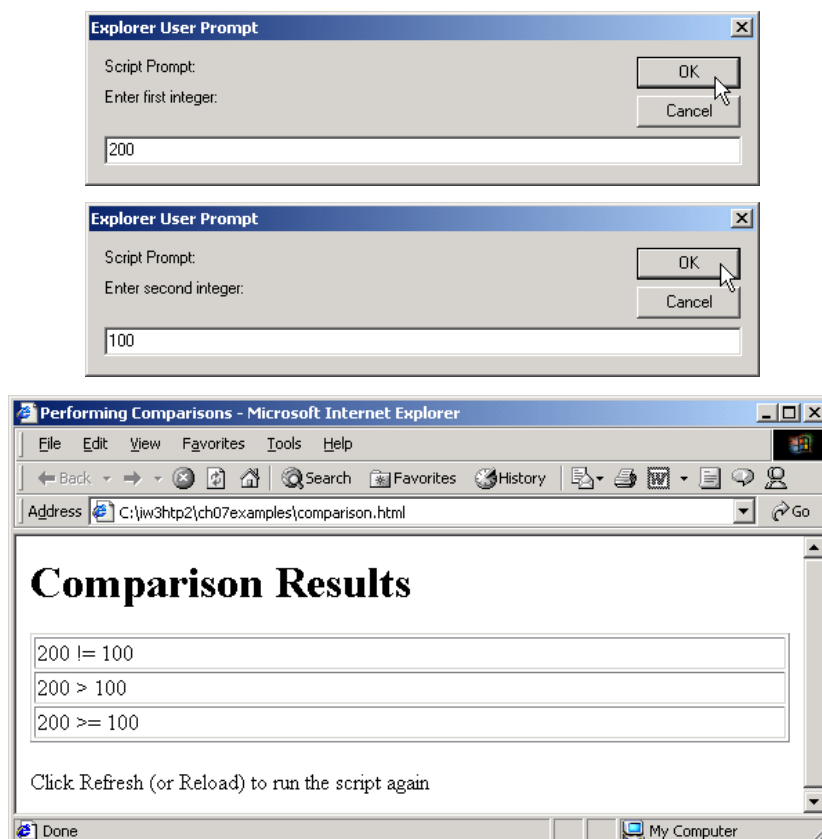
Program Output



© 2001 Prentice Hall, Inc.
All rights reserved.



Program Output



© 2001 Prentice Hall, Inc.
All rights reserved.

7.6 Decision Making: 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. 7.16 Precedence and associativity of the operators discussed so far.

