Chapter 7 - JavaScript: Introduction to Scripting

Outline

7.1 Introduction

- JavaScript scripting language
  - Enhances functionality and appearance
  - Client-side scripting
    - Makes pages more dynamic and interactive
  - Foundation for complex server-side scripting
  - Program development
  - Program control

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

- Inline scripting
  - Written in the <body> of a document
  - <script> tag
    - Indicate that the text is part of a script
    - type attribute
      - Specifies the type of file and the scripting language use
    - writeln method
      - Write a line in the document
    - Escape character (\ )
      - Indicates “special” character is used in the string
    - alert method
      - Dialog box

Outline

welcome.html
(1 of 1)

```
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.html     -->
6  <!-- Displaying a line of text -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
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>
```

Outline

welcome2.html
(1 of 1)

```
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>
```

```
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>
```

Outline

welcome3.html
1 of 1

```
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 (or Reload) to run this script again.</p>
21    </body>
22 </html>
```

Outline

welcome4.html
1 of 1


Microsoft Internet Explorer
Welcome to
JavaScript
Programming!
OK


Printing Multiple Lines in a Dialog Box - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address C:\IW3HTP3\examples\ch07\welcome4.html
Go Links
Click Refresh (or Reload) to run this script again.
Done
My Computer

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

Escape sequence	Description
\n	Newline. Position the screen cursor at 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 characters previously 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. 7.5 Some common escape sequences.

7.3.1 Dynamic Welcome Page

- A script can adapt the content based on input from the user or other variables

Outline

welcome5.html
(1 of 2)

```
1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
3     "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
4
5  <!-- Fig. 7.6: welcome5.html -->
6  <!-- Using Prompt Boxes      -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9     <head>
10       <title>Using Prompt and Alert Boxes</title>
11
12       <script type = "text/javascript">
13          <!--
14          var name; // string entered by the user
15
16          // read the name from the prompt box as a string
17          name = window.prompt( "Please enter your name", "GalAnt" );
18
19          document.writeln( "<h1>Hello, " + name +
20             ", welcome to JavaScript programming!</h1>" );
21          // -->
22       </script>
```

Outline

welcome5.html (2 of 2)

```
23
24      </head>
25
26      <body>
27      <p>Click Refresh (or Reload) to run this script again.</p>
28      </body>
29  </html>
```

7.3.1 Dynamic Welcome Page

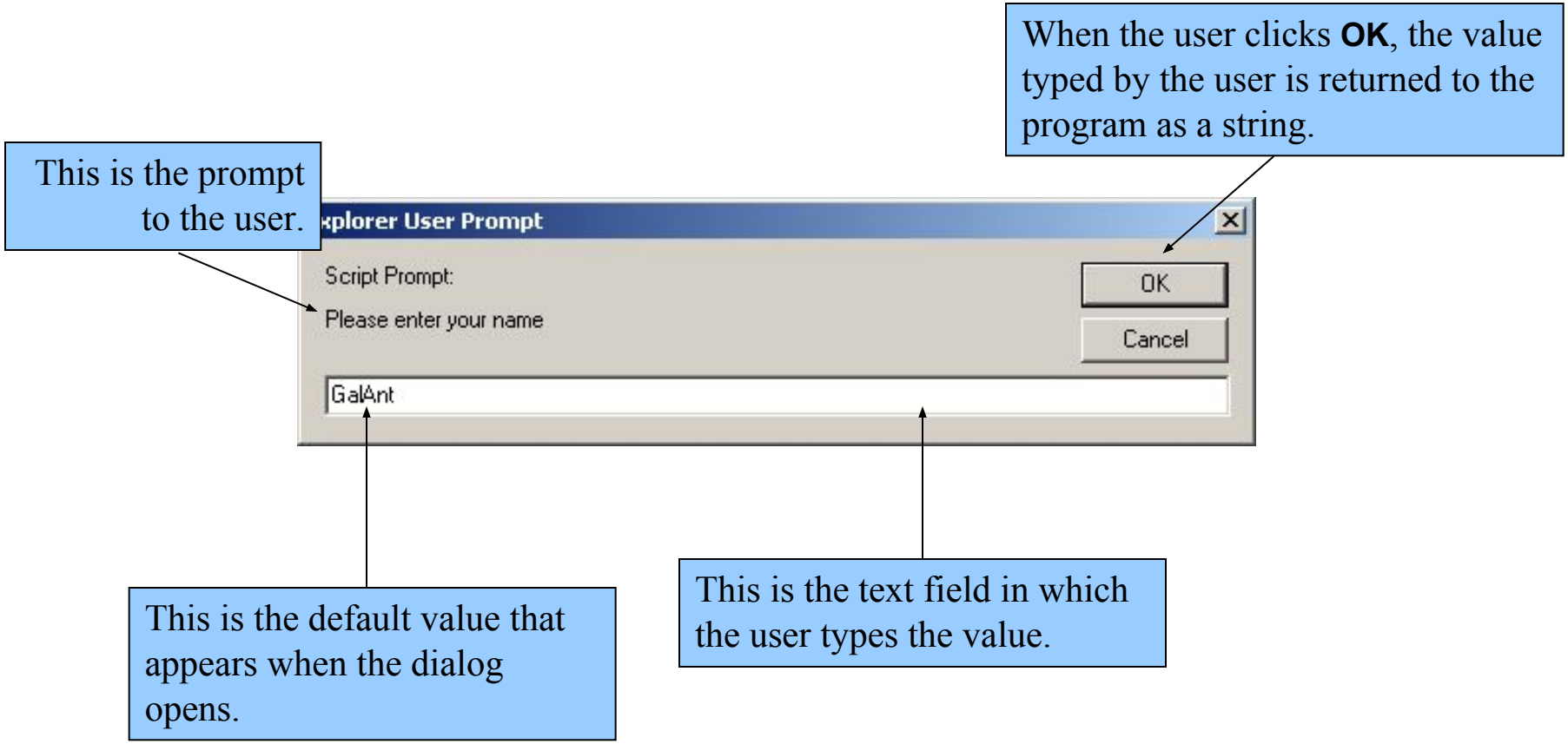


Fig. 7.7 Prompt dialog displayed by the window object's prompt method.

7.3.2 Adding Integers

- Prompt user for two integers and calculate the sum (Fig. 7.8)
- NaN (not a number)
- parseInt
  - Converts its string argument to an integer

Outline

Addition.html
(1 of 2)

```
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. 7.8: 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 string entered by user
15              secondNumber,  // second string entered by user
16              number1,       // first number to add
17              number2,       // second number to add
18              sum;           // sum of number1 and number2
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
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>
```

Outline

Addition.html
(2 of 2)


Explorer User Prompt
Script Prompt:
Enter first integer
OK
Cancel
45


Explorer User Prompt
Script Prompt:
Enter second integer
OK
Cancel
72


An Addition Program - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address C:\IW3HTP3\examples\ch07\Addition.html
Go Links
The sum is 117
Click Refresh (or Reload) to run the script again
Done
My Computer

7.5 Arithmetic

- Many scripts perform arithmetic calculations
  - Expressions in JavaScript must be written in straight-line form

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 $\frac{x}{y}$ or $x y$	x / y
Remainder	%	$r \bmod s$	r % s

Fig. 7.12 Arithmetic operators.

Operator(s)	Operation(s)	Order of evaluation (precedence)
*, / 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. 7.13 Precedence of arithmetic operators.

7.5 Arithmetic

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—place 72 into y)*

Fig. 7.14 Order in which a second-degree polynomial is evaluated.

7.6 Decision Making: Equality and Relational Operators

- Decision based on the truth or falsity of a condition
  - Equality operators
  - Relational operators

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
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. 7.15 Equality and relational operators.

Outline

welcome6.html
(1 of 3)

```
1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
3     "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
4
5  <!-- Fig. 7.16: welcome6.html     -->
6  <!-- Using Relational Operators -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9     <head>
10       <title>Using Relational Operators</title>
11
12       <script type = "text/javascript">
13          <!--
14          var name, // string entered by the user
15             now = new Date(),       // current date and time
16             hour  = now.getHours(); // current hour (0-23)
17
18          // read the name from the prompt box as a string
19          name = window.prompt( "Please enter your name", "GalAnt" );
20
21          // determine whether it is morning
22          if ( hour < 12 )
23             document.write( "<h1>Good Morning, " );
24
```

```
25          // determine whether the time is PM
26          if ( hour >= 12 )
27          {
28             // convert to a 12 hour clock
29             hour = hour - 12;
30
31             // determine whether it is before 6 PM
32             if ( hour < 6 )
33                document.write( "<h1>Good Afternoon, " );
34
35             // determine whether it is after 6 PM
36             if ( hour >= 6 )
37                document.write( "<h1>Good Evening, " );
38          }
39
40          document.writeln( name +
41             ", welcome to JavaScript programming!</h1>" );
42          // -->
43       </script>
44
45    </head>
46
```

Outline

welcome6.html
(2 of 3)

```
47    <body>
48       <p>Click Refresh (or Reload) to run this script again.</p>
49    </body>
50    </html>
```

Outline

welcome6.html
(3 of 3)

7.6 Decision Making: Equality and Relational Operators

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

Fig. 7.17 Precedence and associativity of the operators discussed so far.

Chapter 8 - JavaScript: Control Statements I

Outline

8.4 Control Structures

- Sequential execution
  - Statements execute in the order they are written
- Transfer of control
  - Next statement to execute may not be the next one in sequence
- Three control structures
  - Sequence structure
  - Selection structure
    - if
    - if…else
    - switch
  - Repetition structure
    - while
    - do…while
    - for
    - for…in

8.4 Control Structures

JavaScript Keywords				
break	case	catch	continue	default
delete	do	else	finally	for
function	if	in	instanceof	new
return	switch	this	throw	try
typeof	var	void	while	with
Keywords that are reserved but not currently used by JavaScript				
abstract	boolean	byte	char	class
const	debugger	double	enum	export
extends	final	float	goto	implements
import	int	interface	long	native
package	private	protected	public	short
static	super	synchronized	throws	transient
volatile				

Fig. 8.2 JavaScript keywords.

8.5 if Selection Statement

- Single-entry/single-exit structure
- Indicate action only when the condition evaluates to true

8.5 if Selection Statement

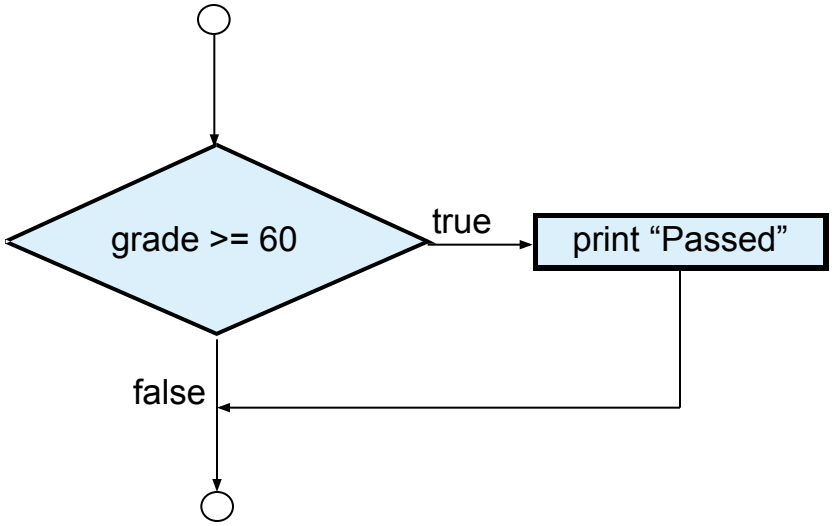


Fig. 8.3 Flowcharting the single-selection `if` statement.

8.6 if…else Selection Statement

- Indicate different actions to be perform when condition is true or false
- Conditional operator (?:)
  - JavaScript's only ternary operator
    - Three operands
    - Forms a conditional expression
- Dangling-else problem

8.6 if…else Selection Statement

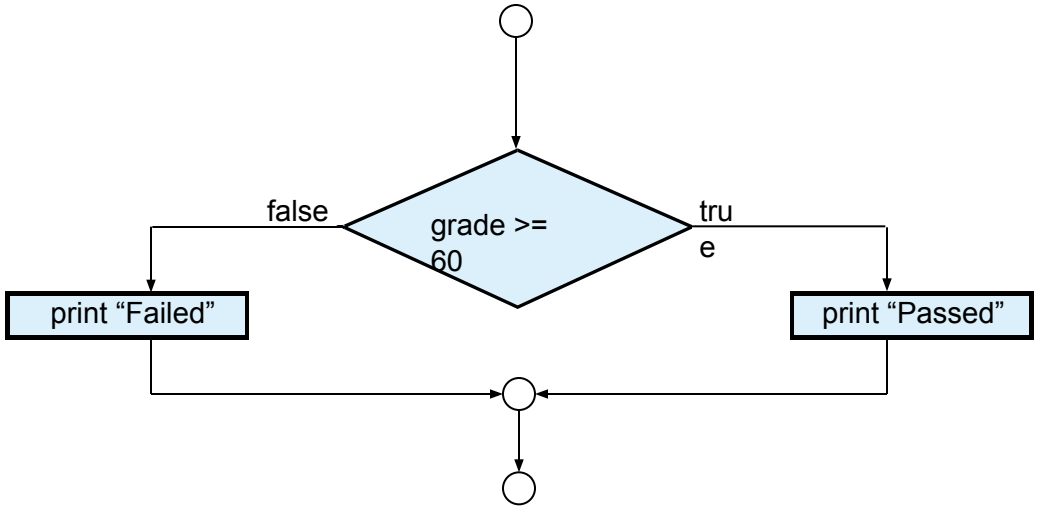


Fig. 8.4 Flowcharting the double-selection if…else statement.

8.7 while Repetition Statement

- Repetition structure (loop)
  - Repeat action while some condition remains true

8.7 while Repetition Statement

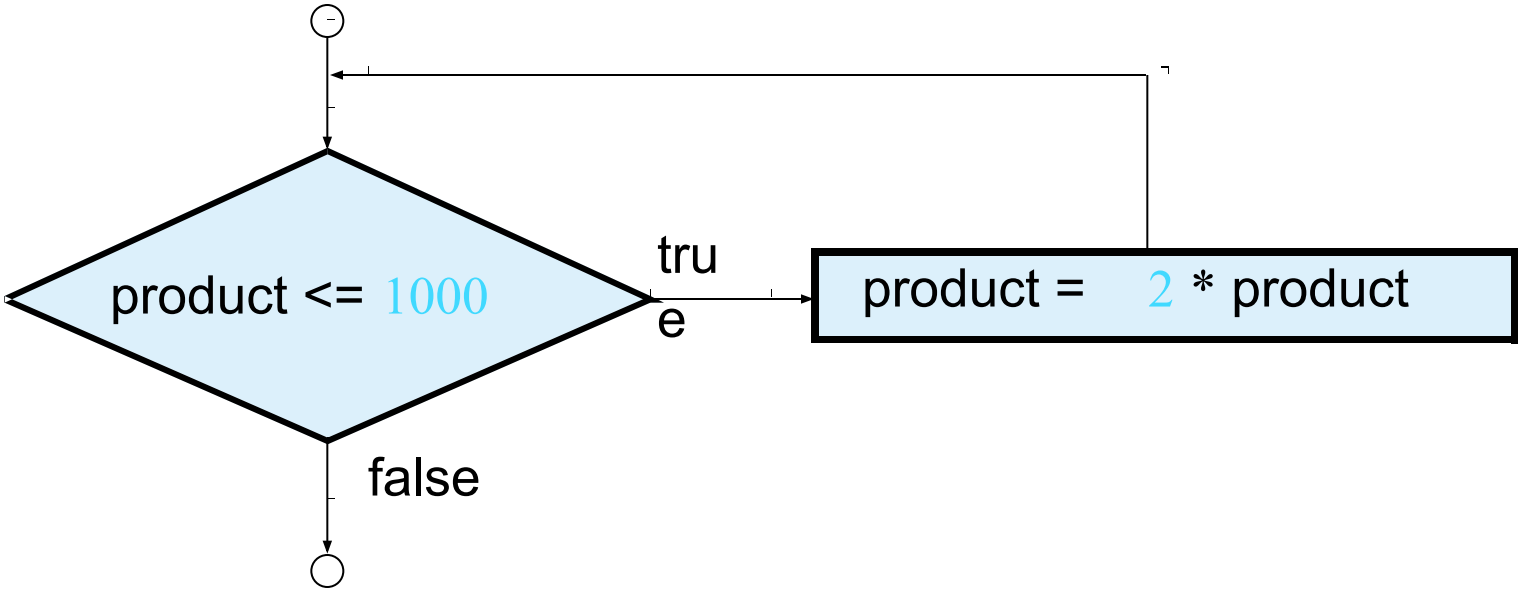


Fig. 8.5 Flowcharting the while repetition statement.

8.8 Formulating Algorithms: Case Study 1 (Counter-Controlled Repetition)

- Counter-controlled repetition
  - Counter
    - Control the number of times a set of statements executes
  - Definite repetition

Outline

average.html
(1 of 3)

```
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. 8.7: average.html -->
6  <!-- Class Average Program -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9     <head>
10       <title>Class Average Program</title>
11
12       <script type = "text/javascript">
13          <!--
14          var total,          // sum of grades
15              gradeCounter,   // number of grades entered
16              gradeValue,     // grade value
17              average,        // average of all grades
18              grade;          // grade typed by user
19
20          // Initialization Phase
21          total = 0;          // clear total
22          gradeCounter = 1;   // prepare to loop
23
```

```
24          // Processing Phase
25          while ( gradeCounter <= 10 ) {  // loop 10 times
26
27             // prompt for input and read grade from user
28             grade = window.prompt( "Enter integer grade:", "0" );
29
30             // convert grade from a string to an integer
31             gradeValue = parseInt( grade );
32
33             // add gradeValue to total
34             total = total + gradeValue;
35
36             // add 1 to gradeCounter
37             gradeCounter = gradeCounter + 1;
38          }
39
40          // Termination Phase
41          average = total / 10;  // calculate the average
42
43          // display average of exam grades
44          document.writeln(
45             "<h1>Class average is " + average + "</h1>" );
46          // -->
47       </script>
```

Outline

average.html
(2 of 3)

```
48
49      </head>
50      <body>
51         <p>Click Refresh (or Reload) to run the script again<p>
52      </body>
53   </html>
```


average.html
(3 of 3)

8.9 Formulating Algorithms with Top-Down, Stepwise Refinement: Case Study 2 (Sentinel-Controlled Repetition)

- Indefinite repetition
  - Sentinel value

Outline

average2.html
(1 of 3)

```
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. 8.9: average2.html          -->
6  <!-- Sentinel-controlled Repetition -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9     <head>
10       <title>Class Average Program:
11           Sentinel-controlled Repetition</title>
12
13       <script type = "text/javascript">
14          <!--
15          var gradeCounter,  // number of grades entered
16              gradeValue,    // grade value
17              total,         // sum of grades
18              average,       // average of all grades
19              grade;         // grade typed by user
20
21          // Initialization phase
22          total = 0;         // clear total
23          gradeCounter = 0;  // prepare to loop
24
```

```
25          // Processing phase
26          // prompt for input and read grade from user
27          grade = window.prompt(
28              "Enter Integer Grade, -1 to Quit:", "0" );
29
30          // convert grade from a string to an integer
31          gradeValue = parseInt( grade );
32
33          while ( gradeValue != -1 ) {
34             // add gradeValue to total
35             total = total + gradeValue;
36
37             // add 1 to gradeCounter
38             gradeCounter = gradeCounter + 1;
39
40             // prompt for input and read grade from user
41             grade = window.prompt(
42                "Enter Integer Grade, -1 to Quit:", "0" );
43
44             // convert grade from a string to an integer
45             gradeValue = parseInt( grade );
46          }
47
```

Outline

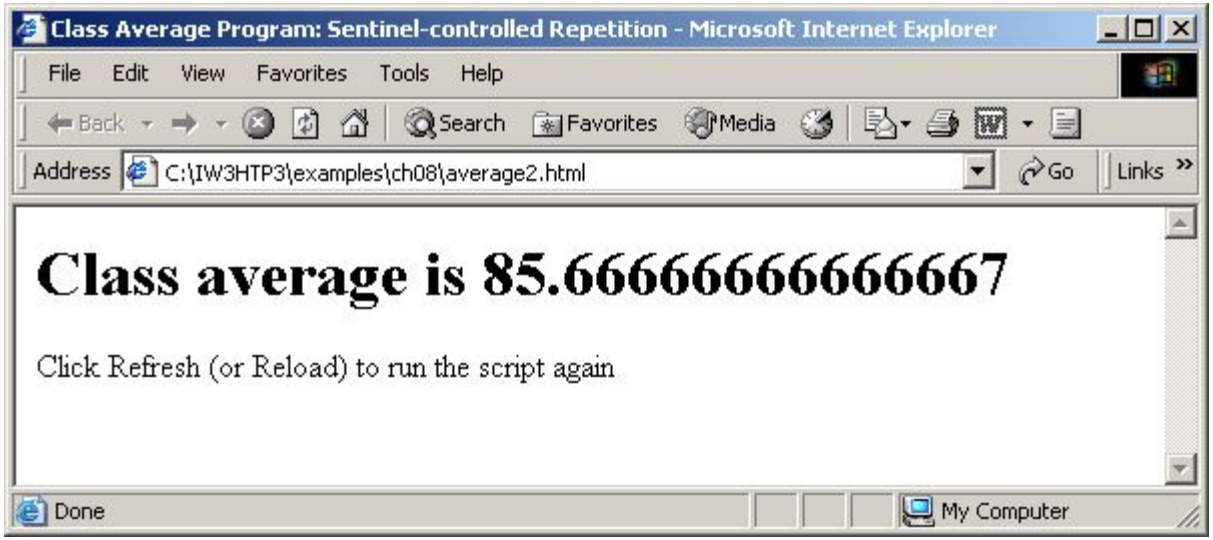
average2.html
(2 of 3)

```
48          // Termination phase
49          if ( gradeCounter != 0 ) {
50             average = total / gradeCounter;
51
52             // display average of exam grades
53             document.writeln(
54                "<h1>Class average is " + average + "</h1>" );
55          }
56          else
57             document.writeln( "<p>No grades were entered</p>" );
58          // -->
59       </script>
60    </head>
61
62    <body>
63       <p>Click Refresh (or Reload) to run the script again</p>
64    </body>
65 </html>
```

Outline

average2.html
(3 of 3)


Explorer User Prompt
Script Prompt:
Enter Integer Grade, -1 to Quit:
OK
Cancel
97


Class Average Program: Sentinel-controlled Repetition - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address C:\IW3HTP3\examples\ch08\average2.html
Go Links
Class average is 85.66666666666667
Click Refresh (or Reload) to run the script again
Done
My Computer

8.10 Formulating Algorithms with Top-Down, Stepwise Refinement: Case Study 3 (Nested Control Structures)

- Consider problem
- Make observations
- Top-down, stepwise refinement

Outline

analysis.html
(1 of 2)

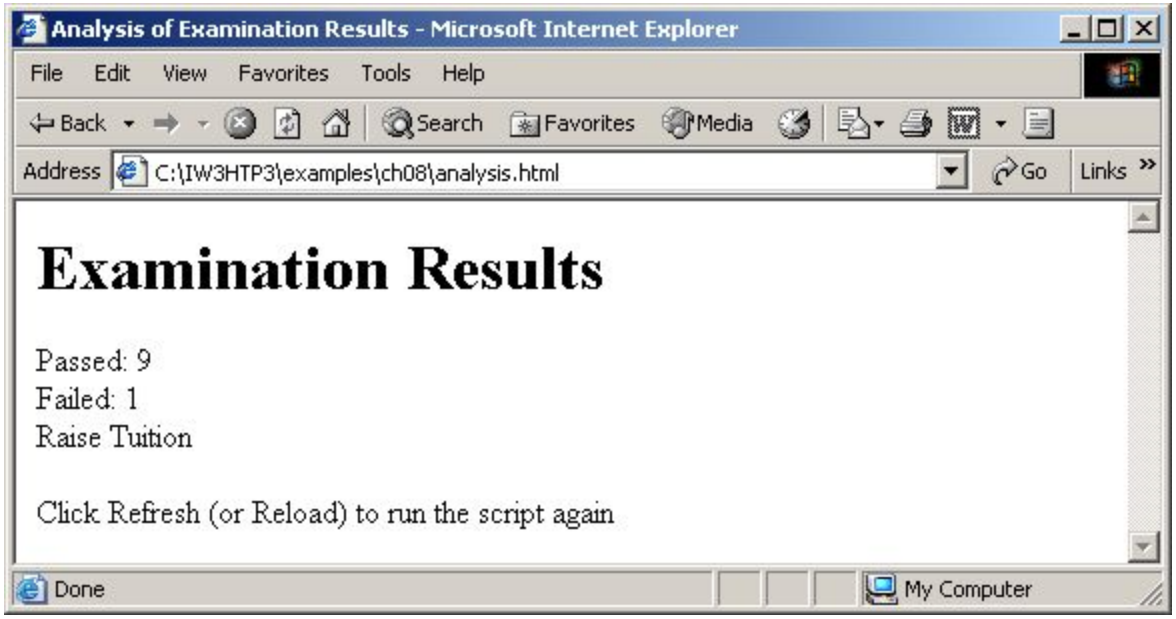
```
1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5  <!-- Fig. 8.11: analysis.html -->
6  <!-- Analyzing Exam Results   -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9     <head>
10       <title>Analysis of Examination Results</title>
11
12       <script type = "text/javascript">
13          <!--
14          // initializing variables in declarations
15          var passes = 0,      // number of passes
16              failures = 0,    // number of failures
17              student = 1,     // student counter
18              result;          // one exam result
19
20          // process 10 students; counter-controlled loop
21          while ( student <= 10 ) {
22             result = window.prompt(
23                "Enter result (1=pass,2=fail)", "0" );
24
```

```
25            if ( result == "1" )
26               passes = passes + 1;
27            else
28               failures = failures + 1;
29
30            student = student + 1;
31         }
32
33         // termination phase
34         document.writeln( "<h1>Examination Results</h1>" );
35         document.writeln(
36            "Passed: " + passes + "<br />Failed: " + failures );
37
38         if ( passes > 8 )
39            document.writeln( "<br />Raise Tuition" );
40         // -->
41      </script>
42
43   </head>
44   <body>
45      <p>Click Refresh (or Reload) to run the script again</p>
46   </body>
47 </html>
```

Outline

analysis.html
(2 of 2)


Explorer User Prompt
Script Prompt:
Enter result (1=pass,2=fail)
OK
Cancel
1


Analysis of Examination Results - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address C:\IW3HTP3\examples\ch08\analysis.html
Go Links
Examination Results
Passed: 9
Failed: 1
Raise Tuition
Click Refresh (or Reload) to run the script again
Done
My Computer


Explorer User Prompt
Script Prompt:
Enter result (1=pass,2=fail)
OK
Cancel
1


Analysis of Examination Results - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address C:\IW3HTP3\examples\ch08\analysis.html
Go Links
Examination Results
Passed: 5
Failed: 5
Click Refresh (or Reload) to run the script again
Done
My Computer

8.11 Assignment Operators

- Compound assignment operators
  - Abbreviate assignment expressions

8.11 Assignment Operators

Assignment operator	Initial value of variable	Sample expression	Explanation	Assigns
+=	c = 3	c += 7	c = c + 7	10 to c
-=	d = 5	d -= 4	d = d - 4	1 to d
*=	e = 4	e *= 5	e = e * 5	20 to e
/=	f = 6	f /= 3	f = f / 3	2 to f
%=	g = 12	g %= 9	g = g % 9	3 to g

Fig. 8.12 Arithmetic assignment operators.

8.12 Increment and Decrement Operators

- Preincrement or predecrement operator
  - Increment of decrement operator placed before a variable
- Postincrement or postdecrement operator
  - Increment of decrement operator placed after a variable

8.12 Increment and Decrement Operators

Operator	Called	Sample expression	Explanation
++	preincrement	++a	Increment a by 1, then use the new value of a in the expression in which a resides.
++	postincrement	a++	Use the current value of a in the expression in which a resides, then increment a by 1.
- -	predecrement	- - b	Decrement b by 1, then use the new value of b in the expression in which b resides.
- -	postdecrement	b- -	Use the current value of b in the expression in which b resides, then decrement b by 1.

Fig. 8.13 increment and decrement operators.

Outline

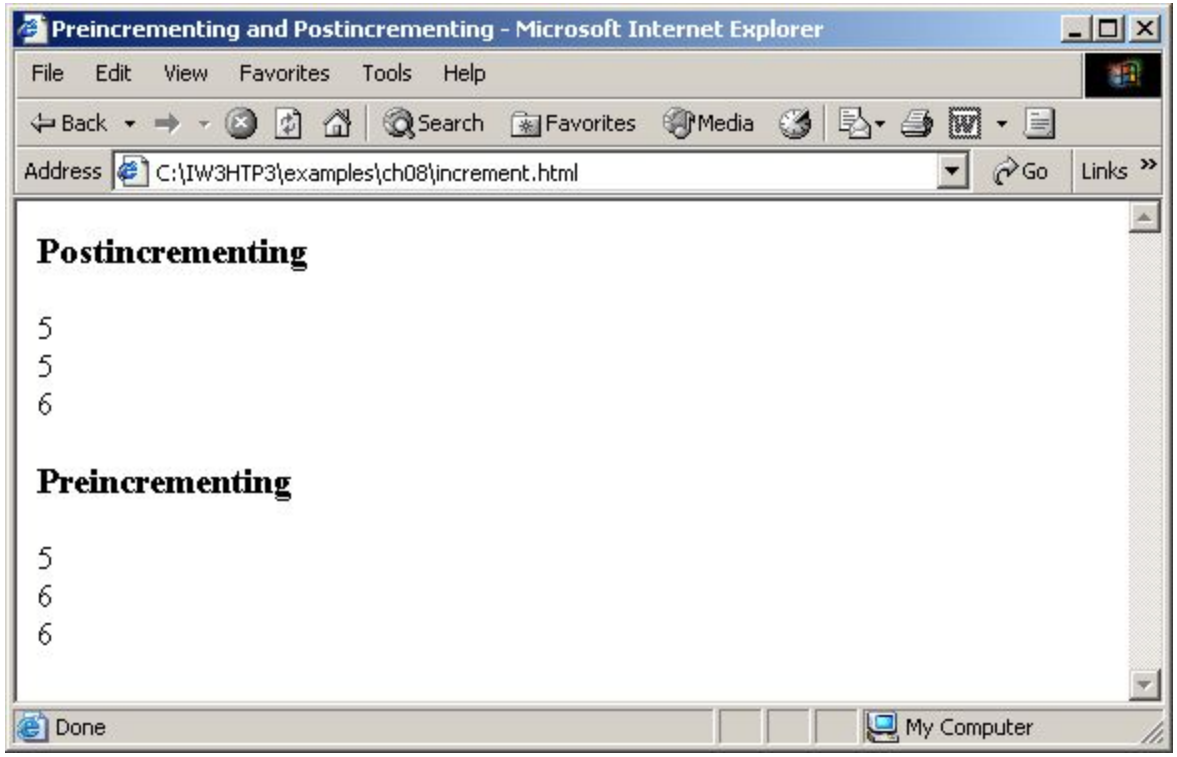
increment.html
(1 of 2)

```
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. 8.14: increment.html              -->
6  <!-- Preincrementing and Postincrementing -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9     <head>
10       <title>Preincrementing and Postincrementing</title>
11
12       <script type = "text/javascript">
13          <!--
14          var c;
15
16          c = 5;
17          document.writeln( "<h3>Postincrementing</h3>" );
18          document.writeln( c );            // print 5
19          // print 5 then increment
20          document.writeln( "<br />" + c++ );
21          document.writeln( "<br />" + c );   // print 6
22
23          c = 5;
24          document.writeln( "<h3>Preincrementing</h3>" );
25          document.writeln( c );            // print 5
```

```
26          // increment then print 6
27          document.writeln( "<br />" + ++c );
28          document.writeln( "<br />" + c );   // print 6
29          // -->
30       </script>
31
32    </head><body></body>
33 </html>
```

Outline

increment.html
(2 of 2)



8.12 Increment and Decrement Operators

Operator	Associativity	Type
++ --	right to left	unary
* / %	left to right	multiplicative
+ -	left to right	additive
< <= > >=	left to right	relational
== !=	left to right	equality
?:	right to left	conditional
= += -= *= /= %=	right to left	assignment

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

Chapter 9 - JavaScript: Control Statements II

Outline

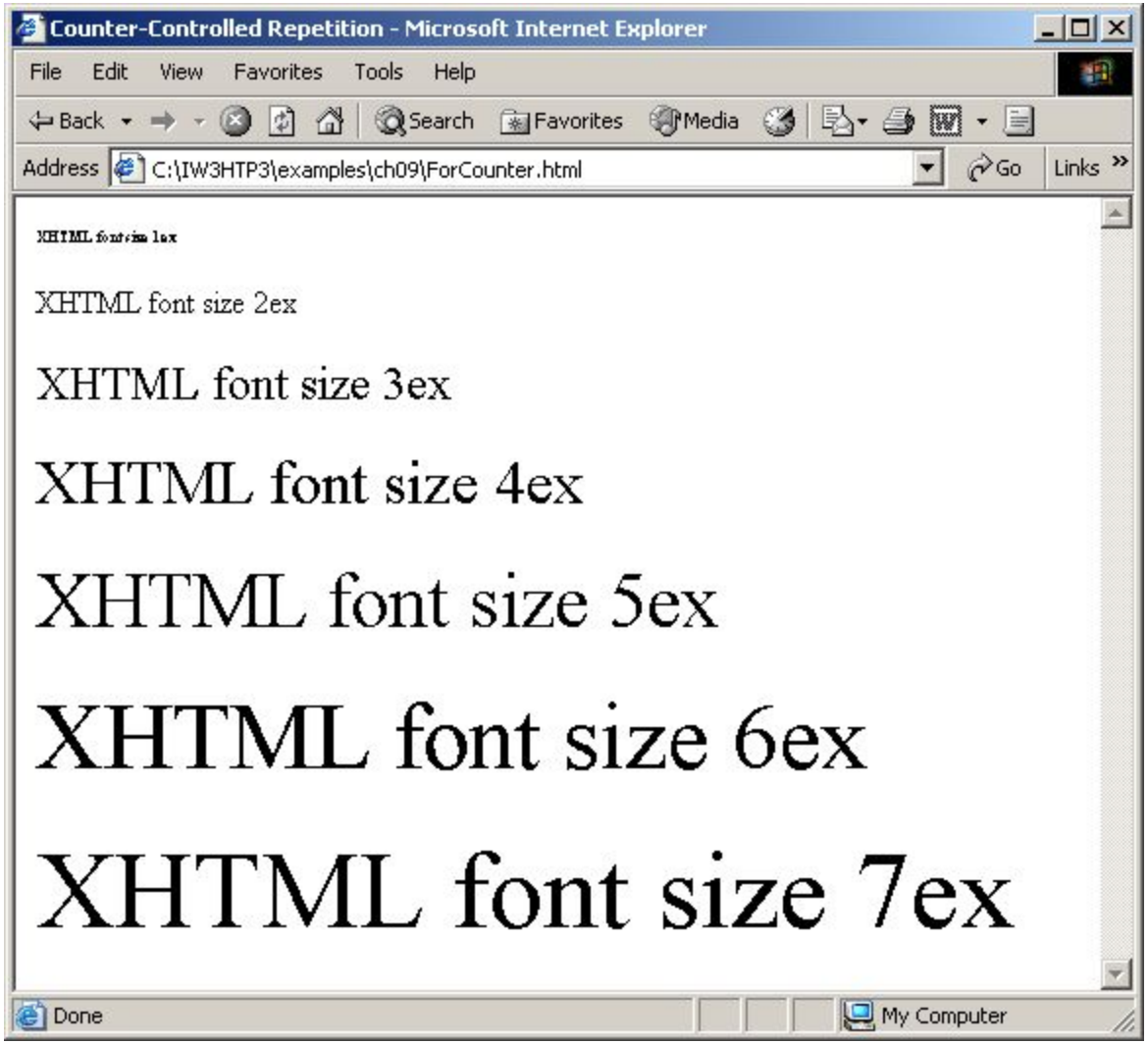
9.3 for Repetition Statement

- for repetition statement
  - Handles all the details of counter-controlled repetition
  - for structure header
    - The first line

```
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. 9.2: ForCounter.html                          -->
6  <!-- Counter-Controlled Repetition with for statement -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9     <head>
10       <title>Counter-Controlled Repetition</title>
11
12       <script type = "text/javascript">
13          <!--
14          // Initialization, repetition condition and
15          // incrementing are all included in the for
16          // statement header.
17          for ( var counter = 1; counter <= 7; ++counter )
18             document.writeln( "<p style = \"font-size: " +
19                counter + "ex\">XHTML font size " + counter +
20                "ex</p>" );
21          // -->
22       </script>
23
24    </head><body></body>
25 </html>
```

Outline

ForCounter.html
(1 of 1)


Counter-Controlled Repetition - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address C:\IW3HTP3\examples\ch09\ForCounter.html
Go Links
XHTML font size 1ex
XHTML font size 2ex
XHTML font size 3ex
XHTML font size 4ex
XHTML font size 5ex
XHTML font size 6ex
XHTML font size 7ex
Done
My Computer

9.3 for Repetition Statement

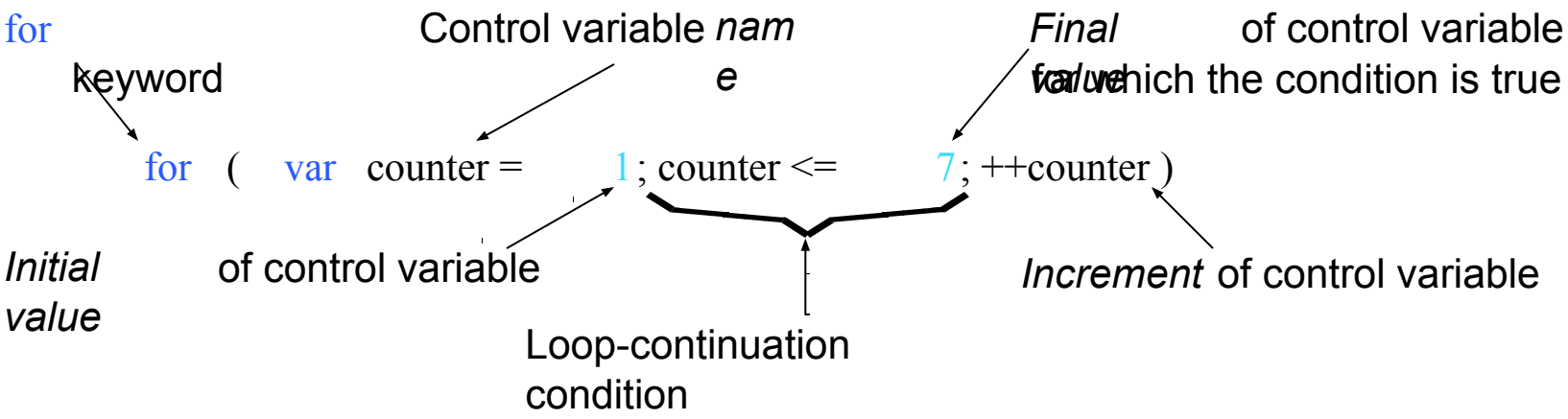


Fig. 9.3 for statement header components.

9.3 for Repetition Statement

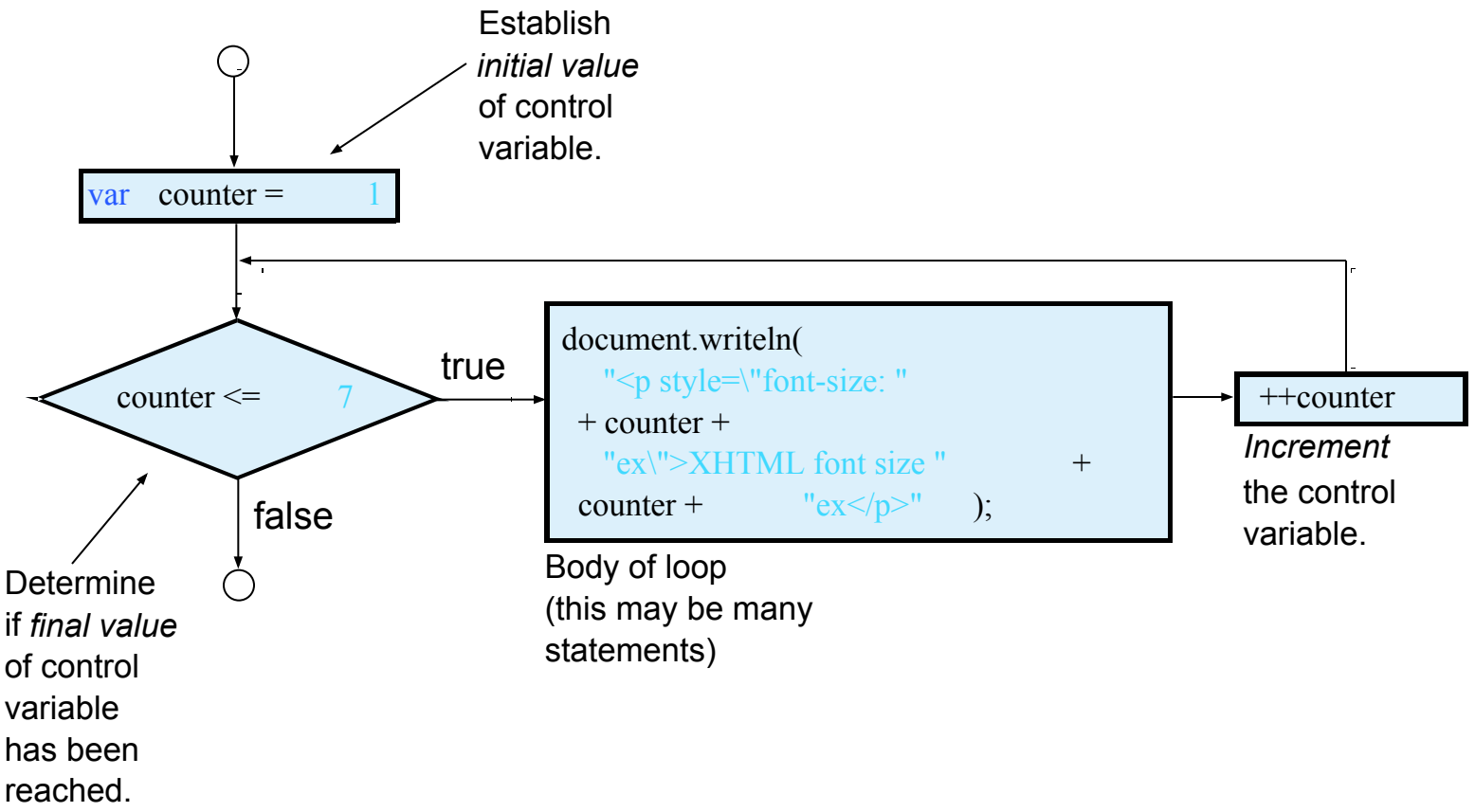


Fig. 9.4 for repetition structure flowchart.

9.4 Examples Using the for Statement

- Summation with for
- Compound interest calculation with for loop
  - Math object
    - Method pow
    - Method round

Outline

Sum.html
(1 of 1)

```
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. 9.5: Sum.html                    -->
6  <!-- Using the for repetition statement -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9     <head>
10       <title>Sum the Even Integers from 2 to 100</title>
11
12       <script type = "text/javascript">
13          <!--
14          var sum = 0;
15
16          for ( var number = 2; number <= 100; number += 2 )
17             sum += number;
18
19          document.writeln( "The sum of the even integers " +
20             "from 2 to 100 is " + sum);
21          // -->
22       </script>
23
24    </head><body></body>
25 </html>
```

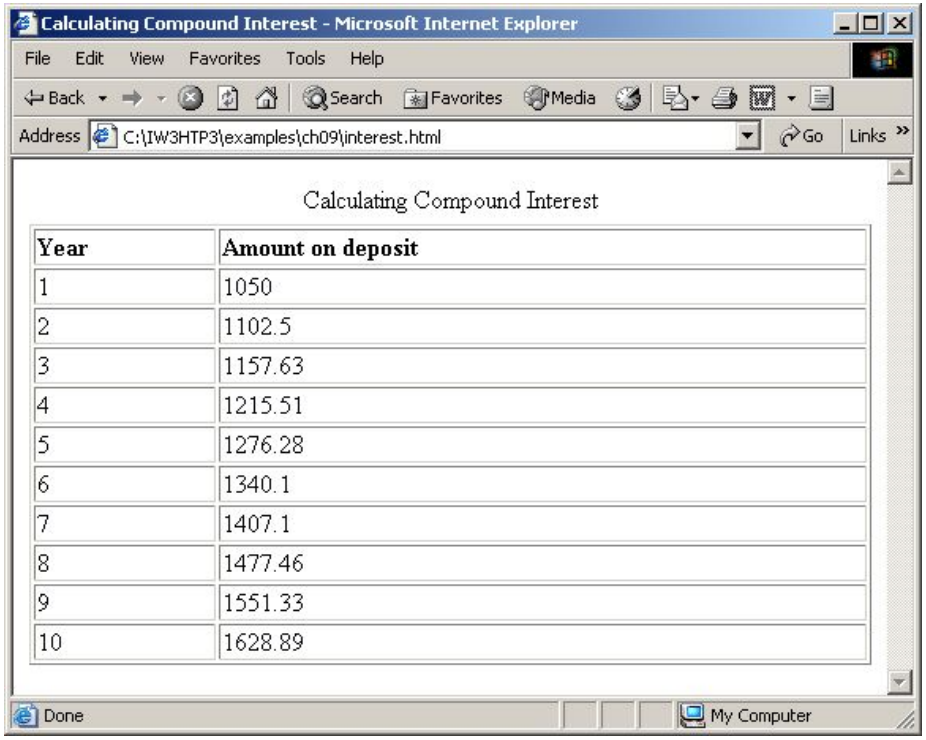

Sum the Even Integers from 2 to 100 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address C:\IW3HTP3\examples\ch09\Sum.html
Go Links
The sum of the even integers from 2 to 100 is 2550
Done
My Computer

Outline

Interest.html
(1 of 2)

```
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. 9.6: Interest.html                -->
6  <!-- Using the for repetition statement -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9     <head>
10       <title>Calculating Compound Interest</title>
11
12       <script type = "text/javascript">
13          <!--
14          var amount, principal = 1000.0, rate = .05;
15
16          document.writeln(
17             "<table border = \"1\" width = \"100%\">" );
18          document.writeln(
19             "<caption>Calculating Compound Interest</caption>" );
20          document.writeln(
21             "<thead><tr><th align = \"left\">Year</th>" );
22          document.writeln(
23             "<th align = \"left\">Amount on deposit</th>" );
24          document.writeln( "</tr></thead>" );
25
```

```
26          for ( var year = 1; year <= 10; ++year ) {
27             amount = principal * Math.pow( 1.0 + rate, year );
28             document.writeln( "<tbody><tr><td>" + year +
29                "</td><td>" + Math.round( amount * 100 ) / 100 +
30                "</td></tr>" );
31          }
32
33          document.writeln( "</tbody></table>" );
34          // -->
35       </script>
36
37    </head><body></body>
38 </html>
```



Outline

Interest.html
(2 of 2)

9.5 switch Multiple-Selection Statement

- Controlling expression
- Case labels
- Default case

```
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. 9.7: SwitchTest.html  -->
6  <!-- Using the switch statement -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9     <head>
10       <title>Switching between XHTML List Formats</title>
11
12       <script type = "text/javascript">
13          <!--
14          var choice,              // user's choice
15              startTag,            // starting list item tag
16              endTag,              // ending list item tag
17              validInput = true,   // indicates if input is valid
18              listType;            // list type as a string
19
20          choice = window.prompt( "Select a list style:\n" +
21              "1 (bullet), 2 (numbered), 3 (lettered)", "1" );
22
```

Outline

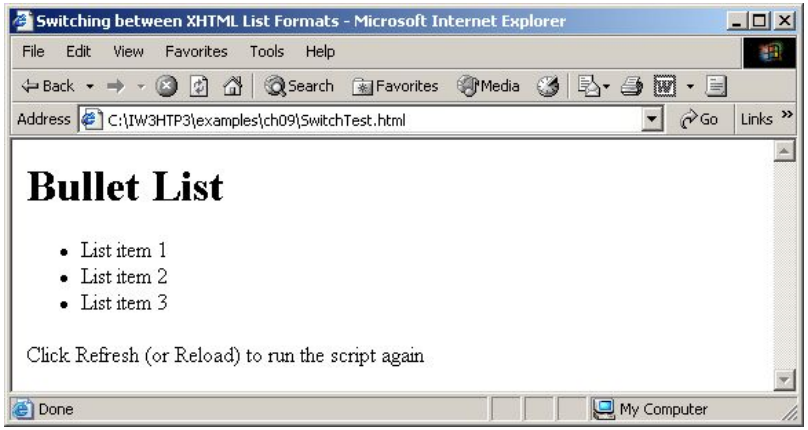
SwitchTest.html
(1 of 3)

```
23          switch ( choice ) {
24             case "1":
25                startTag = "<ul>";
26                endTag = "</ul>";
27                listType = "<h1>Bullet List</h1>";
28                break;
29             case "2":
30                startTag = "<ol>";
31                endTag = "</ol>";
32                listType = "<h1>Ordered List: Numbered</h1>";
33                break;
34             case "3":
35                startTag = "<ol type = \"A\">";
36                endTag = "</ol>";
37                listType = "<h1>Ordered List: Lettered</h1>";
38                break;
39             default:
40                validInput = false;
41          }
42
43          if ( validInput == true ) {
44             document.writeln( listType + startTag );
45
46             for ( var i = 1; i <= 3; ++i )
47                document.writeln( "<li>List item" + i + "</li>" );
```

Outline

SwitchTest.html
(2 of 3)

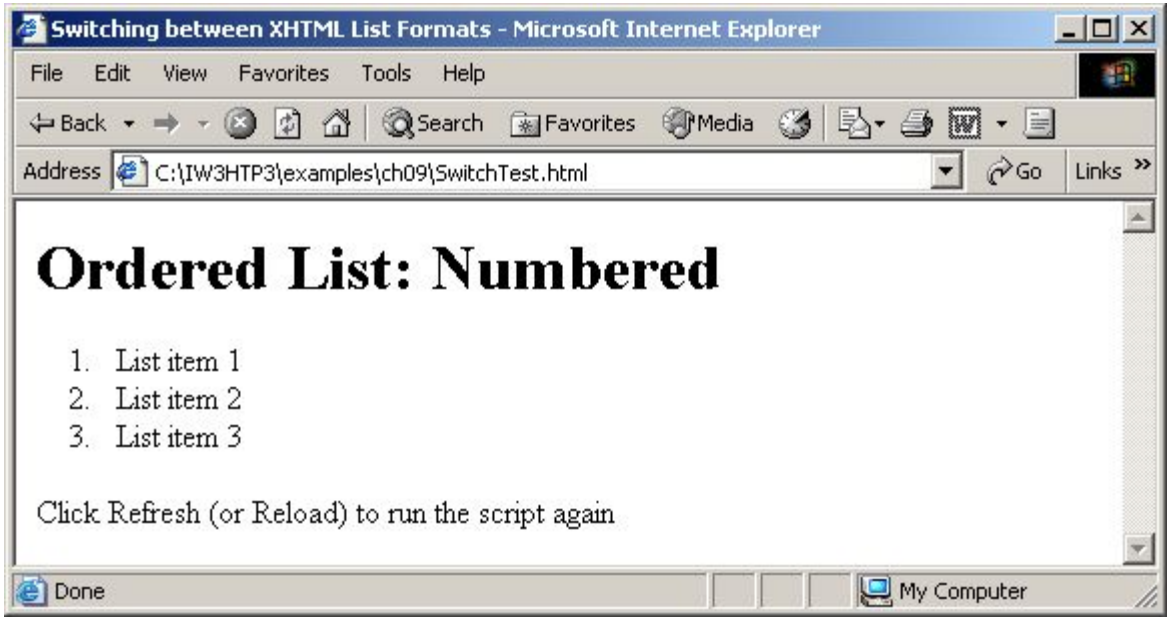
```
48
49            document.writeln( endTag );
50         }
51         else
52            document.writeln( "Invalid choice: " + choice );
53         // -->
54      </script>
55
56   </head>
57   <body>
58      <p>Click Refresh (or Reload) to run the script again</p>
59   </body>
60 </html>
```



Outline

SwitchTest.html (3 of 3)


Explorer User Prompt
Script Prompt:
Select a list style:
1 (bullet), 2 (numbered), 3 (lettered)
OK
Cancel
2


Switching between XHTML List Formats - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address C:\IW3HTP3\examples\ch09\SwitchTest.html
Go Links
Ordered List: Numbered
1. List item 1
2. List item 2
3. List item 3
Click Refresh (or Reload) to run the script again
Done
My Computer


Explorer User Prompt
Script Prompt:
Select a list style:
1 (bullet), 2 (numbered), 3 (lettered)
OK
Cancel
3


Switching between XHTML List Formats - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address C:\IW3HTP3\examples\ch09\SwitchTest.html
Go Links
Ordered List: Lettered
A. List item 1
B. List item 2
C. List item 3
Click Refresh (or Reload) to run the script again
Done
My Computer

9.6 do…while Repetition Statement

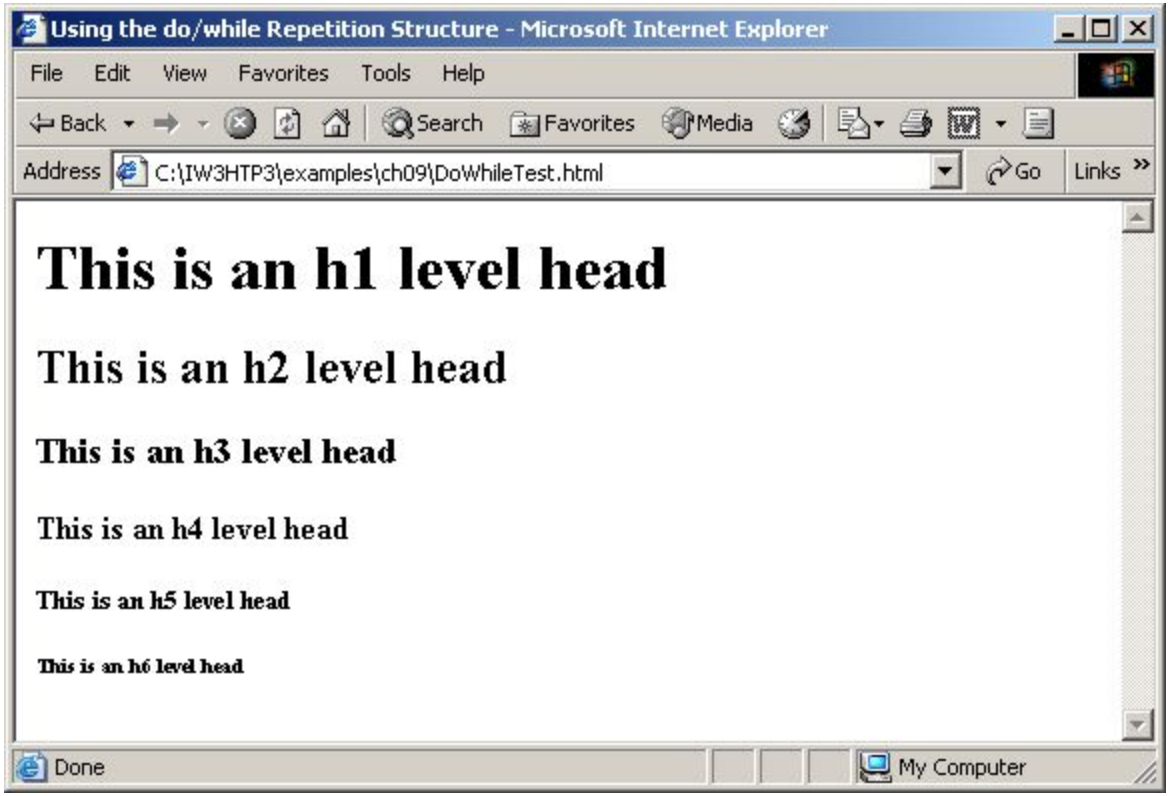
- Similar to the while statement
- Tests the loop continuation condition after the loop body executes
- Loop body always executes at least once

Outline

DoWhileTest.html
(1 of 2)

```
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. 9.9: DoWhileTest.html     -->
6  <!-- Using the do...while statement -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9     <head>
10       <title>Using the do...while Repetition Statement</title>
11
12       <script type = "text/javascript">
13          <!--
14          var counter = 1;
15
16          do {
17             document.writeln( "<h" + counter + ">This is " +
18                "an h" + counter + " level head" + "</h" +
19                counter + ">" );
20
21             ++counter;
22          } while ( counter <= 6 );
23          // -->
24       </script>
```

```
25
26      </head><body></body>
27  </html>
```



Outline

DoWhileTest.html
(2 of 2)

9.7 break and continue Statements

- break
  - Immediate exit from the structure
  - Used to escape early from a loop
  - Skip the remainder of a switch statement
- continue
  - Skips the remaining statements in the body of the structure
  - Proceeds with the next iteration of the loop

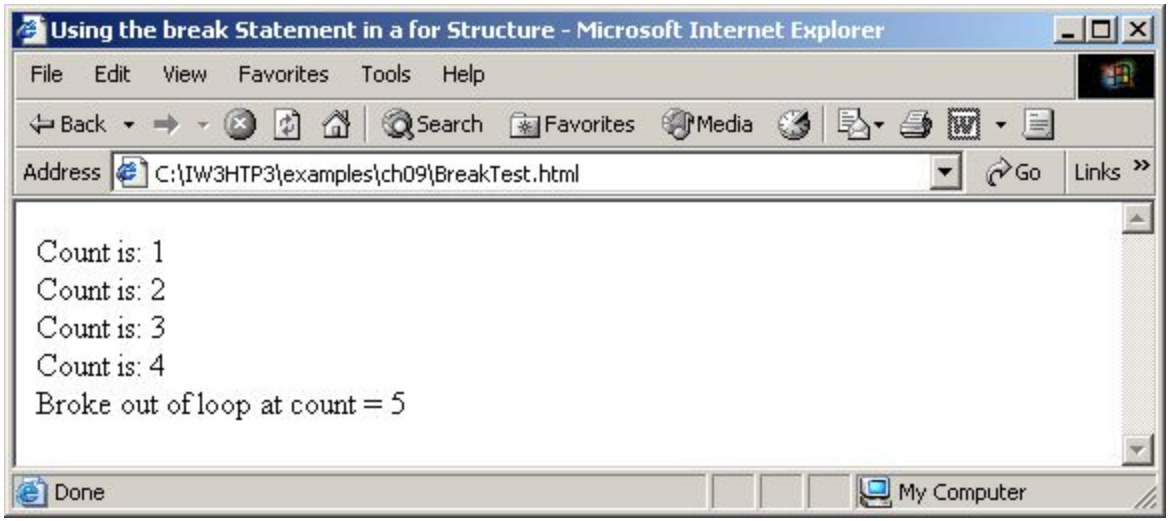
Outline

BreakTest.html
(1 of 2)

```
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. 9.11: BreakTest.html    -->
6  <!-- Using the break statement    -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9     <head>
10       <title>
11          Using the break Statement in a for Structure
12       </title>
13
14       <script type = "text/javascript">
15          <!--
16          for ( var count = 1; count <= 10; ++count ) {
17             if ( count == 5 )
18                break;  // break loop only if count == 5
19
20             document.writeln( "Count is: " + count + "<br />" );
21          }
22
```

```
23          document.writeln(
24             "Broke out of loop at count = " + count );
25          // -->
26       </script>
27
28    </head><body></body>
29 </html>
```


BreakTest.html
(2 of 2)



Outline

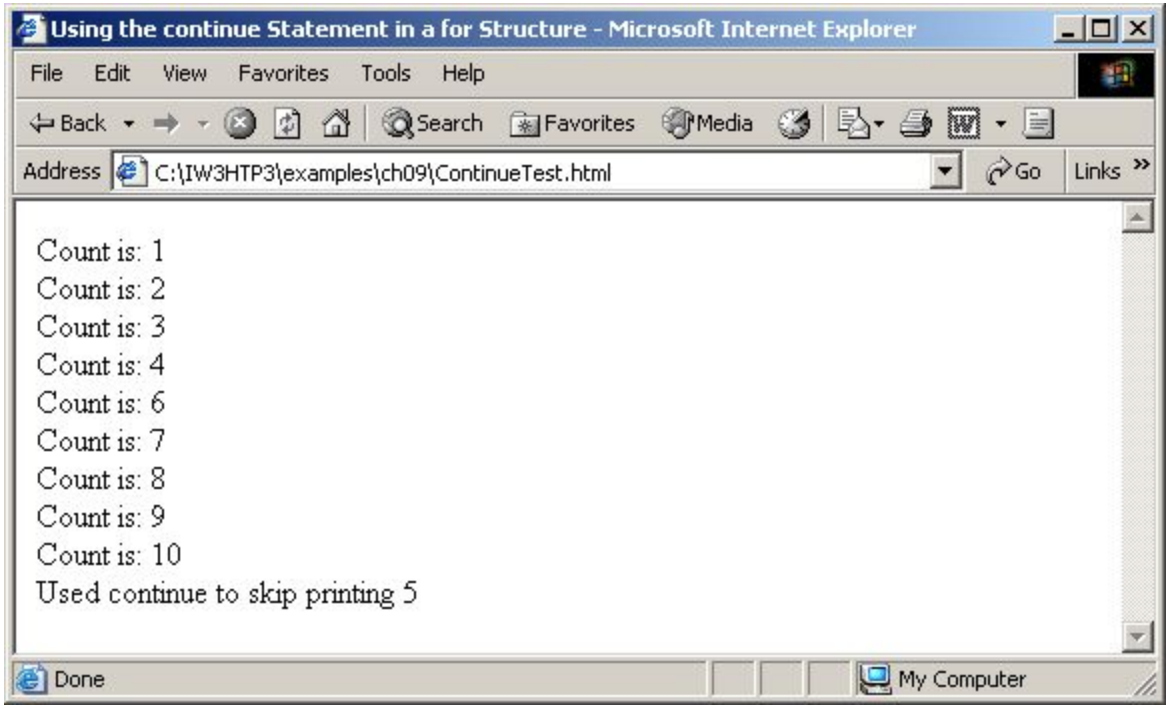
ContinueTest.html
(1 of 2)

```
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. 9.12: ContinueTest.html  -->
6  <!-- Using the break statement     -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9     <head>
10       <title>
11          Using the continue Statement in a for Structure
12       </title>
13
14       <script type = "text/javascript">
15          <!--
16          for ( var count = 1; count <= 10; ++count ) {
17             if ( count == 5 )
18                continue;  // skip remaining code in loop
19                           // only if count == 5
20
21             document.writeln( "Count is: " + count + "<br />" );
22          }
23
```

```
24          document.writeln( "Used continue to skip printing 5" );
25          // -->
26       </script>
27
28    </head><body></body>
29 </html>
```

Outline

ContinueTest.html
(2 of 2)



9.8 Labeled break and continue Statements

- Labeled break statement
  - Break out of a nested set of structures
  - Immediate exit from that structure and enclosing repetition structures
  - Execution resumes with first statement after enclosing labeled statement
- Labeled continue statement
  - Skips the remaining statements in structure's body and enclosing repetition structures
  - Proceeds with next iteration of enclosing labeled repetition structure
  - Loop-continuation test evaluates immediately after the continue statement executes

```
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. 9.13: BreakLabelTest.html          -->
6  <!-- Using the break statement with a Label -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9     <head>
10       <title>Using the break Statement with a Label</title>
11
12       <script type = "text/javascript">
13          <!--
14          stop: {   // labeled block
15             for ( var row = 1; row <= 10; ++row ) {
16                for ( var column = 1; column <= 5 ; ++column ) {
17
18                   if ( row == 5 )
19                      break stop; // jump to end of stop block
20
21                   document.write( "* " );
22                }
23
24                document.writeln( "<br />" );
25             }
```

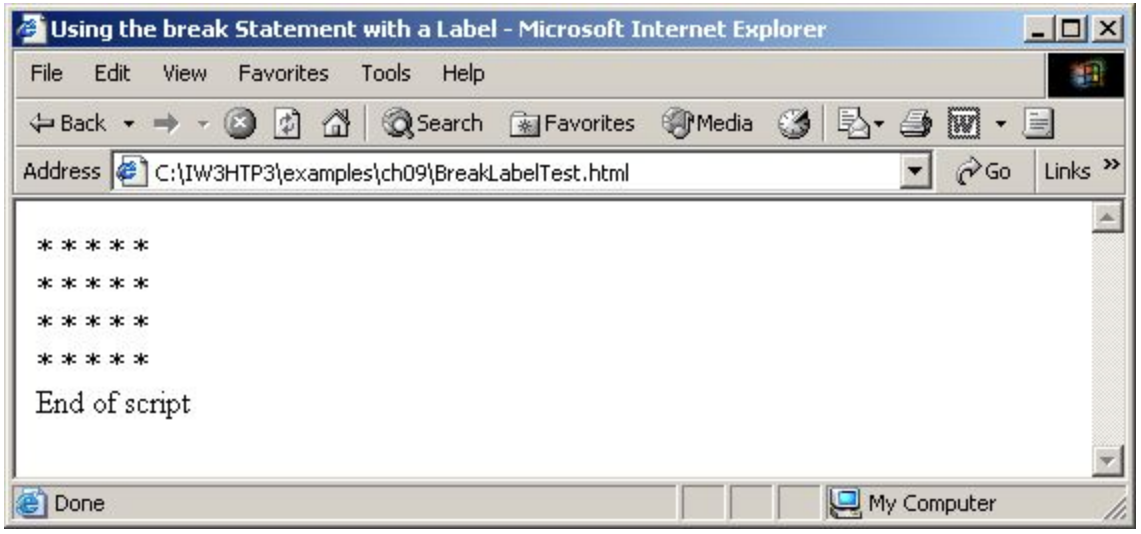
Outline

BreakLabelTest.html
(1 of 2)

```
26
27            // the following line is skipped
28            document.writeln( "This line should not print" );
29         }
30
31         document.writeln( "End of script" );
32         // -->
33      </script>
34
35   </head><body></body>
36</html>
```

Outline

BreakLabelTest.html
(2 of 2)

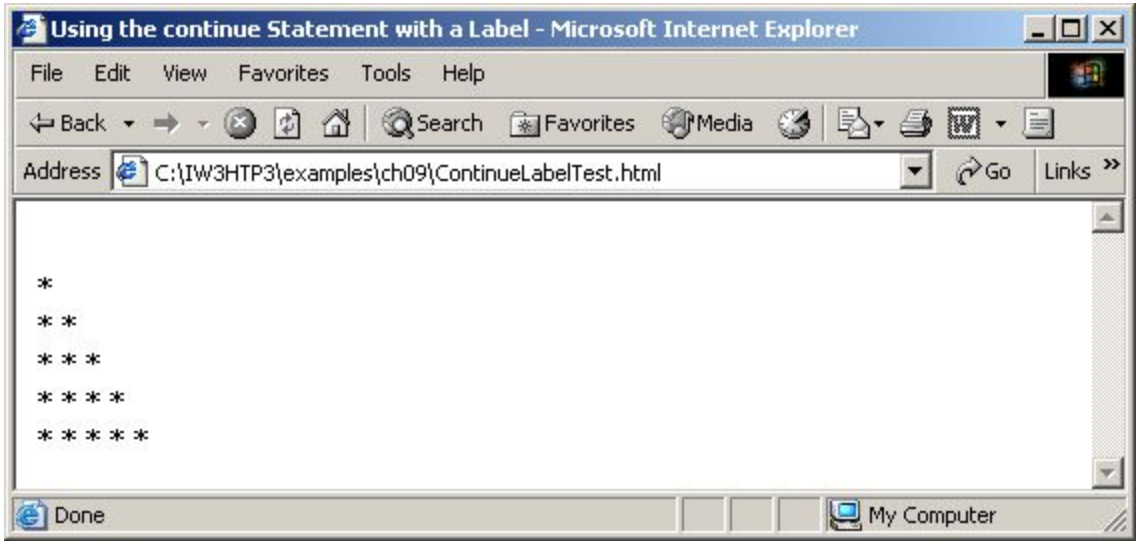


Outline

ContinueLabelTest.html
(1 of 2)

```
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. 9.14: ContinueLabelTest.html -->
6  <!-- Using the continue statement      -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9     <head>
10       <title>Using the continue Statement with a Label</title>
11
12       <script type = "text/javascript">
13          <!--
14          nextRow:    // target label of continue statement
15             for ( var row = 1; row <= 5; ++row ) {
16                document.writeln( "<br />" );
17
18                for ( var column = 1; column <= 10; ++column ) {
19
20                   if ( column > row )
21                      continue nextRow; // next iteration of
22                                        // labeled loop
23
24                   document.write( "* " );
25                }
```

```
26            }
27         // -->
28      </script>
29
30   </head><body></body>
31 </html>
```



Outline

ContinueLabelTest.html
(2 of 2)

9.9 Logical Operators

- More logical operators
  - Logical AND (&&)
  - Logical OR (||)
  - Logical NOT (!)

9.9 Logical Operators

expression1	expression2	expression1 && expression2
false	false	false
false	true	false
true	false	false
true	true	true

Fig. 9.15 Truth table for the && (logical AND) operator.

9.9 Logical Operators

expression1	expression2	expression1 || expression2
false	false	false
false	true	true
true	false	true
true	true	true

Fig. 9.16 Truth table for the || (logical OR) operator.

expression	! expression
false	true
true	false

Fig. 9.17 Truth table for operator ! (logical negation).

Outline

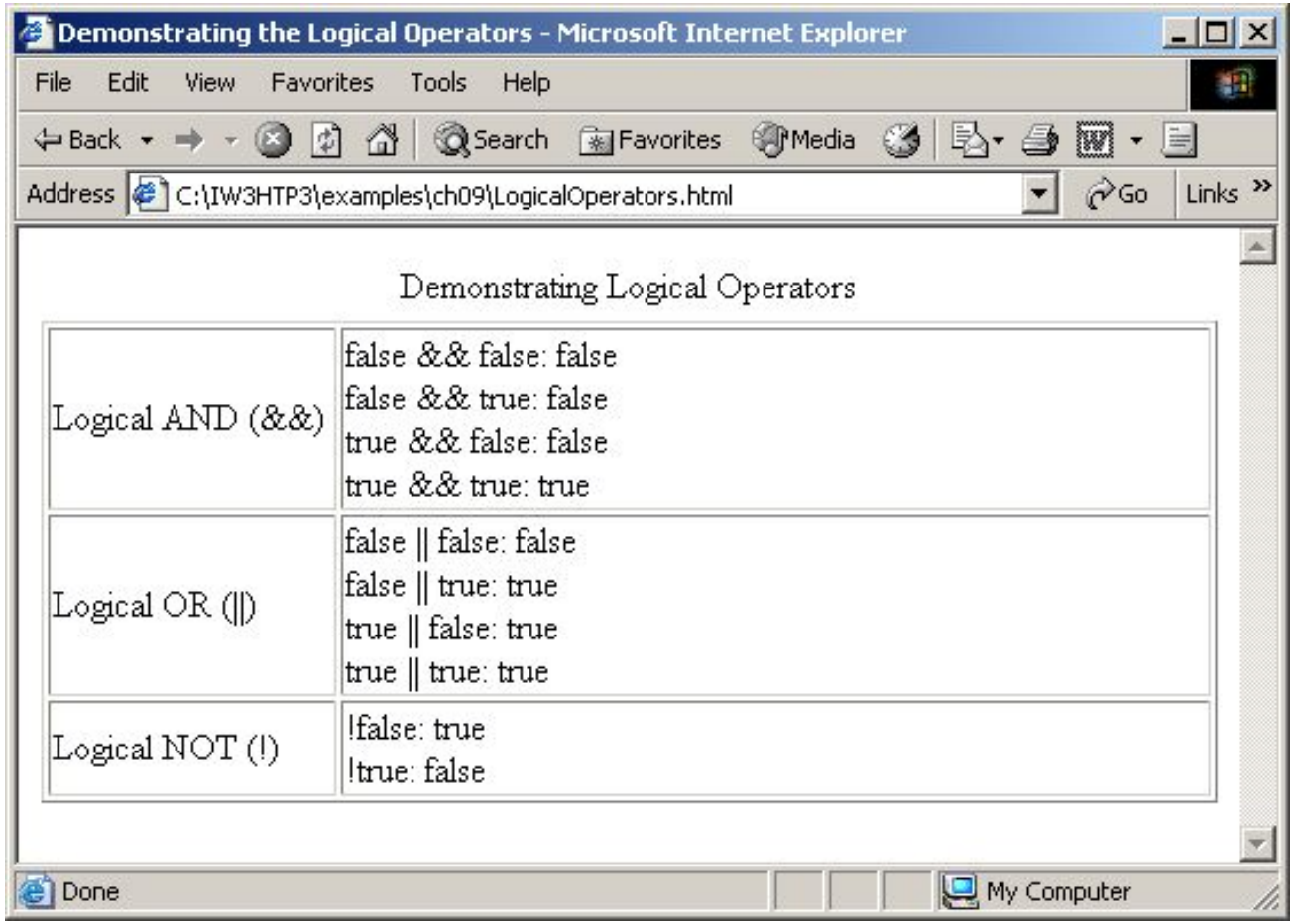
LogicalOperators.html
(1 of 2)

```
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. 9.18: LogicalOperators.html  -->
6  <!-- Demonstrating Logical Operators    -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9     <head>
10       <title>Demonstrating the Logical Operators</title>
11
12       <script type = "text/javascript">
13          <!--
14          document.writeln(
15             "<table border = \"1\" width = \"100%\">" );
16
17          document.writeln(
18             "<caption>Demonstrating Logical " +
19             "Operators</caption" );
20
21          document.writeln(
22             "<tr><td width = \"25%\">Logical AND (&&)</td>" +
23             "<td>false && false: " + ( false && false ) +
24             "<br />false && true: " + ( false && true ) +
25             "<br />true && false: " + ( true && false ) +
```

```
26            "<br />true && true: " + ( true && true ) +
27            "</td>" );
28
29         document.writeln(
30            "<tr><td width = \"25%\">Logical OR (||)</td>" +
31            "<td>false || false: " + ( false || false ) +
32            "<br />false || true: " + ( false || true ) +
33            "<br />true || false: " + ( true || false ) +
34            "<br />true || true: " + ( true || true ) +
35            "</td>" );
36
37         document.writeln(
38            "<tr><td width = \"25%\">Logical NOT (!)</td>" +
39            "<td>!false: " + ( !false ) +
40            "<br />!true: " + ( !true ) + "</td>" );
41
42         document.writeln( "</table>" );
43         // -->
44      </script>
45
46   </head><body></body>
47 </html>
```

Outline

LogicalOperators.html
(2 of 2)


Demonstrating the Logical Operators - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address C:\IW3HTP3\examples\ch09\LogicalOperators.html
Go Links
Demonstrating Logical Operators
Logical AND (&&)
false && false: false
false && true: false
true && false: false
true && true: true
Logical OR (||)
false || false: false
false || true: true
true || false: true
true || true: true
Logical NOT (!)
!false: true
!true: false
Done
My Computer