

# Chapter 8 - JavaScript: Control Structures I

1

## Outline

- 8.1 Introduction
- 8.2 Algorithms
- 8.3 Pseudocode
- 8.4 Control Structures
- 8.5 if Selection Structure
- 8.6 if/else Selection Structure
- 8.7 while Repetition Structure
- 8.8 Formulating Algorithms: Case Study 1 (Counter-Controlled Repetition)
- 8.9 Formulating Algorithms with Top-Down, Stepwise Refinement: Case Study 2 (Sentinel-Controlled Repetition)
- 8.10 Formulating Algorithms with Top-Down, Stepwise Refinement: Case Study 3 (Nested Control Structures)
- 8.11 Assignment Operators
- 8.12 Increment and Decrement Operators
- 8.13 Note on Data Types
- 8.14 JavaScript Internet and World Wide Web Resources

© 2001 Prentice Hall, Inc. All rights reserved.



2

## 8.4 Control Structures

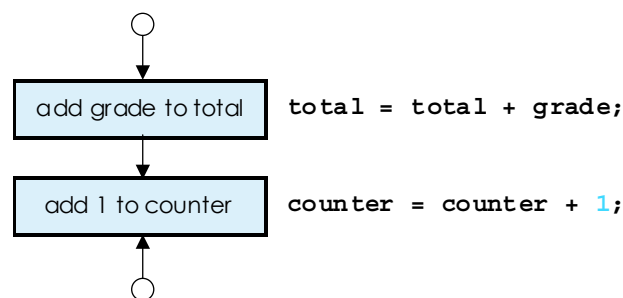


Fig. 8.1 Flowcharting JavaScript's sequence structure.

© 2001 Prentice Hall, Inc. All rights reserved.



# 8.4 Control Structures

JavaScript Keywords				
break	case	continue	delete	do
else	false	for	function	if
in	new	null	return	switch
this	true	typeof	var	void
while	with			
Keywords that are reserved, but not used by JavaScript				
catch	class	const	debugger	default
enum	export	extends	finally	import
super	try			

**Fig. 8.2** JavaScript keywords.

# 8.6 if/else Selection Structure

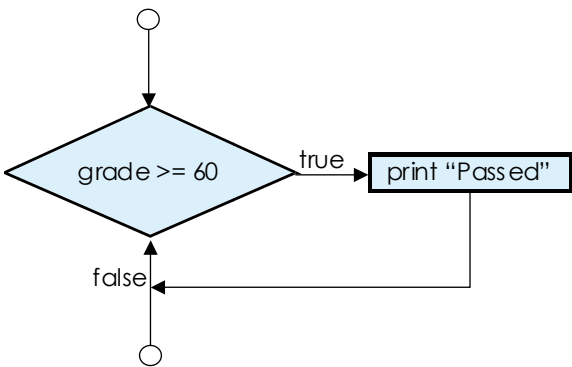


Fig. 8.3 Flowcharting the single-selection if structure.

## 8.6 if/else Selection Structure

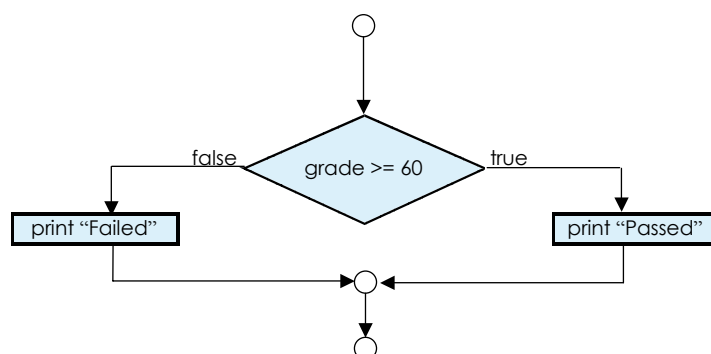


Fig. 8.4 Flowcharting the double-selection if/else structure.



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

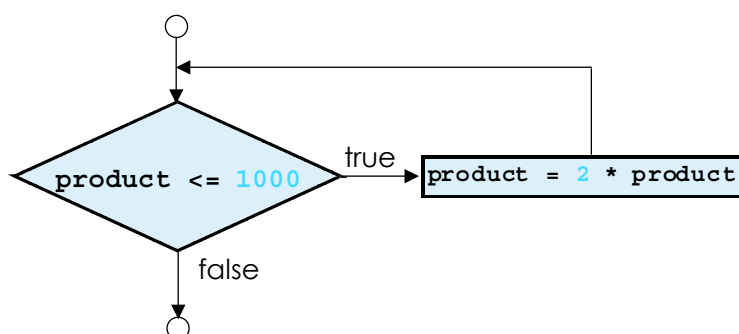


Fig. 8.5 Flowcharting the while repetition structure.





```

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,           // $
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
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

```

The **while** loop will execute the statements in the body of the loop until the value of **gradeCounter** equals 10.

Prompt for the user input a grade.

Convert input to an integer.

Add new grade to **total**.

© 2001 Prentice Hall, Inc.  
All rights reserved.



```

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
44   document.writeln(
45     "<h1>Class average is " + average + "</h1>"
46   );
47 </script>
48
49 </head>
50 <body>
51   <p>Click Refresh (or Reload) to run the script again</p>
52 </body>
53 </html>

```

Increment the counter.

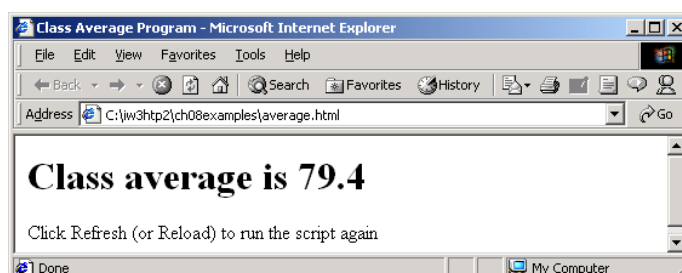
Calculate the average of the grades input by the user.

Write the result to the XHTML document.



This dialog is displayed 10 times. User input is 100, 88, 93, 55, 68, 77, 83, 95, 73 and 62.

### 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. 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
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;

```

Prompt for the user to enter a grade, -1 to end.

The **while** loop will continue until the user input equals -1.

© 2001 Prentice Hall, Inc.  
All rights reserved.



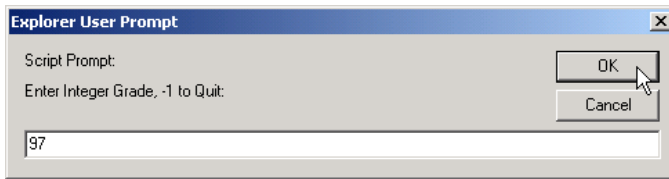
```

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
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>

```

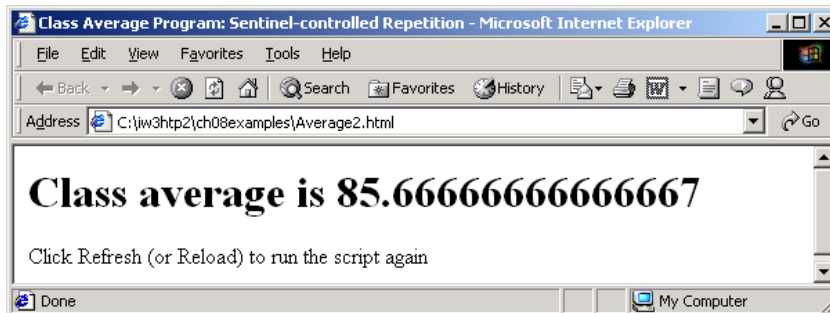
Each iteration of the loop will open a prompt dialog allowing the user to input another grade.

© 2001 Prentice Hall, Inc.  
All rights reserved.



This dialog is displayed four times.  
User input is 97, 88, 72 and -1.

## Program Output



```
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/
9    <head>
10     <title>Analysis of Examination Results</title>
11
12     <script type = "text/javascript">
13       <!--
14       // initializing variables in declar
15       var passes = 0,      // number of
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       {
23         result = window.prompt(
24           "Enter result (1=pass,2=fa
25
26         if ( result == "1" )
27           passes = passes + 1;
28         else
29           failures = failures + 1;
30         student = student + 1;
31       }
32
```

The **while** loop will continue until the value of **student** is 10 meaning 10 results were entered.

Entering a 1 into the prompt dialog means the student passed the exam. A value of 2 means the student failed.

If a value of 1 was entered, the value of **passes** is incremented by one, otherwise, **failures** is incremented.



## Outline

Analysis.html

```

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>

```

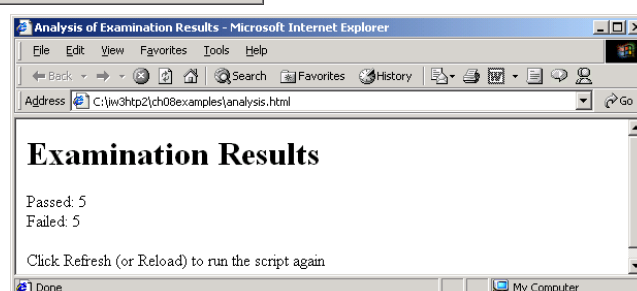
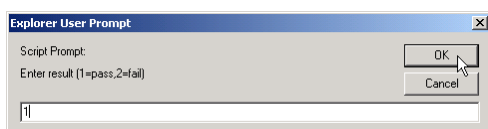
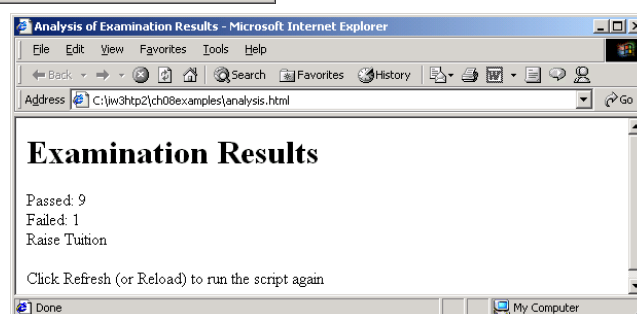
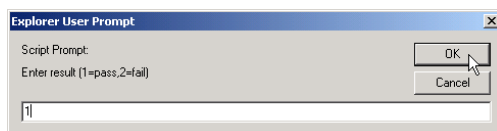
If more than 8 students passed the exam, the program says to "Raise Tuition".

© 2001 Prentice Hall, Inc.  
All rights reserved.



## Outline

Program Output



© 2001 Prentice Hall, Inc.  
All rights reserved.

## 8.12 Increment and Decrement Operators

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

**Fig. 8.12** Arithmetic assignment operators.

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

**Fig. 8.13** increment and decrement operators.



```

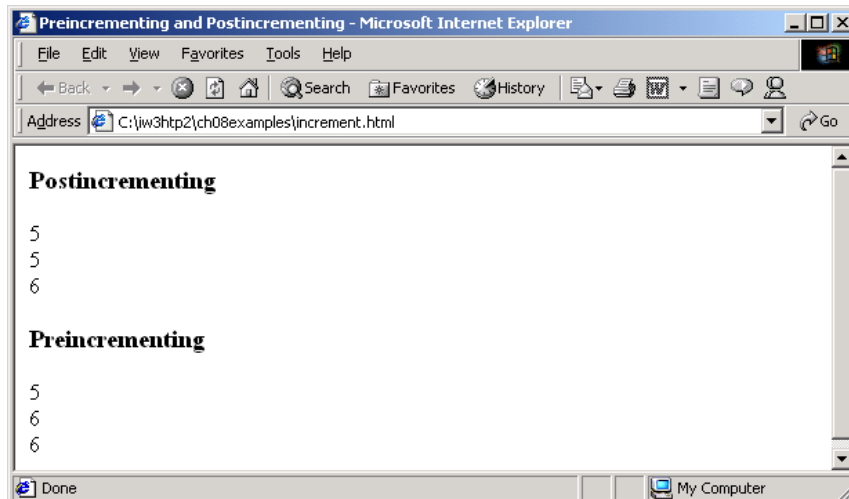
1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5  <!-- Fig. 8.14: increment.html -->
6  <!-- Preincrementing and Postincrementing -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9    <head>
10     <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 );
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>

```

Postincrementing the variable will print the variable and then increment the value by one.

Preincrementing the variable will increment the value by one and then print the value.





## Program Output

© 2001 Prentice Hall, Inc.  
All rights reserved.

## 8.13 Note on Data Types

Operators	Associativity	Type
( )	left to right	parentheses
++ --	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.

