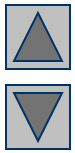
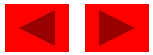


Outline

JavaScript: Control Structure

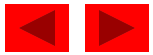
Algorithms

- Actions to be executed
- Order in which the actions are to be executed
- Pseudocode: informal representation of an algorithm
  - If prepared carefully, pseudocode statements can be converted in to actual programming code in a fairly straightforward fashion



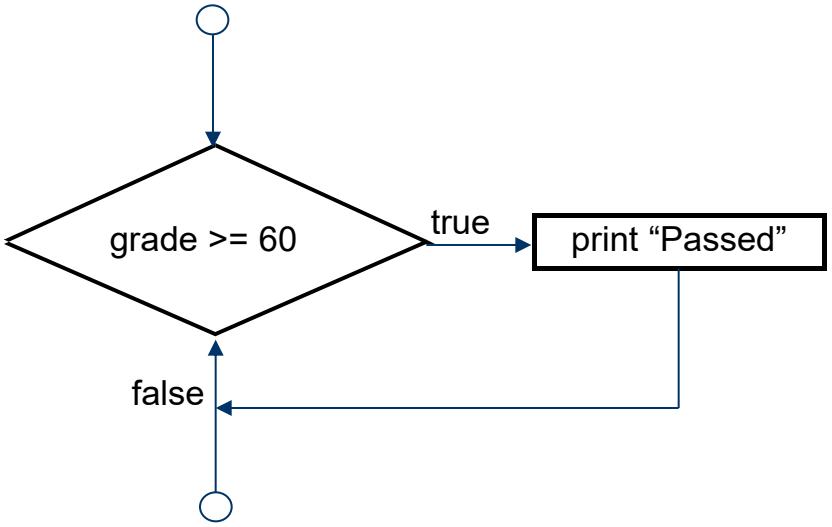
Control Structures

- Elements of code that define an individual action
- Like most programming languages, JavaScript has three control structures:
  - Sequence structure
    - Any individual non-selection and non-repetition statement falls into this category: individual calculation, input or output statement, type conversion, etc.
  - Selection structure: three in JavaScript
    - `if`
    - `if...else`
    - `switch`
  - Repetition structure: four in JavaScript
    - `while`
    - `do...while`
    - `for`
    - `for...in`



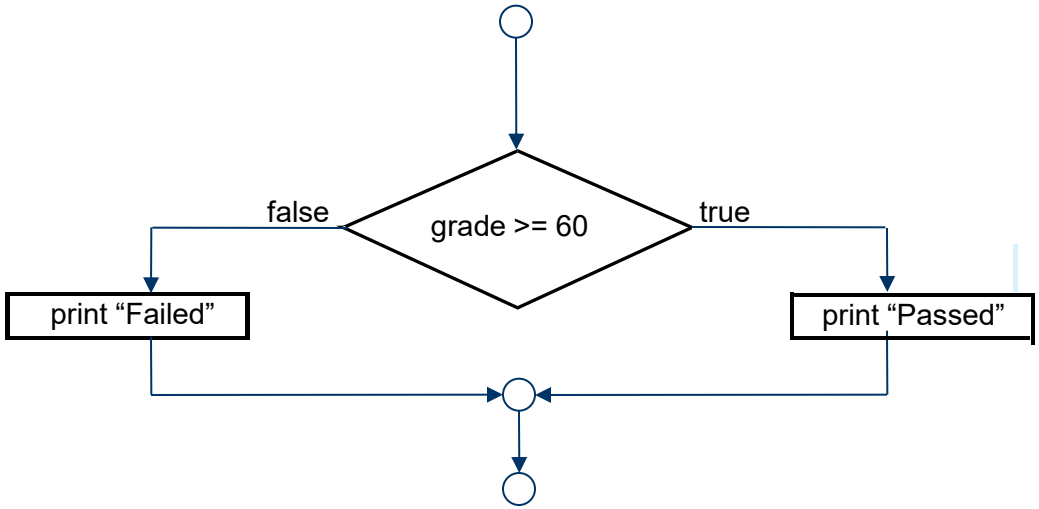
`if` Selection Statement

- Single-entry/single-exit structure
- Indicate action only when the condition evaluates to `true`. No action for `false`



`if…else` Selection Statement

- Indicate different actions to be perform when condition is `true` or `false`



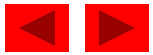
- Conditional operator (`?:`) (see page 217), closely related to `if…else`
  - JavaScript’s only so called “ternary” operator
    - Three operands
    - Forms a conditional expression

Nested `if…else` Selection Statement

- When we have one decision criterion but with multiple and mutually exclusive range of values

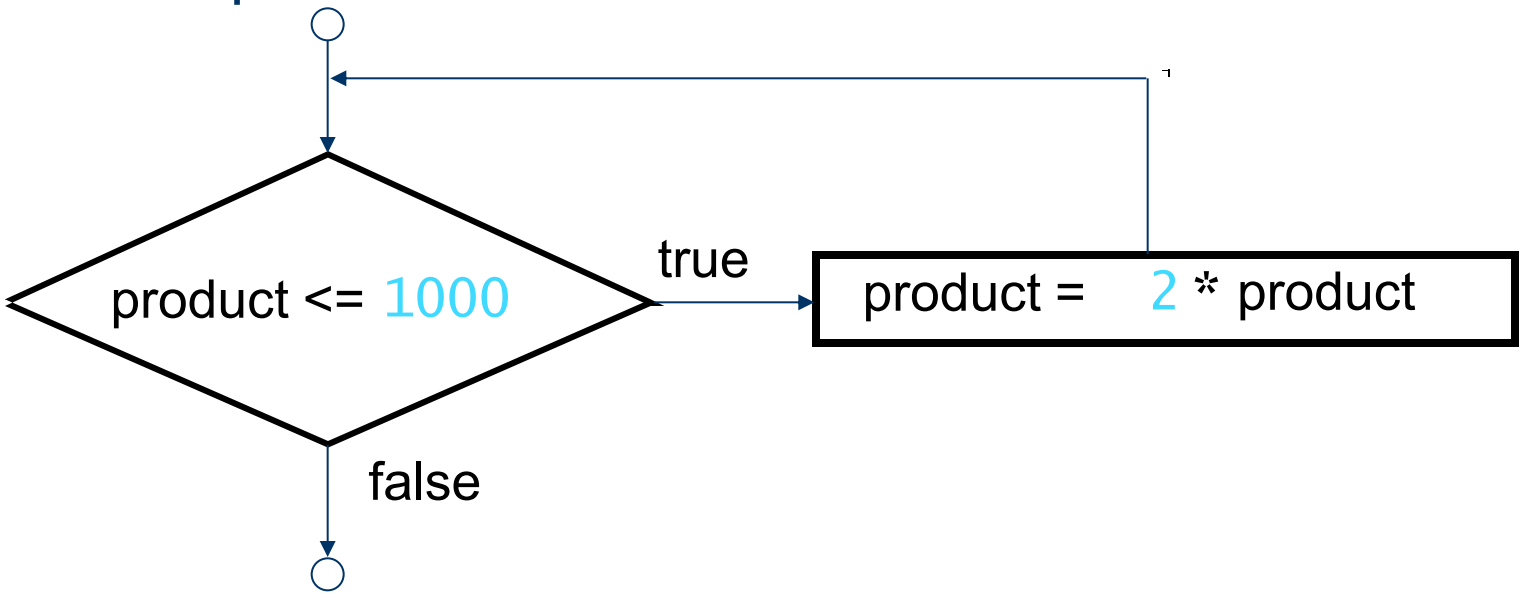
```
If student = “Senior” …
Else if student = “Junior” …
Else if student = “Sophomore” …
Else …
```

  - `Switch` clause can be used instead
- When we have more than one decision criterion
  - for example when making decisions based on combined values of variable “age” and “income”:
  - Logic errors vs. syntax errors
  - Can be simplified by using logical AND (&&) , OR (||) operators
    - In class example



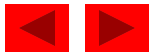
while Repetition Statement

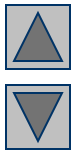
- Repetition structure (loop)
  - Repeat action while some condition remains true



Formulating Algorithms: Example 1 (Counter-Controlled Repetition)

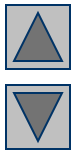
- Counter-controlled repetition
  - Counter
    - Control the number of times a set of statements executes
  - Definite repetition





Outline

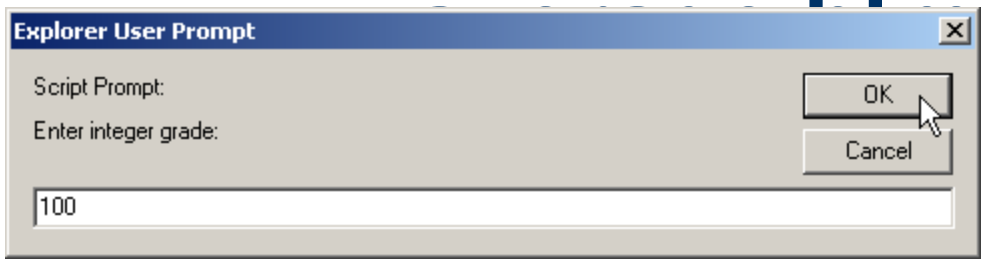
```
1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5  <!-- Fig. 8.7: average.html -->
6  <!-- Class Average Program  -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9     <head>
10       <title>Class 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
```



Outline

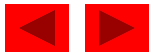
```
24         // Processing Phase
25         while ( gradeCounter <= 10 ) {  // loop 10 times
26
27            // prompt for input and read grade from user
28            grade = window.prompt( "Enter integer grade:", "0" );
29
30            // convert grade from a string to an integer
31            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>
```

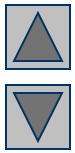
```
48
49     </head>
50     <body>
51        <p>Click Refresh (or Reload) to run the script again<p>
52     </body>
53  </html>
```



Example 2 (Sentinel-Controlled Repetition)

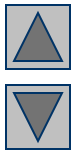
- Indefinite repetition
  - Sentinel value indicates the end of data entry: should be out of range of acceptable values





Outline

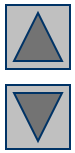
```
1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5  <!-- Fig. 8.9: average2.html        -->
6  <!-- Sentinel-controlled Repetition -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9     <head>
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
```



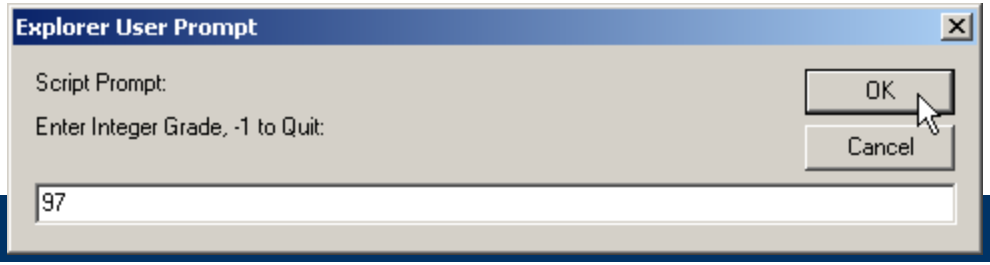
Outline

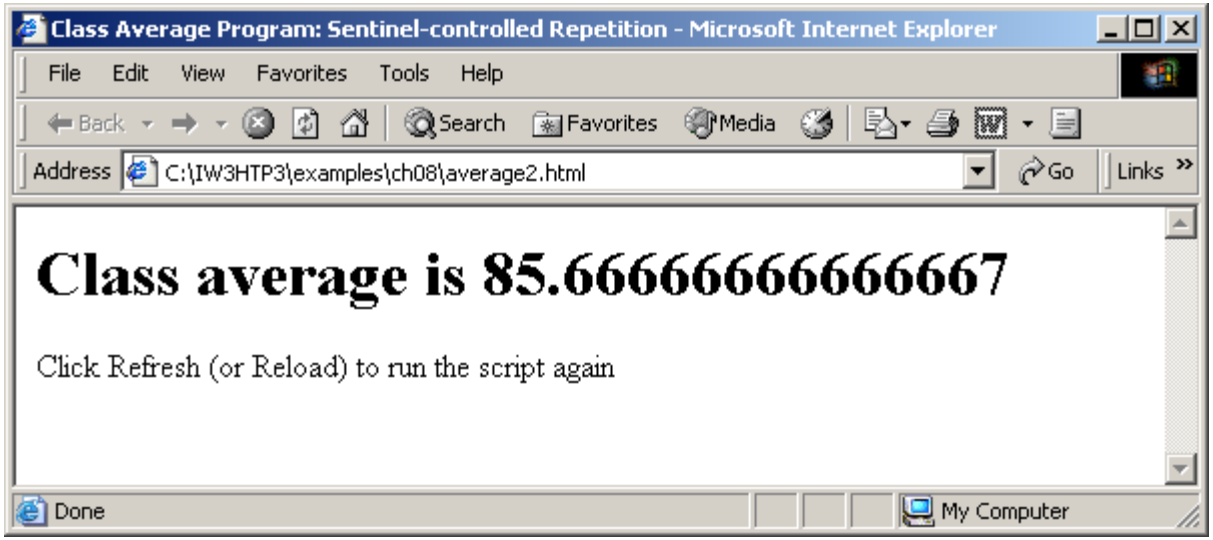
```
25          // Processing phase
26          // prompt for input and read grade from user
27          grade = window.prompt(
28               "Enter Integer Grade, -1 to Quit:", "0" );
29
30          // convert grade from a string to an integer
31          gradeValue = parseInt( grade );
32
33          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
```

```
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>
```



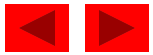
Outline

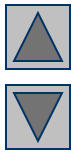

Explorer User Prompt
Script Prompt:
Enter Integer Grade, -1 to Quit:
OK
Cancel
97


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

Example 3 (Nested Control Structures)

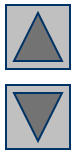
- Consider problem
- Make observations
- Top-down, stepwise refinement





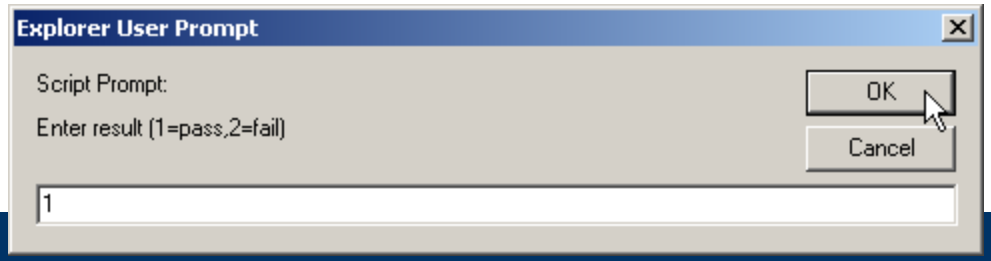
Outline

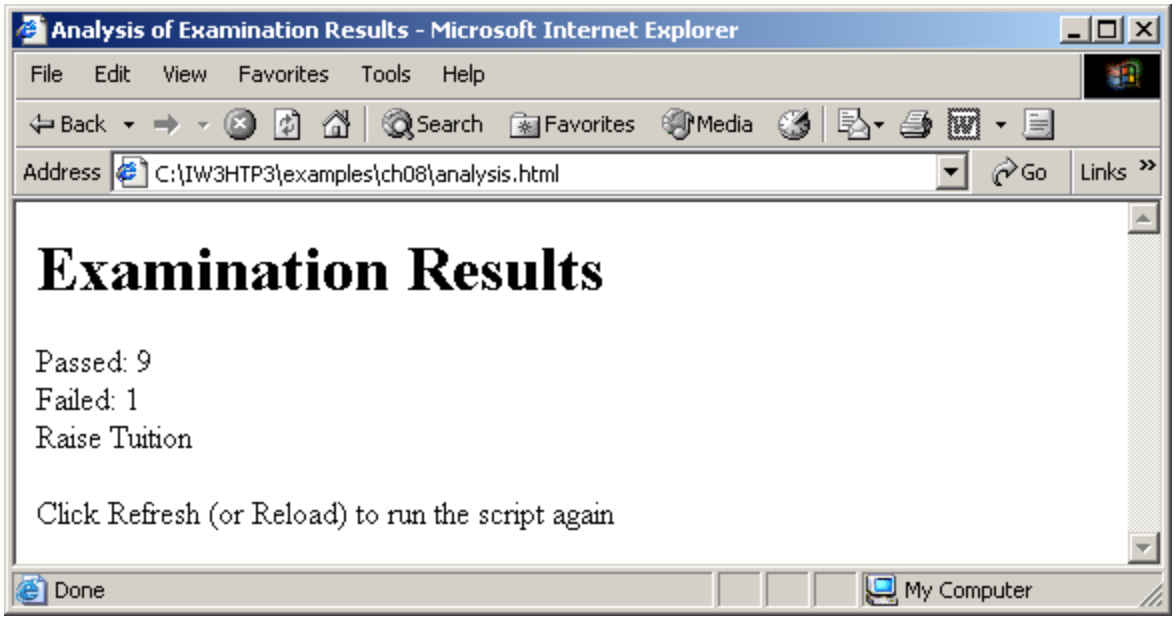
```
1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5  <!-- Fig. 8.11: analysis.html -->
6  <!-- Analyzing Exam Results   -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9     <head>
10       <title>Analysis 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
```

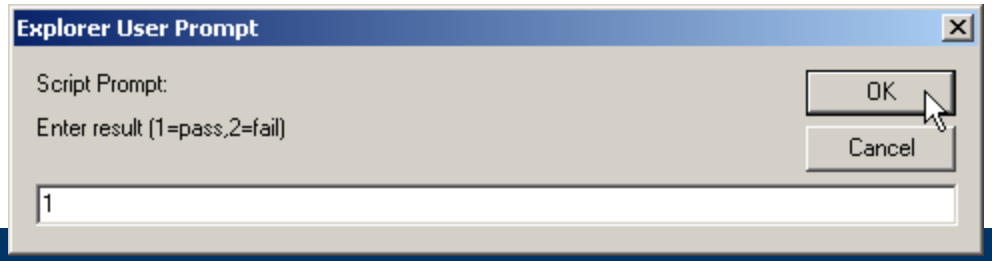


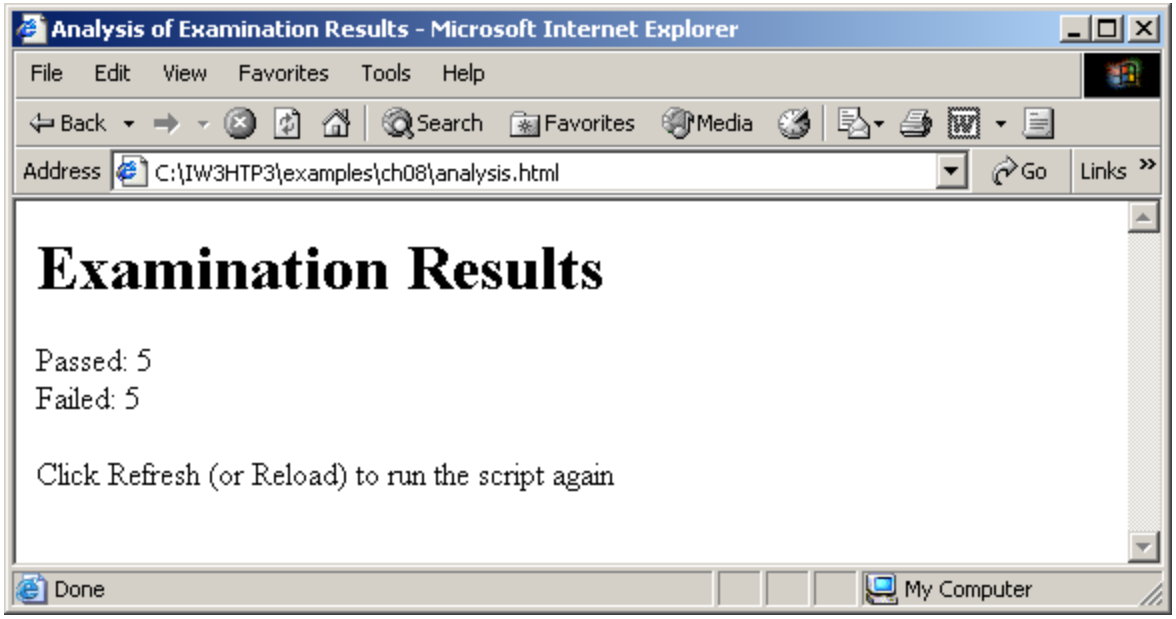
Outline

```
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>
```


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

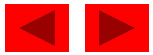

Analysis of Examination Results - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address C:\IW3HTP3\examples\ch08\analysis.html
Go Links
Examination Results
Passed: 9
Failed: 1
Raise Tuition
Click Refresh (or Reload) to run the script again
Done
My Computer


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

Assignment Operators

- Compound assignment operators
  - Abbreviate assignment expressions



8.13 Note on Data Types

- Loosely typed
  - Automatically converts between values of different types