JavaScript

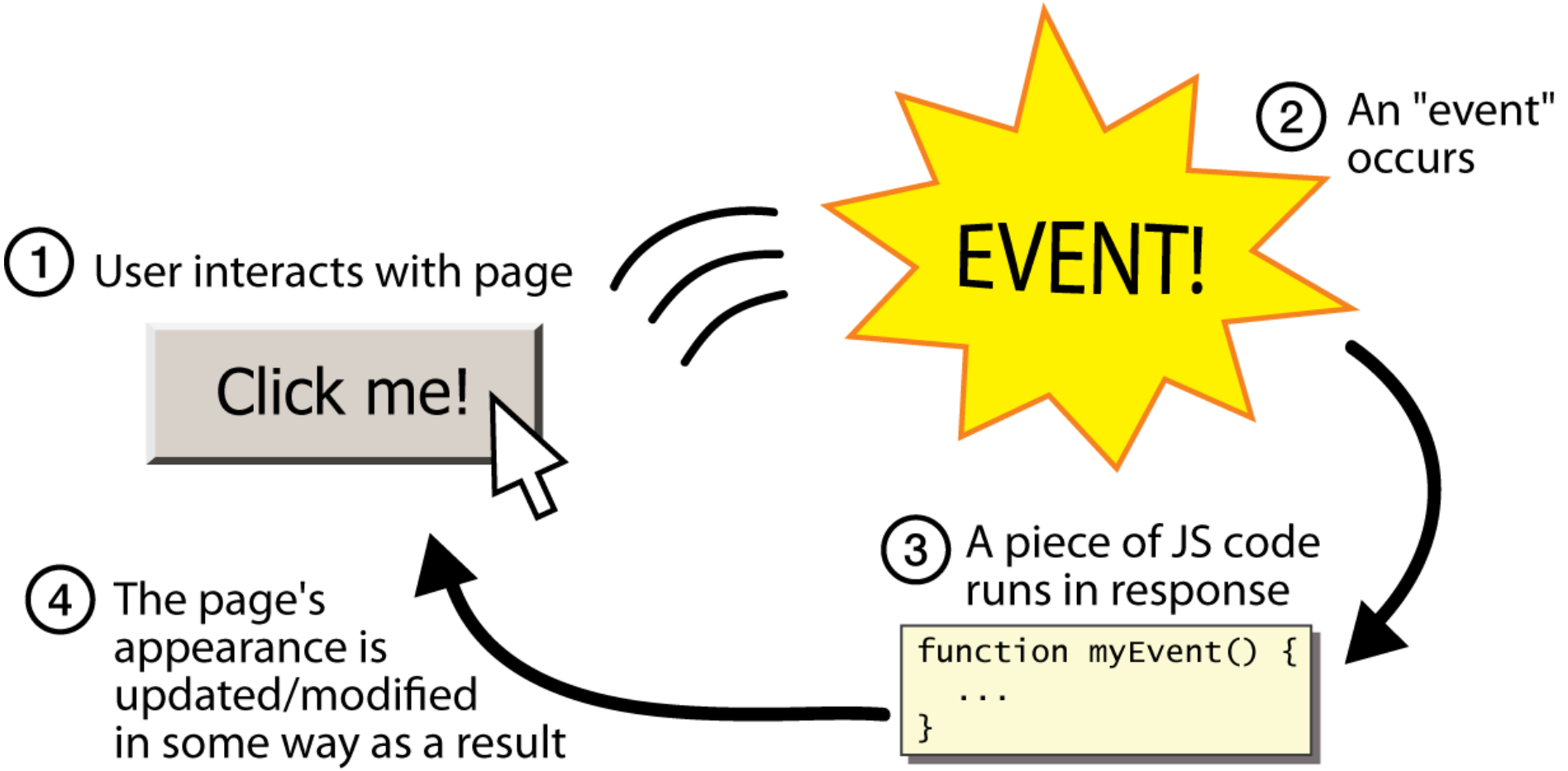
Introduction

- What is JavaScript?
  - It is designed to add interactivity to HTML pages
  - It is a **scripting language** (a lightweight programming language)
  - It is an **interpreted language** (it executes without preliminary compilation)
  - Usually embedded directly into HTML pages

Introduction

- JavaScript gives HTML designers a programming tool:
  - simple syntax
- JavaScript
  - can put dynamic text into an HTML page
  - can react to events
  - can read and write HTML elements
  - can be used to validate data
  - can be used to detect the visitor's browser
  - can be used to create cookies
  - Store and retrieve information on the visitor's computer

Event-driven programming



Introduction

- The HTML <script> tag is used to insert a JavaScript into an HTML page

```
<script type="text/javascript">
    document.write("Hello World!");
</script>
```

- Ending statements with a semicolon?
  - Optional; required when you want to put multiple statements on a single line
- JavaScript can be inserted within the head, the body, or use external JavaScript file

External JavaScript

- `<script type=“text/javascript” src=“filename”></script>`

JavaScript Basics

- Variables
- If … Else
- Switch
- Operators
- Popup Boxes
- Functions
- Loops (for, while)
- Events
- Try … Catch
- Throw

JavaScript Basics

- Java Objects:
  - String
  - Date
  - Array
  - Boolean
  - Math
  - RegExp
  - HTML DOM

Variable

- Variables are used to store data.
- A variable is a "container" for information you want to store. A variable's value can change during the script. You can refer to a variable by name to see its value or to change its value.
- Rules for variable names:
  - Variable names are case sensitive
  - They must begin with a letter or the underscore character
    - strname – STRNAME (not same)

Variables

- `var clientName = “ABC";`
- `var age = 32;`
- `var weight = 127.4;`
- `var enrollment = 99;`
- `var medianGrade = 2.8;`
- `var credits = 5 + 4 + (2 * 3);`

Variables

- integers and real numbers are the same type (no int vs. double)
- same operators: + - * / % ++ -- = += -= *= /= %=

JavaScript Operators

- Arithmetic Operators

Operator	Description	Example	Result
+	Addition	x=2 y=2 x+y	4
-	Subtraction	x=5 y=2 x-y	3
*	Multiplication	x=5 y=4 x*y	20
/	Division	15/5 5/2	3 2,5
%	Modulus (division remainder)	5%2 10%8 10%2	1 2 0
++	Increment	x=5 x++	x=6
--	Decrement	x=5 x--	x=4

JavaScript Operators

- Assignment Operator

Operator	Example	Is The Same As
=	x=y	x=y
+=	x+=y	x=x+y
-=	x-=y	x=x-y
*=	x*=y	x=x*y
/=	x/=y	x=x/y
%=	x%=y	x=x%y

JavaScript Operators

- Comparison Operators

Operator	Description	Example
==	is equal to	5==8 returns false
===	is equal to (checks for both value and type)	x=5 y="5" x==y returns true x===y returns false
!=	is not equal	5!=8 returns true
>	is greater than	5>8 returns false
<	is less than	5<8 returns true
>=	is greater than or equal to	5>=8 returns false
<=	is less than or equal to	5<=8 returns true

Logical Operator

- `var ned = null;`
- `var benson = 9;`
- `// at this point in the code,`
- `// ned is null`
- `// benson's 9`
- `// caroline is undefined`

- `undefined` : has not been declared, does not exist
- `null` : exists, but was specifically assigned an empty or null value

Logical Operator

- > < >= <= && || ! == != === !==
- most logical operators automatically convert types:
  - 5 < "7" is true
  - 42 == 42.0 is true
  - "5.0" == 5 is true
- === and !== are strict equality tests; checks both type and value
  - "5.0" === 5 is false

Popup boxes

- `alert("message"); // message`
- `confirm("message"); // returns true or false`
- `prompt("message"); // returns user input string`

Comments

- `// single-line comment`
- `/* multi-line comment */`

JavaScript: Control Structures II

if/else statement (same as Java)

```
if (condition) {
  statements;
} else if (condition) {
  statements;
} else {
  statements;
}
```

For Loop

```
var sum = 0;
for (var i = 0; i < 100; i++) {
  sum = sum + i;
}
```

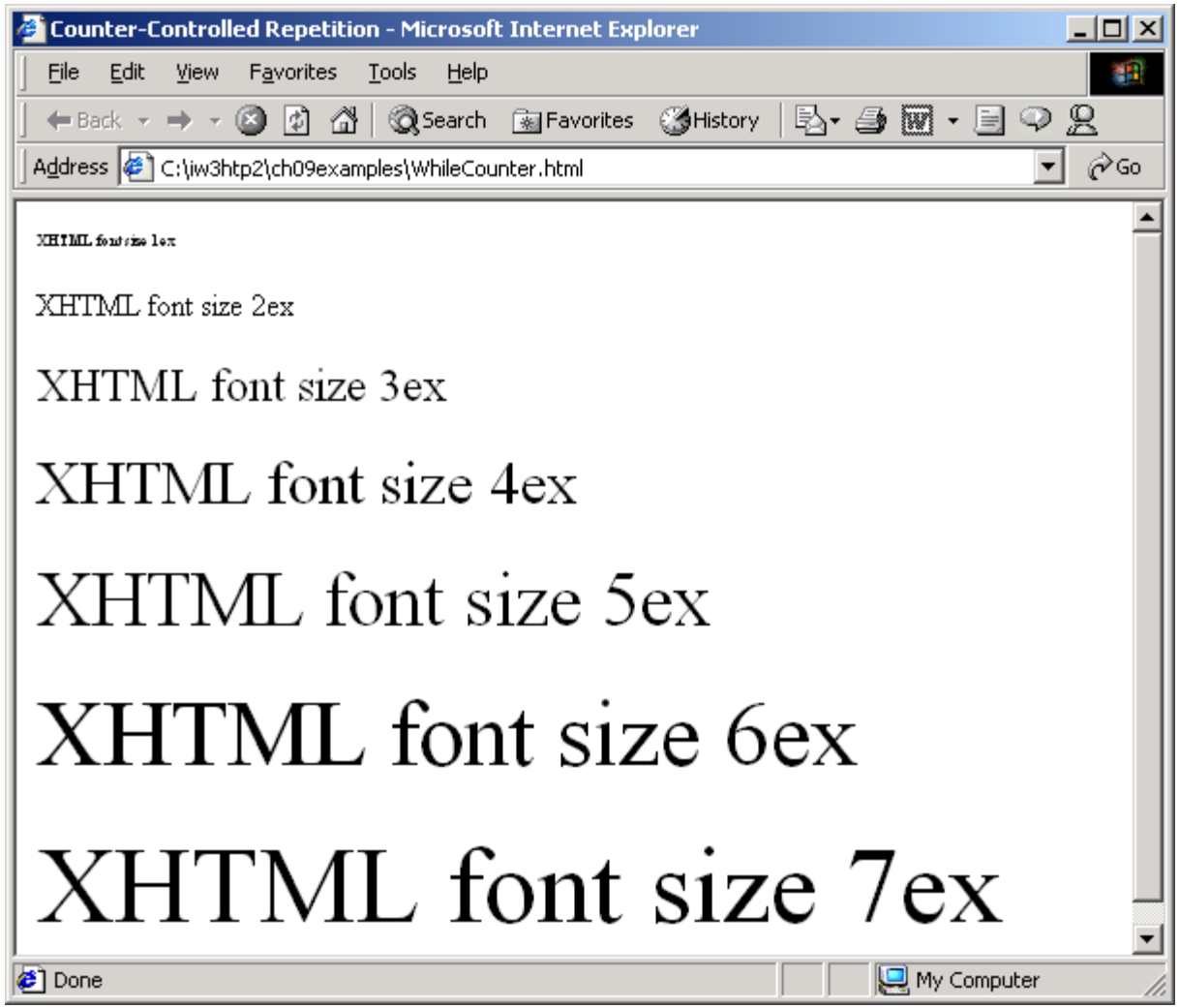
While Loop

```
while (condition) {
  statements;
}

do {
   statements;
} while (condition);
```

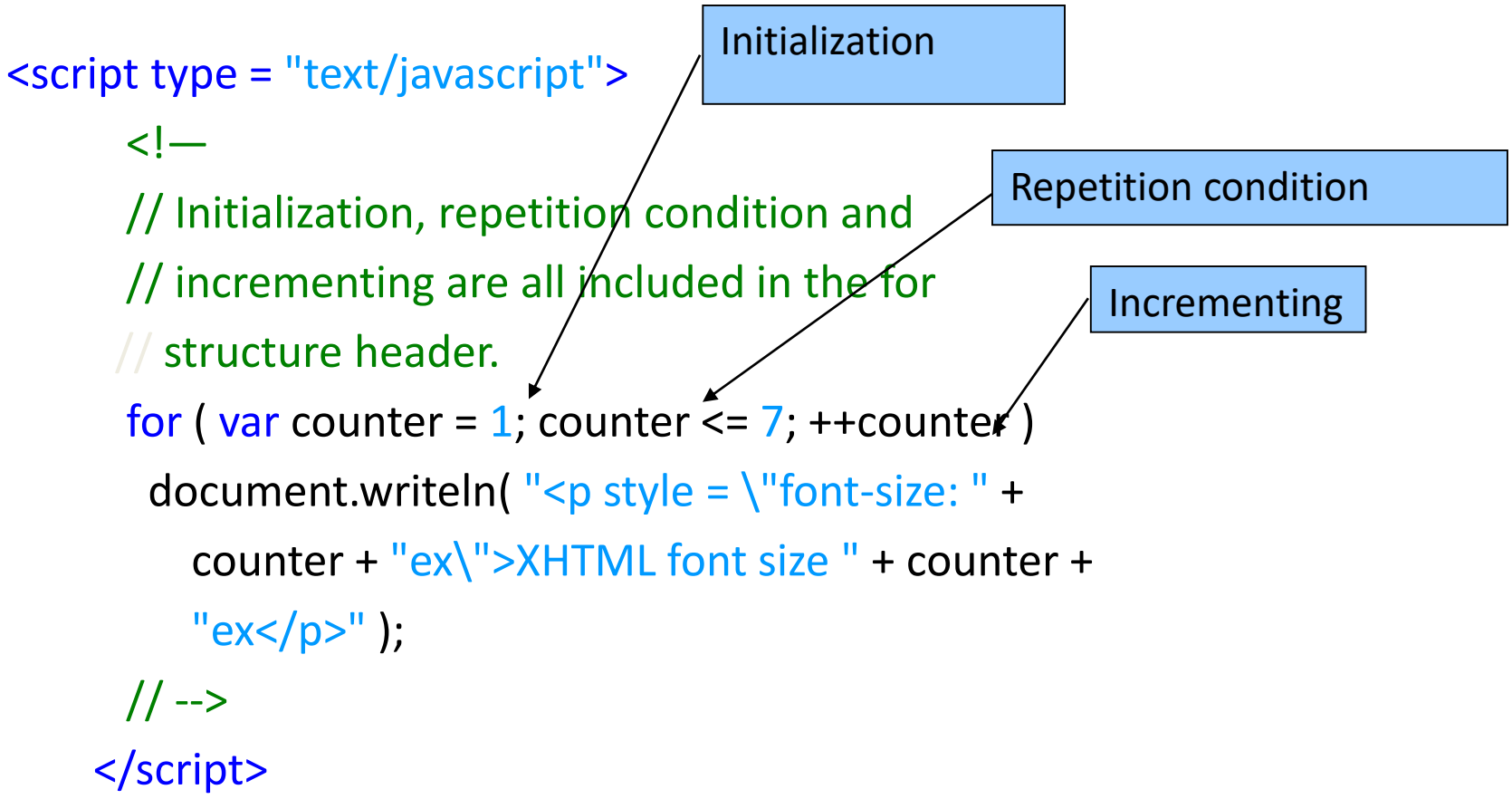
```
<script type = "text/javascript">
  var counter = 1;          // initialization

    while ( counter <= 7 ) {    // repetition condition
      document.writeln( "<p style = \"font-size: " +
        counter + "ex\">XHTML font size " + counter +
        "ex</p>" );
      ++counter;            // increment
    }

  </script>
```

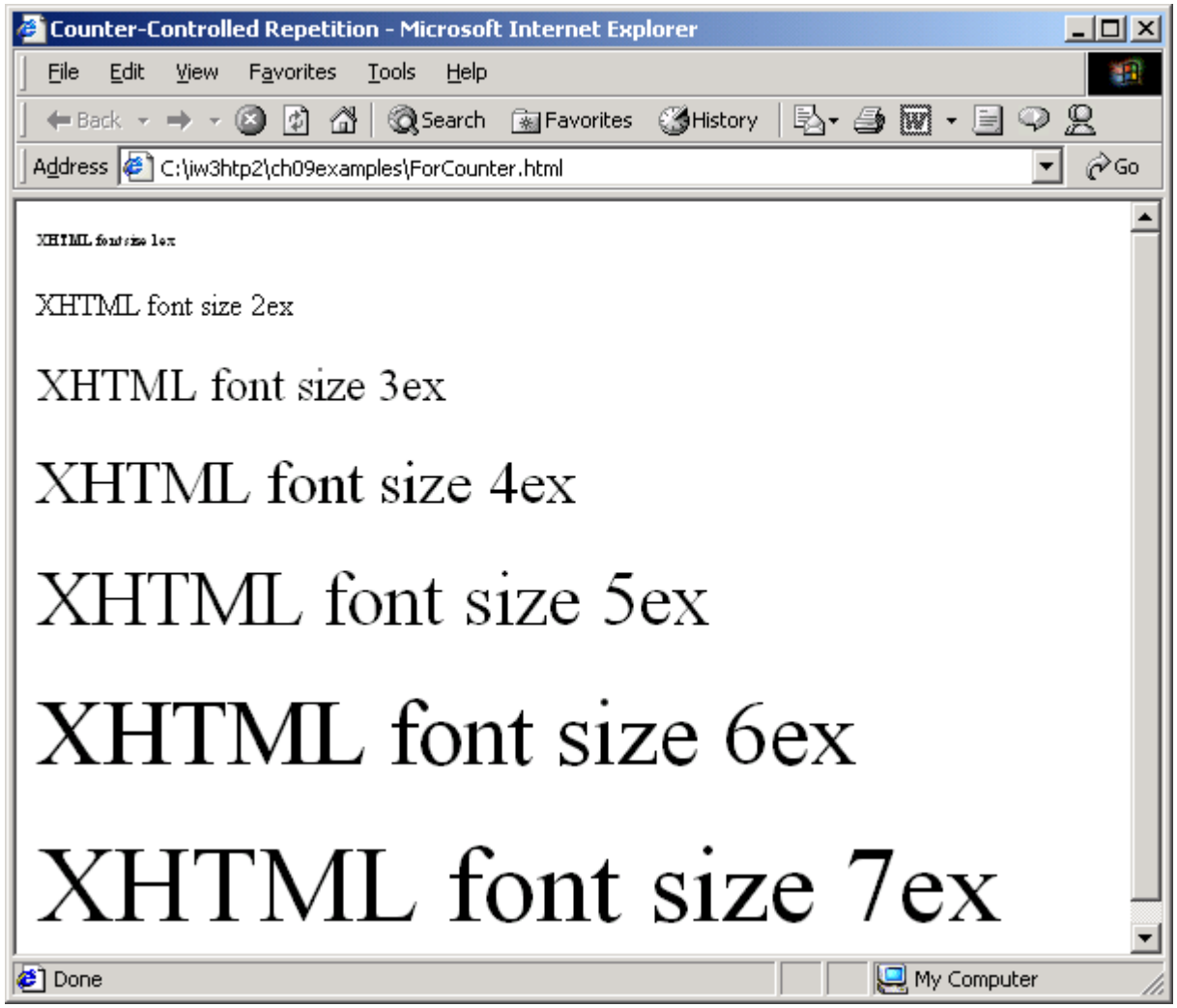
The **`while`** loop will continue until the value of **`counter`** is greater than 7.

Increment the counter.


Counter-Controlled Repetition - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address C:\iw3htp2\ch09examples\WhileCounter.html
Go
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

For Loop


Initialization
Repetition condition
Incrementing
<script type = "text/javascript">
<!—
// Initialization, repetition condition and
// incrementing are all included in the for
// structure header.
for (var counter = 1; counter <= 7; ++counter)
document.writeln("<p style = \"font-size: " +
counter + "ex\">XHTML font size " + counter +
"ex</p>");
// -->
</script>


Counter-Controlled Repetition - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address C:\iw3htp2\ch09examples\ForCounter.html
Go
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

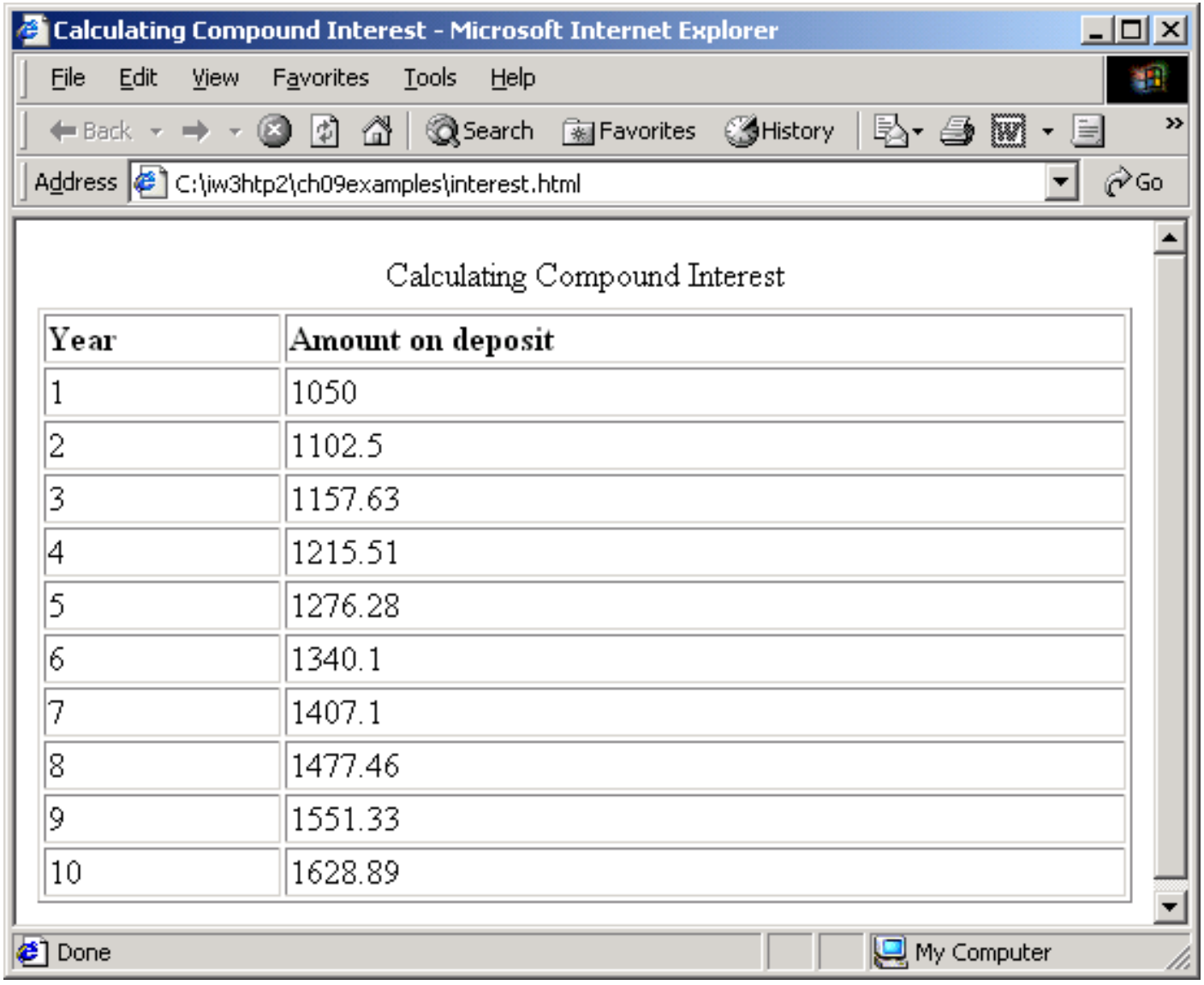
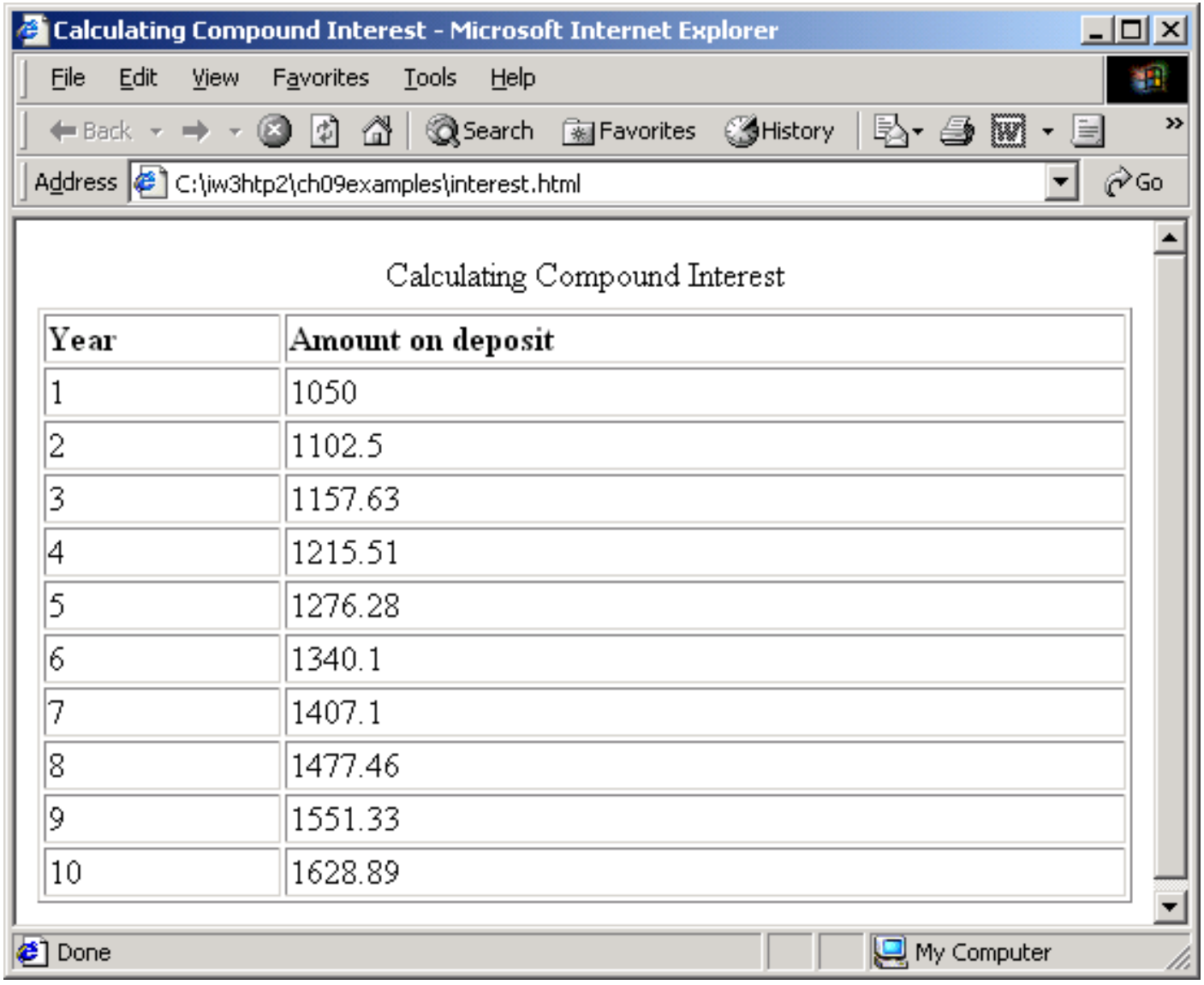
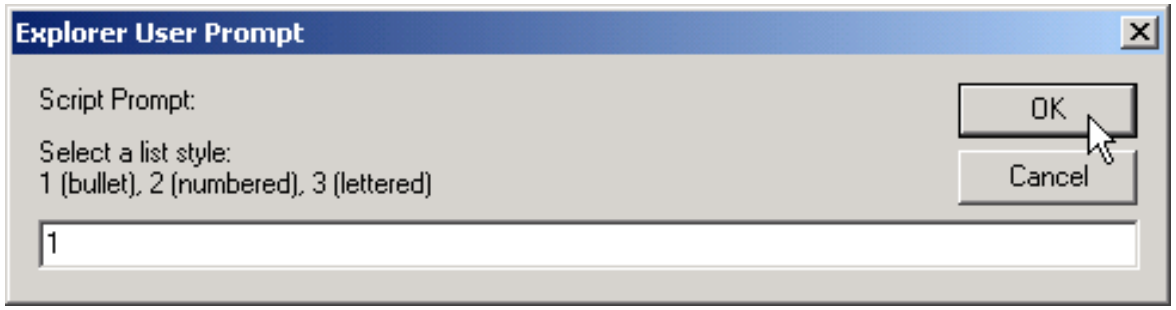

Calculating Compound Interest - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address C:\iw3htp2\ch09examples\interest.html
Go
Calculating Compound Interest
Year
Amount on deposit
1
1050
2
1102.5
3
1157.63
4
1215.51
5
1276.28
6
1340.1
7
1407.1
8
1477.46
9
1551.33
10
1628.89
Done
My Computer

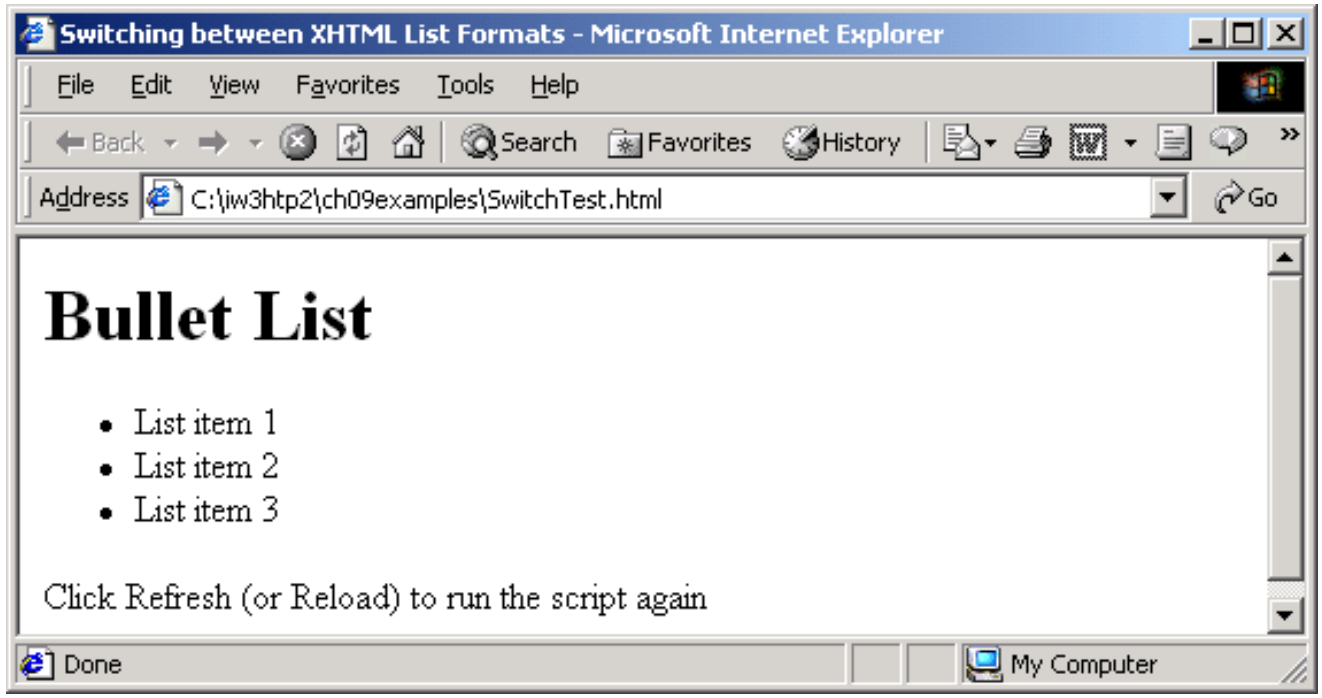
Table using JavaScript

```
<script type = "text/javascript">
  var amount, principal = 1000.0, rate = .05;

  document.writeln("<table border = \"1\" width = \"100%\">" );
  document.writeln( "<caption>Calculating Compound Interest</caption>" );
  document.writeln( "<thead><tr><th align = \"left\">Year</th>" );
  document.writeln("<th align = \"left\">Amount on deposit</th>" );
  document.writeln( "</tr></thead>" );

  for ( var year = 1; year <= 10; ++year ) {
    amount = principal * Math.pow( 1.0 + rate, year );
    document.writeln( "<tbody><tr><td>" + year + "</td><td>" + Math.round(
amount * 100 ) / 100 + "</td></tr>" );
  }
    document.writeln( "</tbody></table>" );
</script>
```


Calculating Compound Interest - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address C:\iw3htp2\ch09examples\interest.html
Go
Calculating Compound Interest
Year
Amount on deposit
1
1050
2
1102.5
3
1157.63
4
1215.51
5
1276.28
6
1340.1
7
1407.1
8
1477.46
9
1551.33
10
1628.89
Done
My Computer

Switch Case


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

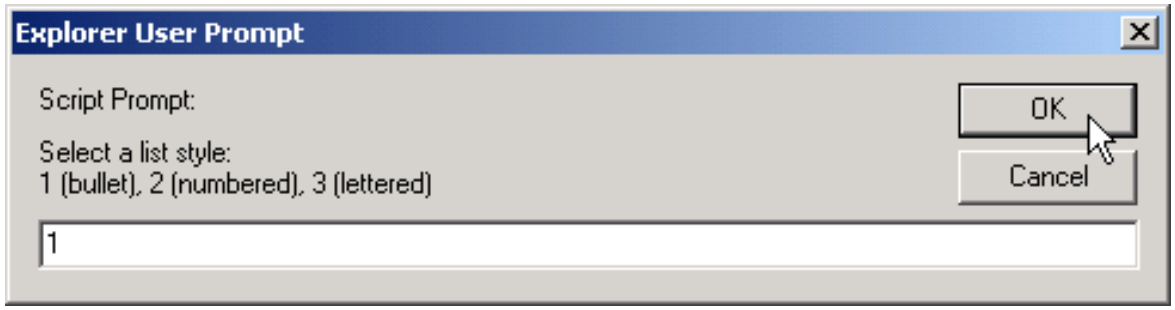

Switching between XHTML List Formats - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address C:\iw3htp2\ch09examples\SwitchTest.html
Go
Bullet List
List item 1
List item 2
List item 3
Click Refresh (or Reload) to run the script again
Done
My Computer

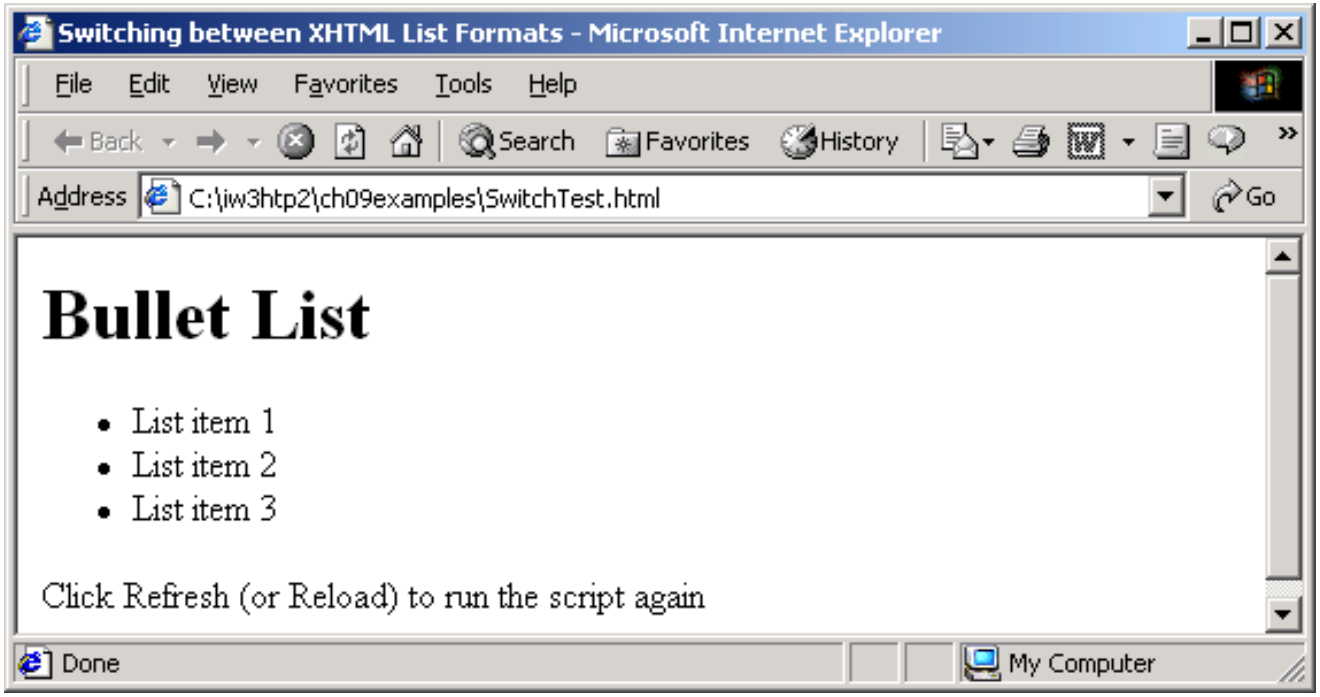
Switch Case

```
<script type = "text/javascript">
        <!—
        var choice,           // user's choice
           startTag,          // starting list item tag
           endTag,            // ending list item tag
           validInput = true,  // indicates if input is valid
           listType;          // list type as a string

        choice = window.prompt( "Select a list style:\n" +
           "1 (bullet), 2 (numbered), 3 (lettered)", "1" );

        switch ( choice ) {
          case "1":
            startTag = "<ul>";
            endTag = "</ul>";
            listType = "<h1>Bullet List</h1>";
            break;
          case "2":
            startTag = "<ol>";
            endTag = "</ol>";
            listType = "<h1>Ordered List: Numbered</h1>";
            break;
```

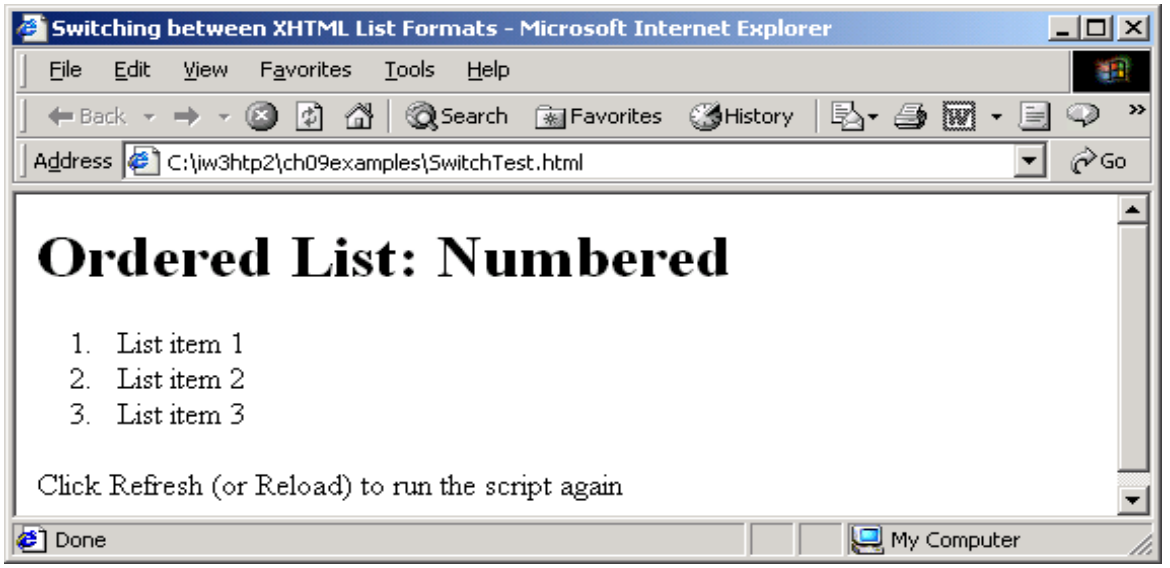
Switch Case

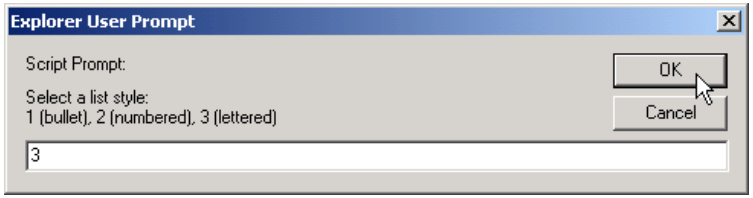
```
case "3":
        startTag = "<ol type = \"A\">";
        endTag = "</ol>";
        listType = "<h1>Ordered List: Lettered</h1>";
        break;
    default:
        validInput = false;
   }

   if ( validInput == true ) {
     document.writeln( listType + startTag );

     for ( var i = 1; i <= 3; ++i )
       document.writeln( "<li>List item " + i + "</li>" );

     document.writeln( endTag );
   }
   else
     document.writeln( "Invalid choice: " + choice );
   // -->
 </script>
```

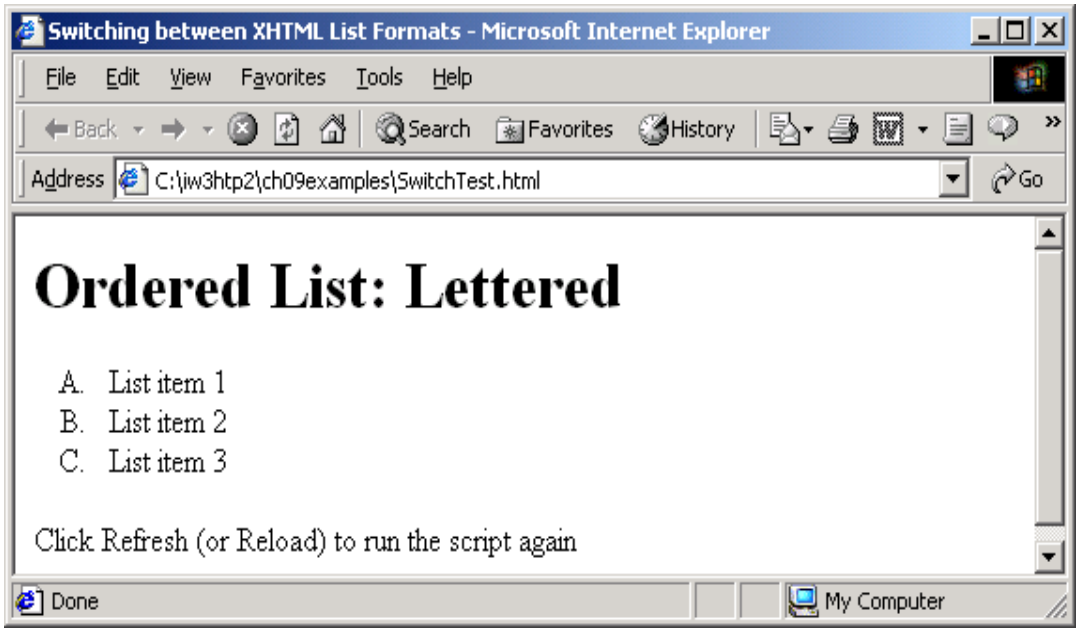

Explorer User Prompt
Script Prompt:
Select a list style:
1 (bullet), 2 (numbered), 3 (lettered)
OK
Cancel
1


Switching between XHTML List Formats - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address C:\iw3htp2\ch09examples\SwitchTest.html
Go
Bullet List
List item 1
List item 2
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
2

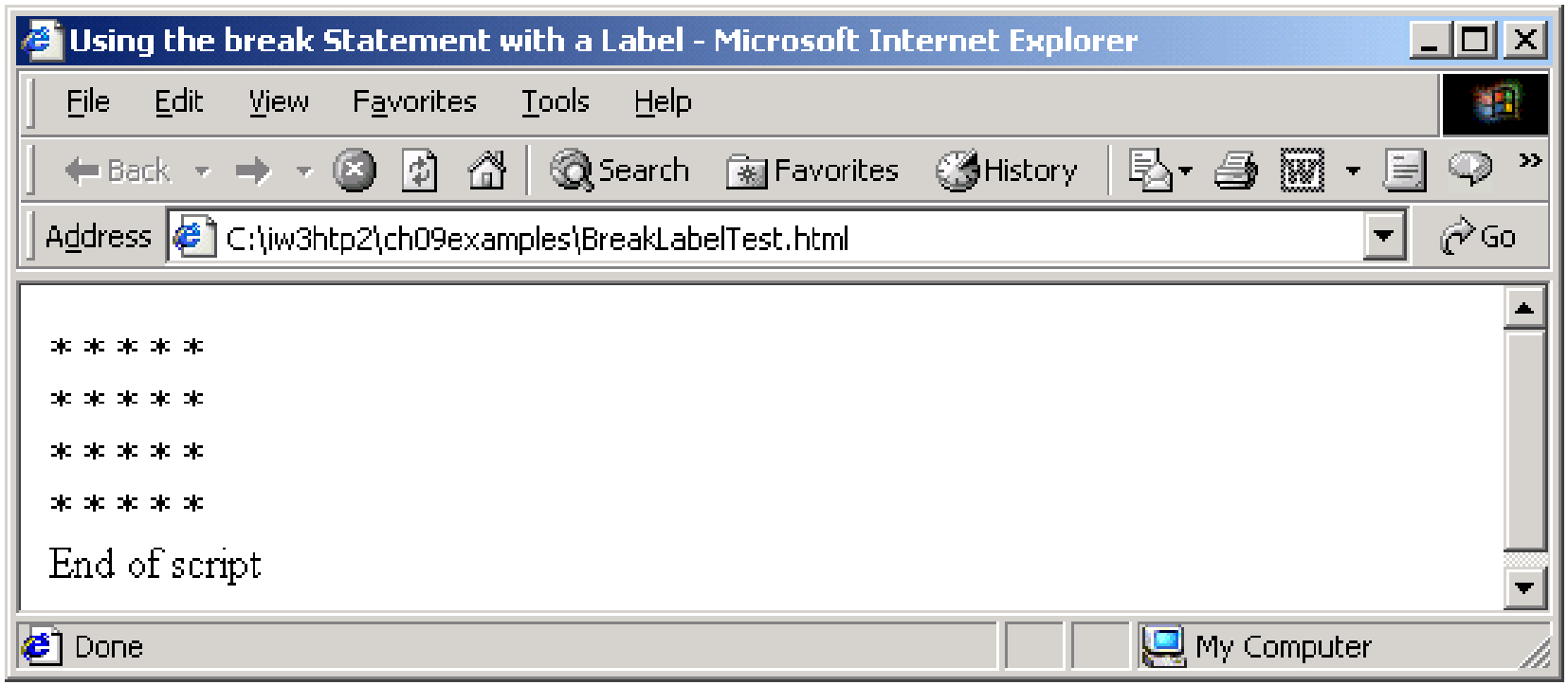

Switching between XHTML List Formats - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address C:\iw3htp2\ch09examples\SwitchTest.html
Go
Ordered List: Numbered
1. List item 1
2. List item 2
3. List item 3
Click Refresh (or Reload) to run the script again
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 History
Address C:\iw3htp2\ch09examples\SwitchTest.html
Go
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

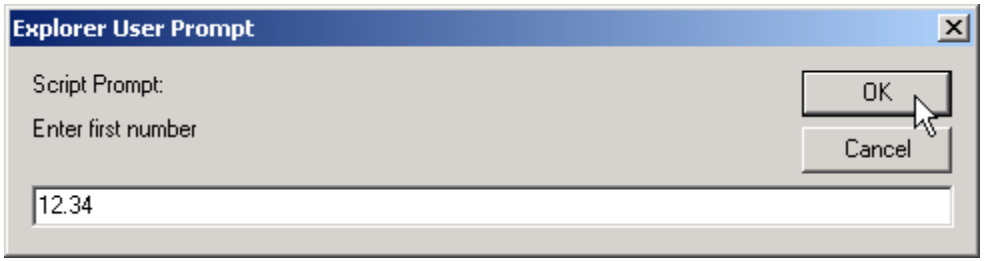
Break statement

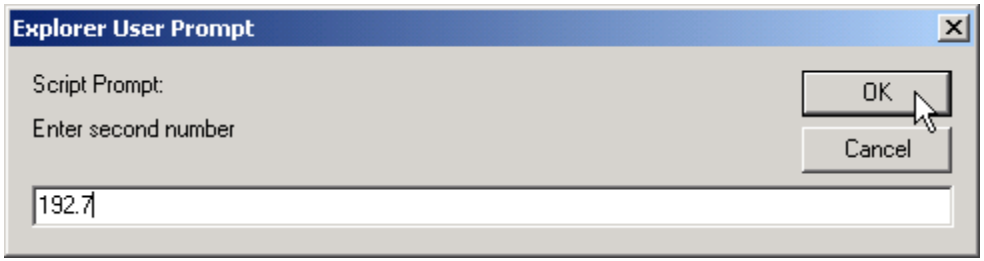
```
<script type = "text/javascript">
stop: {  // labeled compound statement
   for ( var row = 1; row <= 10; ++row ) {
      for ( var column = 1; column <= 5 ; ++column ) {

         if ( row == 5 )
            break stop; // jump to end of stop block

         document.write( "* " );
      }

      document.writeln( "<br />" );
   }
   // the following line is skipped
   document.writeln( "This line should not print" );
}
   document.writeln( "End of script" );
</script>
```

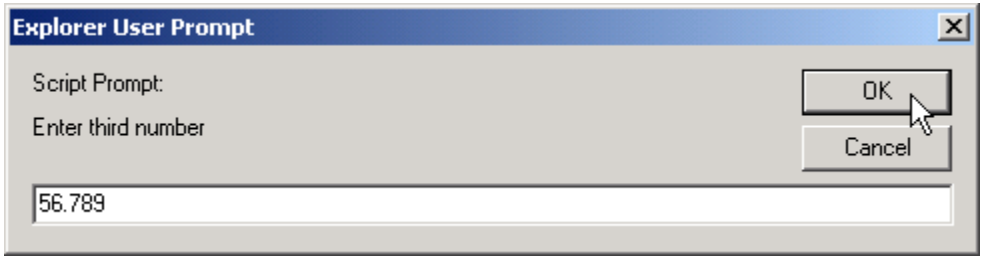

Using the break Statement with a Label - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address C:\iw3htp2\ch09examples\BreakLabelTest.html
Go
* * * * *
* * * * *
* * * * *
* * * * *
End of script
Done
My Computer

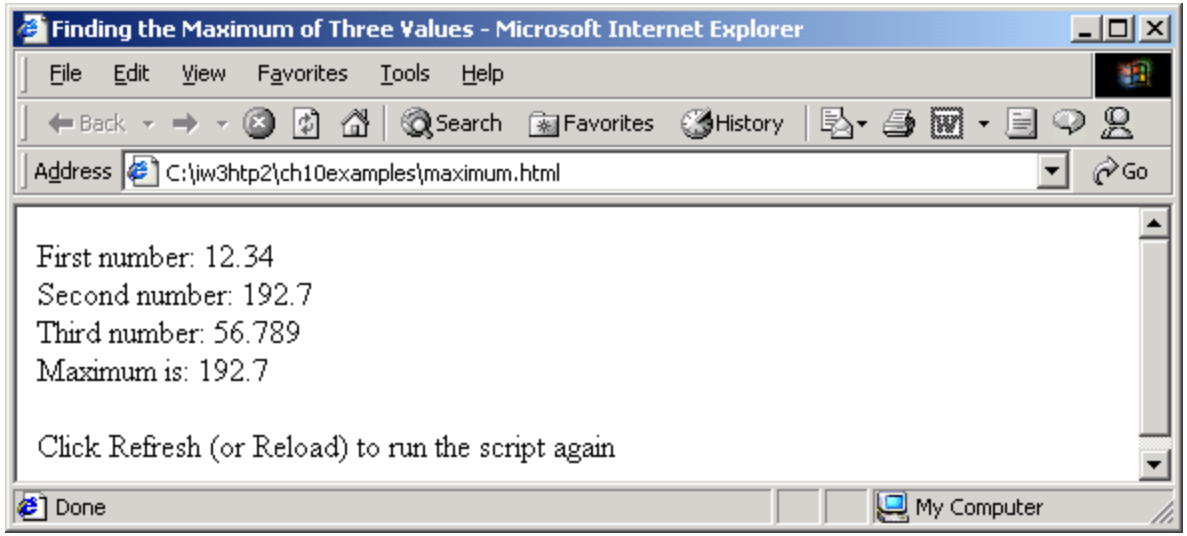
Example

```
<script type = "text/javascript">
    var input1 = window.prompt( "Enter first number", "0" );
    var input2 = window.prompt( "Enter second number", "0" );
    var input3 = window.prompt( "Enter third number", "0" );

    var value1 = parseFloat( input1 );
    var value2 = parseFloat( input2 );
    var value3 = parseFloat( input3 );

  var maxValue = maximum( value1, value2, value3 );

 document.writeln( "First number: " + value1 + "<br />Second number: " + value2
    + "<br />Third number: " + value3 + "<br />Maximum is: " + maxValue );

function maximum( x, y, z ) {
        return Math.max( x, Math.max( y, z ) );
} </script>
```


Explorer User Prompt
Script Prompt:
Enter first number
OK
Cancel
12.34

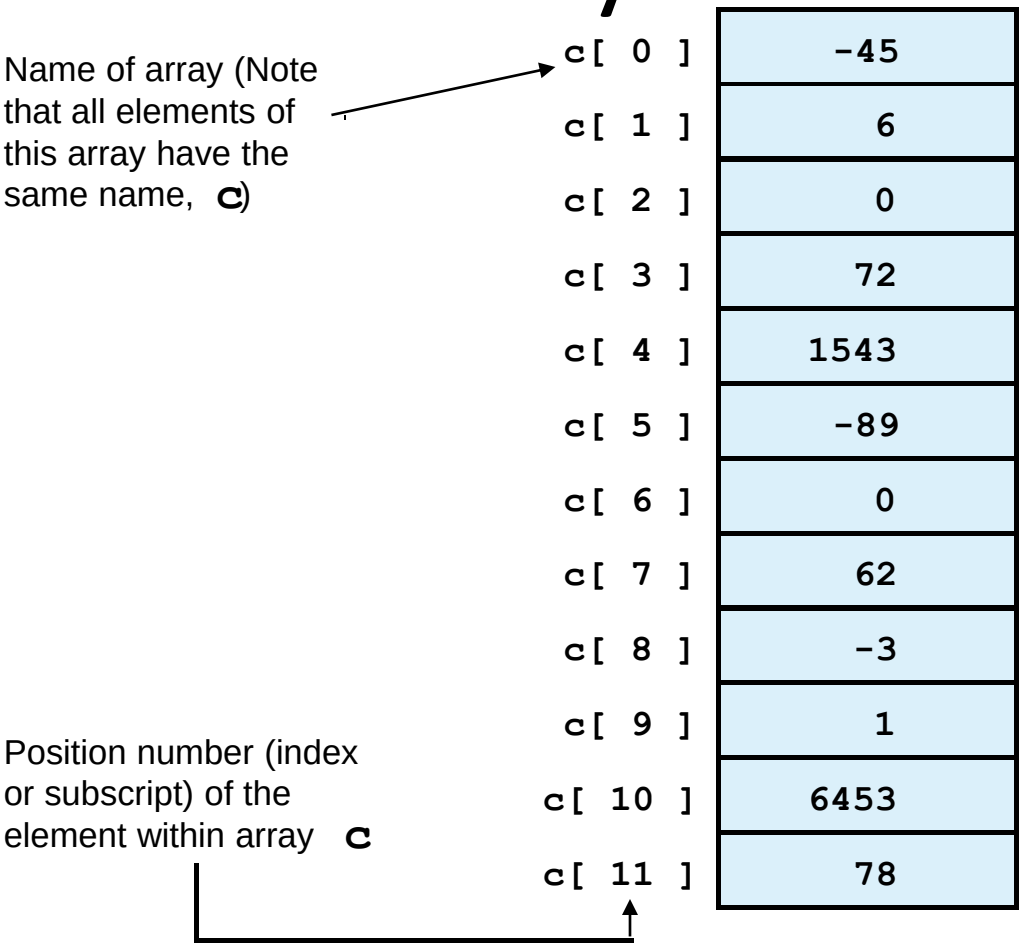

Explorer User Prompt
Script Prompt:
Enter second number
OK
Cancel
192.7


Explorer User Prompt
Script Prompt:
Enter third number
OK
Cancel
56.789


Finding the Maximum of Three Values - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address C:\iw3htp2\ch10examples\maximum.html
Go
First number: 12.34
Second number: 192.7
Third number: 56.789
Maximum is: 192.7
Click Refresh (or Reload) to run the script again
Done
My Computer

JavaScript: Arrays

Arrays



A 12-element array.

JavaScript Arrays

- arrays store a sequence of items, accessible via an index
  - since JavaScript is loosely typed, elements do not have to be the same type
  - to create an array, allocate space using new (or can assign directly)

```
items = new Array(10);          // allocates space for 10 items

items = new Array();            // if no size, will adjust dynamically

items = [0,0,0,0,0,0,0,0,0,0]; // can assign size & values []
```

JavaScript Arrays

- to access an array element, use [] (as in C++/Java)

```
for (i = 0; i < 10; i++) {
   items[i] = 0;              // stores 0 at each index

}
```

- the length property stores the number of items in the array

```
for (i = 0; i < items.length; i++) {
   document.write(items[i]  +  "<br>");        //  displays
   elements
}
```

JavaScript Arrays

```
<script type="text/javascript">
   numRolls = 60000;
   dieSides = 6;
   rolls = new Array(dieSides+1);
   for (i = 1; i < rolls.length; i++) {
      rolls[i] = 0;
   }
   for(i = 1; i <= numRolls; i++) {
      rolls[RandomInt(1, dieSides)]++;
   }
   for (i = 1; i < rolls.length; i++) {
      document.write("Number of " + i + "'s = " +
                     rolls[i] + "<br />");
   }
 </script>
```

RandomNum(low, high)
- returns random real in range [low..high)

RandomInt(low, high)
- returns random integer in range [low..high)

RandomChar(string)
- returns random character from the string

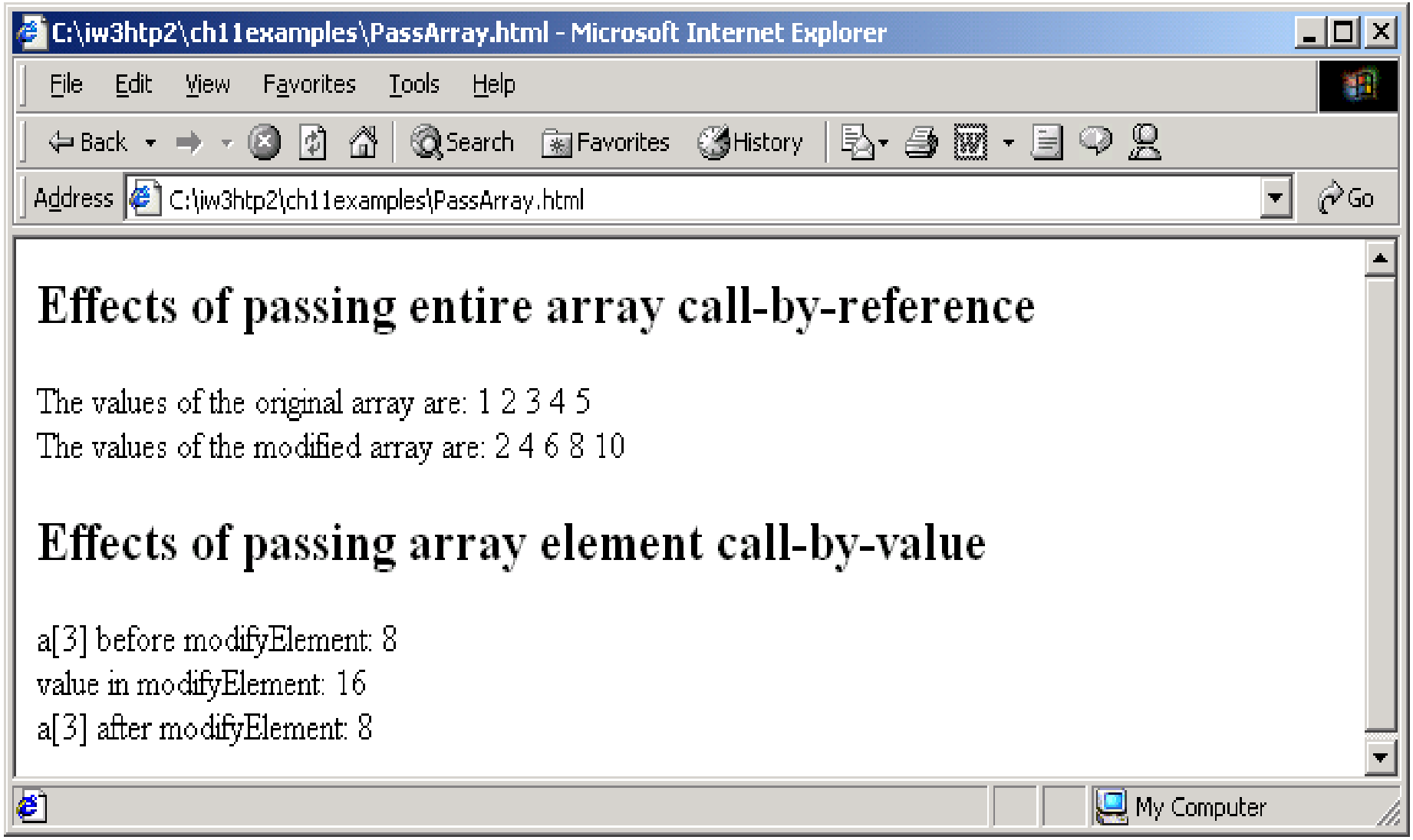
RandomOneOf([item1,…,itemN])
- returns random item from list/array

Array Example

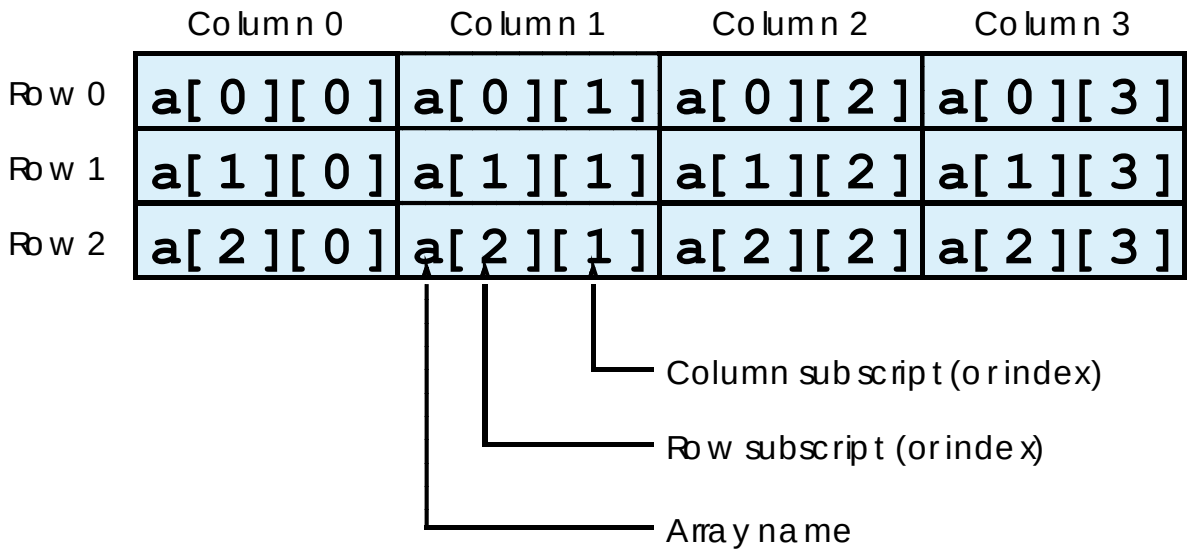
```
<body onload = "start()">
  <script type = "text/javascript">
        function start() {
          var a = [ 1, 2, 3, 4, 5 ];

document.writeln( "<h2>Effects of passing entire " +  "array call-by-
    reference</h2>" );
 outputArray( "The values of the original array are: ", a );

          modifyArray( a );  // array a passed call-by-reference

          outputArray( "The values of the modified array are: ", a );

document.writeln( "<h2>Effects of passing array " + "element call-by-value</h2>"
    + "a[3] before modifyElement: " + a[ 3 ] );

          modifyElement( a[ 3 ] );
```

```
   document.writeln(
      "<br />a[3] after modifyElement: " + a[ 3 ] );
}

// outputs "header" followed by the contents of "theArray“
function outputArray( header, theArray )
{
   document.writeln(
      header + theArray.join( " " ) + "<br />" );
}

// function that modifies the elements of an array
function modifyArray( theArray )
{
   for ( var j in theArray )
      theArray[ j ] *= 2;
}
```

```
      // function that attempts to modify the value
   passed
      function modifyElement( e )
      {
        e *= 2;
        document.writeln( "<br />value in
   modifyElement: " + e );
      }
    </script>

</body>
```


C:\iw3htp2\ch11examples\PassArray.html - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address C:\iw3htp2\ch11examples\PassArray.html
Go
Effects of passing entire array call-by-reference
The values of the original array are: 1 2 3 4 5
The values of the modified array are: 2 4 6 8 10
Effects of passing array element call-by-value
a[3] before modifyElement: 8
value in modifyElement: 16
a[3] after modifyElement: 8
My Computer

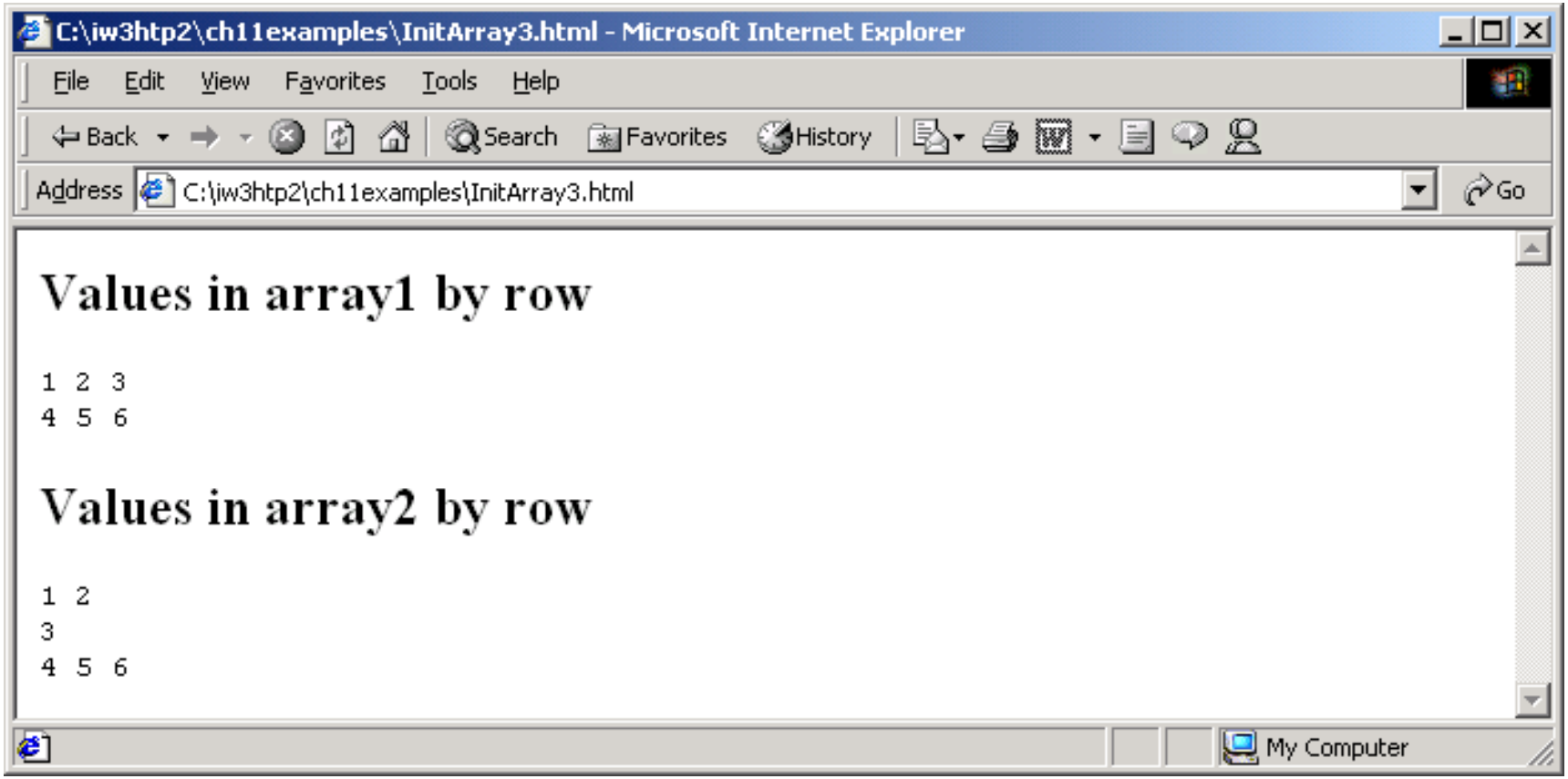
Multiple-Subscripted Arrays



Double-subscripted array with three rows and four columns.

```
<body onload = "start()">
<script type = "text/javascript">
        function start()
        {
          var array1 = [ [ 1, 2, 3 ],     // first row
                       [ 4, 5, 6 ] ];   // second row
          var array2 = [ [ 1, 2 ],       // first row
                       [ 3 ],          // second row
                       [ 4, 5, 6 ] ];   // third row

          outputArray( "Values in array1 by row", array1 );
          outputArray( "Values in array2 by row", array2 );
        }
```

```
function outputArray( header, theArray )
        {
          document.writeln( "<h2>" + header + "</h2><tt>" );

      for ( var i in theArray ) {

            for ( var j in theArray[ i ] )
              document.write( theArray[ i ][ j ] + " " );

            document.writeln( "<br />" );
          }

          document.writeln( "</tt>" );
        }
</script>
</body>
```


C:\iw3htp2\ch11examples\InitArray3.html - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address C:\iw3htp2\ch11examples\InitArray3.html
Go
Values in array1 by row
1 2 3
4 5 6
Values in array2 by row
1 2
3
4 5 6
My Computer

JavaScript Events

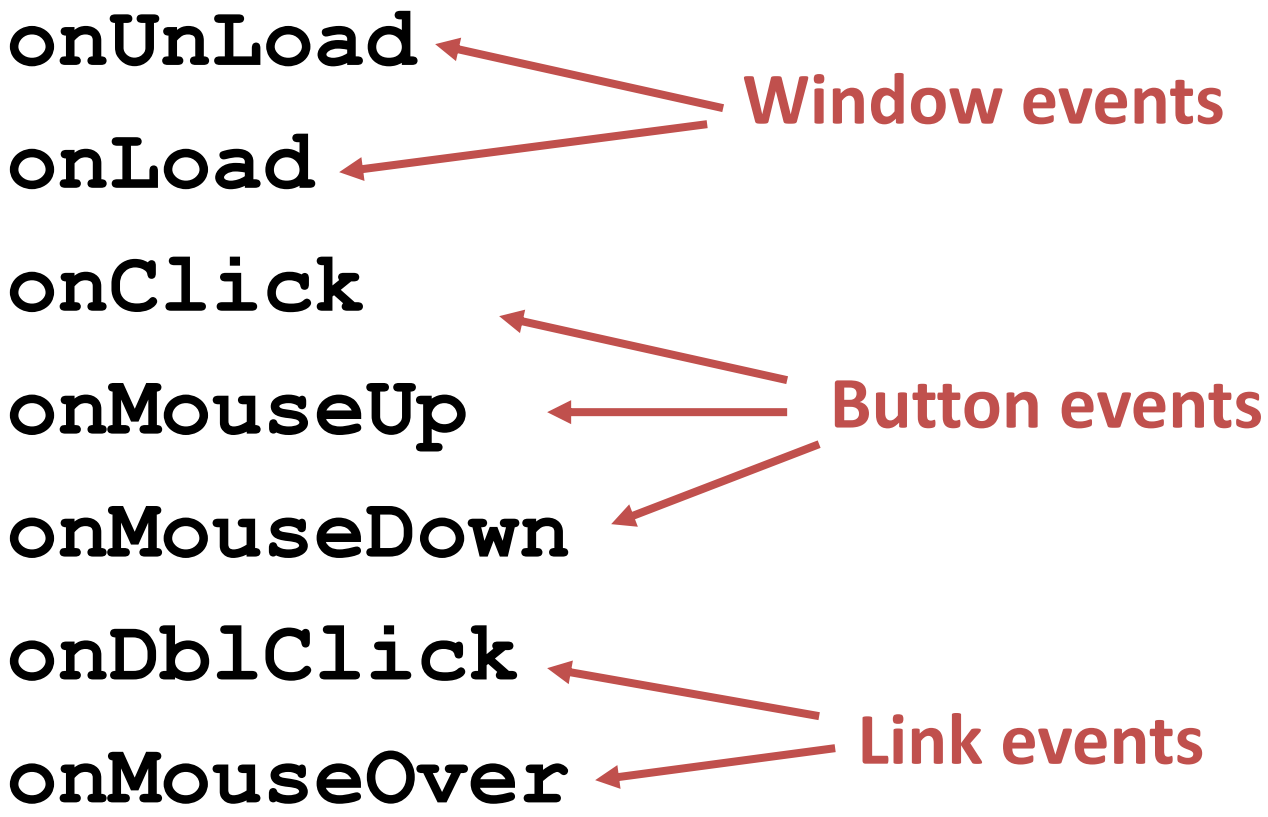
- JavaScript supports an event handling system.
  - You can tell the browser to execute javascript commands when some event occurs.
  - Sometimes the resulting *value of the command* determines the browser action.

Buttons

- You can associate buttons with JavaScript events (buttons in HTML forms)

```
<FORM>
<INPUT TYPE=BUTTON
VALUE="Don't Press Me"
onClick="alert('now you are in trouble!')" >
</FORM>
```

Some Events (a small sample)



JavaScript Date Object

- String & Array are the most commonly used classes in JavaScript
  - other, special purpose classes & objects also exist
- the Date class can be used to access the date and time
  - to create a Date object, use new & supply year/month/day/… as desired
    - today = new Date(); // sets to current date & time
    - newYear = new Date(2002,0,1); //sets to Jan 1, 2002 12:00AM

JavaScript Date Object

- methods include:

  - newYear.getYear() - can access individual components of a date
  - newYear.getMonth()
  - newYear.getDay()
  - newYear.getHours()
  - newYear.getMinutes()
  - newYear.getSeconds()
  - newYear.getMilliseconds()