9

JavaScript: Functions

Form ever follows function.
—Louis Sullivan

E pluribus unum.
(One composed of many.)
—Virgil

O! call back yesterday, bid time return.
—William Shakespeare

Call me Ishmael.
—Herman Melville

When you call me that, smile.
—Owen Wister

OBJECTIVES

In this chapter you will learn:

- To construct programs modularly from small pieces called functions.
- To create new functions.
- How to pass information between functions.
- Simulation techniques that use random number generation.
- How the visibility of identifiers is limited to specific regions of programs.

Self-Review Exercises

9.1 Fill in the blanks in each of the following statements:

a) Program modules in JavaScript are called __________.

ANS: functions.

b) A function is invoked using a(n) __________.

ANS: function call.

c) A variable known only within the function in which it is defined is called a(n) __________.

ANS: local variable.

d) The __________ statement in a called function can be used to pass the value of an expression back to the calling function.

ANS: `return`.

e) The keyword __________ indicates the beginning of a function definition.

ANS: `function`.

9.2 For the given program, state the scope (either global scope or function scope) of each of the following elements:

a) The variable `x`.

ANS: Global scope.

b) The variable `y`.

ANS: Function scope.

c) The function `cube`.

ANS: Global scope.

d) The function `output`.

ANS: Global scope.

```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Exercise 9.2: cube.html -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Scoping</title>
      <script type = "text/javascript">
         <!--
         var x;

         function output()
         {
            for ( x = 1; x <= 10; x++ )
               document.writeln( cube( x ) + "<br />" );
         } // end function output

         function cube( y )
         {
            return y * y * y;
         } // end function cube
         // -->
      </script>
   </head><body onload = "output()"></body>
</html>
```

9.3 Fill in the blanks in each of the following statements:

a) Programmer-defined functions, global variables and JavaScript's global functions are all part of the ________ object.

ANS: Global.

b) Function ________ determines if its argument is or is not a number.

ANS: `isNaN`.

c) Function ________ takes a string argument and returns a string in which all spaces, punctuation, accent characters and any other character that is not in the ASCII character set are encoded in a hexadecimal format.

ANS: `escape`.

d) Function ________ takes a string argument representing JavaScript code to execute.

ANS: `eval`.

e) Function ________ takes a string as its argument and returns a string in which all characters that were previously encoded with `escape` are decoded.

ANS: `unescape`.

9.4 Fill in the blanks in each of the following statements:

a) The ________ of an identifier is the portion of the program in which the identifier can be used.

ANS: scope.

b) The three ways to return control from a called function to a caller are ________, ________ and ________.

ANS: `return;` or `return` *expression*`;` or encountering the closing right brace of a function.

c) The ________ function is used to produce random numbers.

ANS: `Math.random`.

d) Variables declared in a block or in a function's parameter list are of ________ scope.

ANS: local.

9.5 Locate the error in each of the following program segments and explain how to correct it:

a)
```
method g()
{
   document.writeln( "Inside method g" );
}
```

ANS: Error: `method` is not the keyword used to begin a function definition.
Correction: Change `method` to `function`.

b)
```
// This function should return the sum of its arguments
function sum( x, y )
{
   var result;
   result = x + y;
}
```

ANS: Error: The function is supposed to return a value, but does not.
Correction: Delete variable `result`, and either place the statement

```
return x + y;
```

in the function or add the following statement at the end of the function body:

```
return result;
```

c)
```
function f( a );
{
   document.writeln( a );
}
```

ANS: Error: The semicolon after the right parenthesis that encloses the parameter list.
Correction: Delete the semicolon after the right parenthesis of the parameter list.

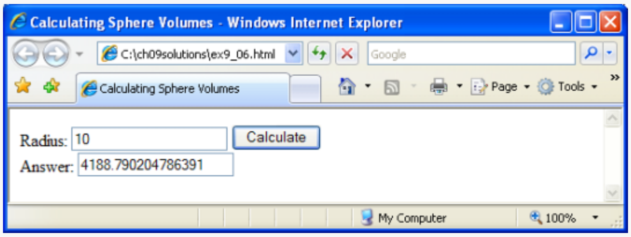
9.6 Write a complete JavaScript program to prompt the user for the radius of a sphere, and call function `sphereVolume` to calculate and display the volume of the sphere. Use the statement

```
volume = ( 4.0 / 3.0 ) * Math.PI * Math.pow( radius, 3 );
```

to calculate the volume. The user should input the radius through an XHTML text field in a `<form>` and click an XHTML button to initiate the calculation.

ANS: The following solution calculates the volume of a sphere using the radius entered by the user.

```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Exercise 9.6: volume.html -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Calculating Sphere Volumes</title>
      <script type = "text/javascript">
         <!--
         function displayVolume()
         {
            var inputField = document.getElementById( "radiusField" );
            var radius = parseFloat( inputField.value );
            var answerField = document.getElementById( "answer" );
            answerField.value = sphereVolume( radius );
         } // end function displayVolume

         function sphereVolume( radius )
         {
            return ( 4.0 / 3.0 ) * Math.PI * Math.pow( radius, 3 );
         } // end function sphereVolume
         // -->
      </script>
   </head>
   <body>
      <form action = "">
         <div>
            <label>Radius:
               <input id = "radiusField" type = "text" /></label>
               <input type = "button" value = "Calculate"
                  onclick = "displayVolume()" />
            <br />
            <label>Answer:
               <input id = "answer" type = "text" /></label>
         </div>
      </form>
   </body>
</html>
```

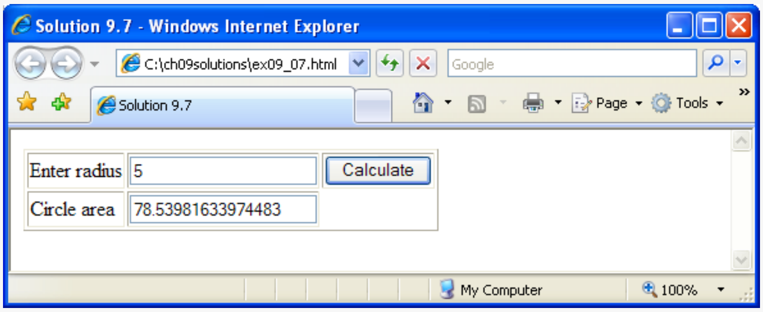

Calculating Sphere Volumes - Windows Internet Explorer
C:\ch09solutions\ex9_06.html
Google
Calculating Sphere Volumes
Page
Tools
My Computer
100%

Radius: 10 [Calculate]
Answer: 4188.790204786391

Exercises

9.7 Write a script that prompts the user for the radius of a circle, uses a function `circleArea` to calculate the area of the circle, and prints the area of the circle.

```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Exercise 9.7: Solution -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Solution 9.7</title>
      <script type = "text/javascript">
         <!--
         function getArea()
         {
            var form = document.getElementById( "myForm" );
            var input = parseFloat( form.number.value );
            form.result.value = circleArea( input );
         } // end function getArea

         function circleArea( radius )
         {
            return Math.PI * radius * radius;
         } // end function circleArea
         // -->
      </script>
   </head>
   <body>
      <form id = "myForm" action = "">
         <table border = "1">
            <tr><td>Enter radius</td>
                <td><input id = "number" type = "text" />
                </td>
                <td><input type = "button" value = "Calculate"
                       onclick = "getArea()" /></td>
            </tr>
            <tr><td>Circle area</td>
                <td><input id = "result" type = "text" />
                </td>
            </tr>
         </table>
      </form>
   </body>
</html>
```


Solution 9.7 - Windows Internet Explorer
C:\ch09solutions\ex09_07.html
Google
Solution 9.7
Page
Tools
Calculate
My Computer
100%

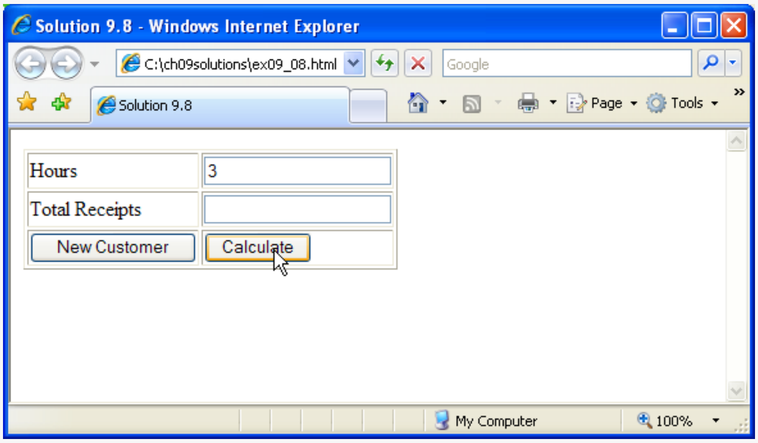
Enter radius 5

Circle area 78.53981633974483

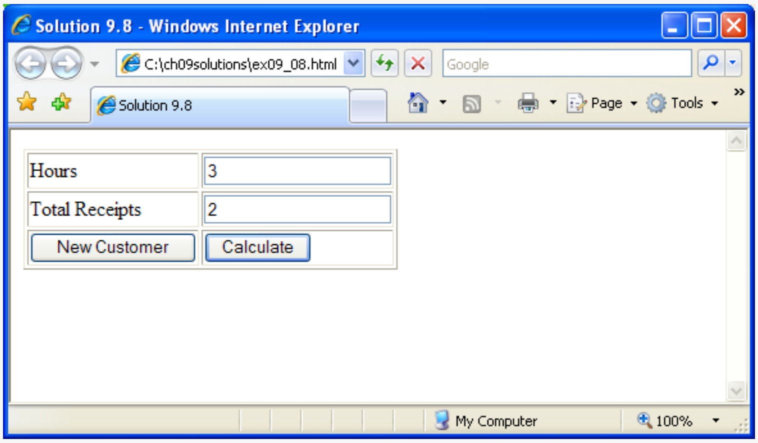
9.8 A parking garage charges a $2.00 minimum fee to park for up to three hours. The garage charges an additional $0.50 per hour for each hour *or part thereof* in excess of three hours. The maximum charge for any given 24-hour period is $10.00. Assume that no car parks for longer than 24 hours at a time. Write a script that calculates and displays the parking charges for each customer who parked a car in this garage yesterday. You should input from the user the hours parked for each customer. The program should display the charge for the current customer and should calculate and display the running total of yesterday's receipts. The program should use the function `calculateCharges` to determine the charge for each customer. Use a text input field to obtain the input from the user.

ANS:

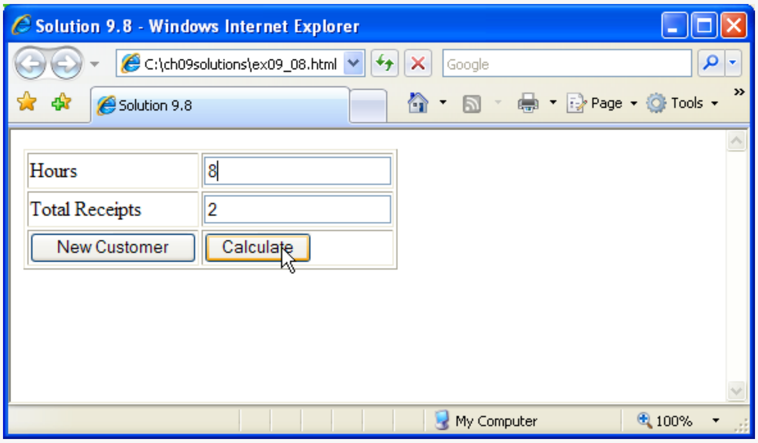
```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Exercise 9.8: Solution -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Solution 9.8</title>
      <script type = "text/javascript">
         <!--
         var totalReceipts = 0;
         var fee;

         function init()
         {
            var hoursField = document.getElementById( "hours" );
            var totalField = document.getElementById( "total" );
            hoursField.value = 0;
            totalField.value = "";
         } // end function init

         function getCharges()
         {
            var hoursField = document.getElementById( "hours" );
            var totalField = document.getElementById( "total" );
            var hours = parseFloat( hoursField.value );
            fee = calculateCharges( hours );
            totalReceipts += fee;
            totalField.value = parseFloat( totalReceipts );
         } // end function getCharges

         function calculateCharges( hours )
         {
            var charge = 0.0;

            if ( hours <= 3.0 && hours > 0.0 )
               charge = 2.0;
            else if ( hours > 3.0 && hours <= 19.0 )
               charge = 2.0 + 0.5 * Math.ceil( hours - 3.0 );
            else if ( hours > 19.0 )
               charge = 10.0;

            return charge;
         } // end function calculateCharges
         // -->
      </script>
   </head>
   <body>
      <form action = "">
         <table border = "1">
            <tr><td>Hours</td>
                <td><input id = "hours" type = "text" /></td>
            </tr>
            <tr><td>Total Receipts</td>
                <td><input id = "total" type = "text" /></td>
            </tr>
            <tr><td>
                <input type = "button" value = "New Customer"
                   onclick = "init()" /></td><td>
                <input type = "button" value = "Calculate"
                   onclick = "getCharges()" />
                </td>
            </tr>
         </table>
      </form>
   </body>
</html>
```


Solution 9.8 - Windows Internet Explorer
C:\ch09solutions\ex09_08.html
Google
Solution 9.8
Page
Tools
My Computer
100%

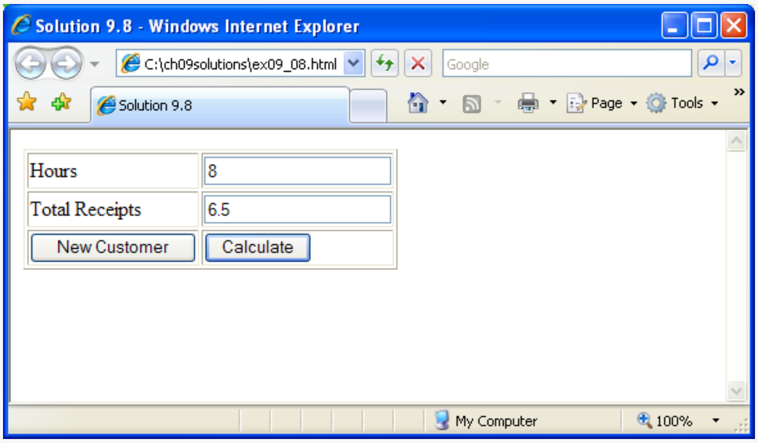
Hours	3
Total Receipts	
New Customer	Calculate


Solution 9.8 - Windows Internet Explorer
C:\ch09solutions\ex09_08.html
Google
Solution 9.8
Page
Tools
My Computer
100%

Hours	3
Total Receipts	2
New Customer	Calculate


Solution 9.8 - Windows Internet Explorer
C:\ch09solutions\ex09_08.html
Google
Solution 9.8
Page
Tools
My Computer
100%

Hours	8
Total Receipts	2
New Customer	Calculate


Solution 9.8 - Windows Internet Explorer
C:\ch09solutions\ex09_08.html
Google
Solution 9.8
Page
Tools
My Computer
100%

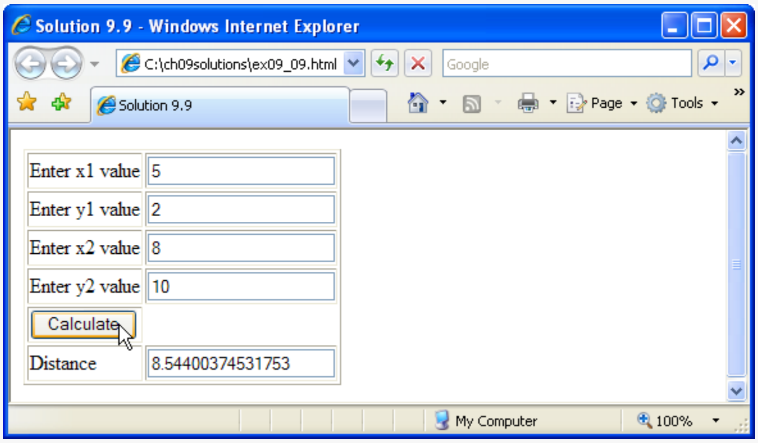
Hours	8
Total Receipts	6.5
New Customer	Calculate

9.9 Write function `distance` that calculates the distance between two points (*x1*, *y1*) and (*x2*, *y2*). All numbers and return values should be floating-point values. You'll need the `Math.sqrt` method for calculating square roots. Incorporate this function into a script that enables the user to enter the coordinates of the points through an XHTML form.

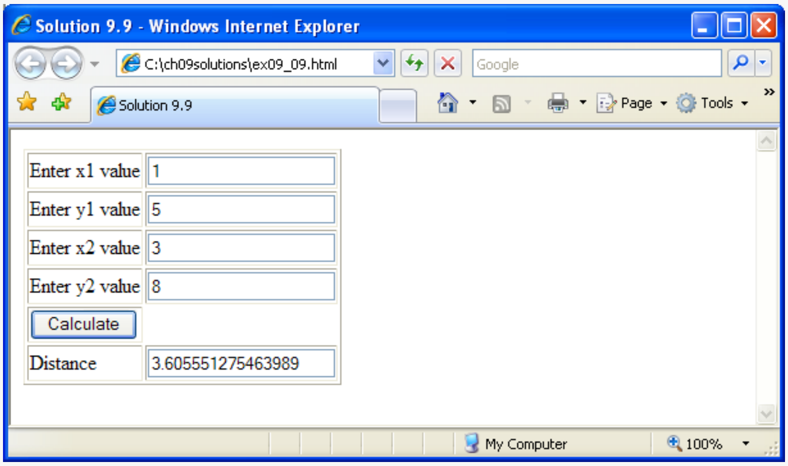
ANS:

```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Exercise 9.9: Solution -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Solution 9.9</title>
      <script type = "text/javascript">
         <!--
         function getDistance()
         {
            var form = document.getElementById( "myForm" );
            var x1 = parseInt( form.x1.value );
            var y1 = parseInt( form.y1.value );
            var x2 = parseInt( form.x2.value );
            var y2 = parseInt( form.y2.value );
            form.result.value = distance( x1, y1, x2, y2 );
         } // end function getDistance

         function distance( x1, y1, x2, y2 )
         {
            return Math.sqrt( Math.pow( ( x1 - x2 ) , 2 )
               + Math.pow( ( y1 - y2 ) , 2 ) );
         } // end function distance
         // -->
      </script>
   </head>
   <body>
      <form id = "myForm" action = "">
```

```
            <table border = "1">
               <tr><td>Enter x1 value</td>
                   <td><input name = "x1" type = "text" />
                   </td>
               </tr>
               <tr><td>Enter y1 value</td>
                   <td><input name = "y1" type = "text" />
                   </td>
               </tr>
               <tr><td>Enter x2 value</td>
                   <td><input name = "x2" type = "text" />
                   </td>
               </tr>
               <tr><td>Enter y2 value</td>
                   <td><input name = "y2" type = "text" />
                   </td>
               </tr>
               <tr><td colspan = "2" style = "text-align: right">
                  <input type = "button" value = "Calculate"
                     onclick = "getDistance()" /></td></tr>
               <tr><td>Distance</td>
                   <td><input name = "result" type = "text" /></td>
               </tr>
            </table>
         </form>
      </body>
   </html>
```


Solution 9.9 - Windows Internet Explorer
C:\ch09solutions\ex09_09.html
Google
Solution 9.9
Page
Tools
My Computer
100%

Enter x1 value	5
Enter y1 value	2
Enter x2 value	8
Enter y2 value	10
Calculate	
Distance	8.54400374531753


Solution 9.9 - Windows Internet Explorer
C:\ch09solutions\ex09_09.html
Google
Solution 9.9
Page
Tools
My Computer
100%

Enter x1 value	1
Enter y1 value	5
Enter x2 value	3
Enter y2 value	8
Calculate	
Distance	3.605551275463989

9.10 Answer each of the following questions:

a) What does it mean to choose numbers "at random"?

ANS: Every number has an equal chance of being chosen at any time.

b) Why is the `Math.random` function useful for simulating games of chance?

ANS: `Math.random` produces a series of random numbers between 0.1 and 1.0, and randomness is useful in making simulations appear realistic.

c) Why is it often necessary to scale and/or shift the values produced by `Math.random`?

ANS: To produce random numbers in a range other than 0.1 to 1.0.

d) Why is computerized simulation of real-world situations a useful technique?

ANS: It enables more accurate predictions of random events such as cars arriving at toll booths and people arriving in lines at a supermarket. The results of a simulation can help determine how many toll booths to have open or how many cashiers to have open at a specified time.

9.11 Write statements that assign random integers to the variable *n* in the following ranges:

a) $1 \le n \le 2$

ANS: `n = Math.floor( 1 + Math.random() * 2 );`

b) $1 \le n \le 100$

ANS: `n = Math.floor( 1 + Math.random() * 100 );`

c) $0 \le n \le 9$

ANS: `n = Math.floor( Math.random() * 10 );`

d) $1000 \le n \le 1112$

ANS: `n = Math.floor( 1000 + Math.random() * 113 );`

e) $-1 \le n \le 1$

ANS: `n = Math.floor( -1 + Math.random() * 3 );`

f) $-3 \le n \le 11$

ANS: `n = Math.floor( -3 + Math.random() * 15 );`

9.12 For each of the following sets of integers, write a single statement that will print a number at random from the set:

a) 2, 4, 6, 8, 10.

ANS: `document.write( ( parseInt( Math.random() * 5) + 1) * 2);`

b) 3, 5, 7, 9, 11.

ANS: `document.write( ( parseInt( Math.random() * 5) + 1) * 2 + 1);`

c) 6, 10, 14, 18, 22.

ANS: `document.write( parseInt( Math.random() * 5) * 4 + 6);`

9.13 Write a function `integerPower( base, exponent )` that returns the value of

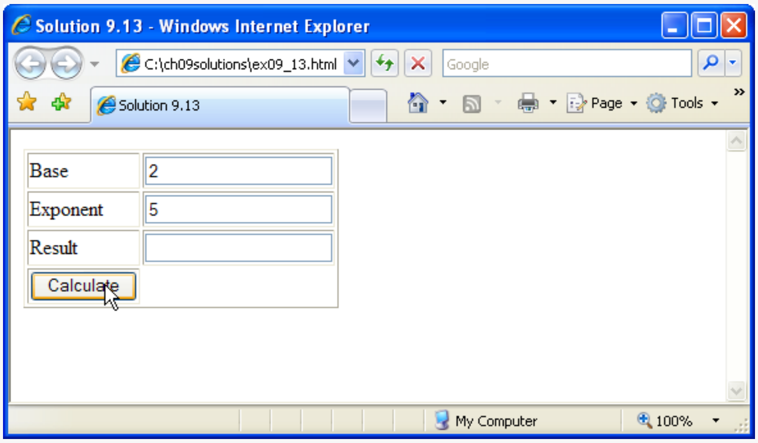
$base^{exponent}$

For example, `integerPower( 3, 4 ) = 3 * 3 * 3 * 3`. Assume that `exponent` and `base` are integers. Function `integerPower` should use a `for` or `while` statement to control the calculation. Do not use any math library functions. Incorporate this function into a script that reads integer values from an XHTML form for `base` and `exponent` and performs the calculation with the `integerPower` function. The XHTML form should consist of two text fields and a button to initiate the calculation. The user should interact with the program by typing numbers in both text fields then clicking the button.

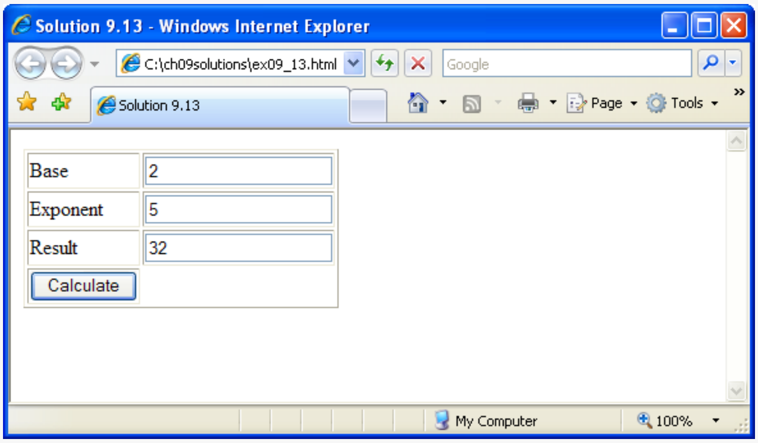
ANS:

```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Exercise 9.13: Solution -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Solution 9.13</title>
      <script type = "text/javascript">
         <!--
         function getExpo()
         {
            var form = document.getElementById( "myForm" );
            var base;
            var expo;
            base = parseInt( form.base.value );
            expo = parseInt( form.expo.value );
            form.result.value = integerPower( base, expo );
         } // end function getExpo

         function integerPower( base, expo )
         {
            var result = 1;

            for( var i = 0; i < expo; ++i )
               result *= base;

            return result;
         } // end function integerPower
         // -->
      </script>
   </head>
   <body>
      <form id = "myForm" action = "">
         <table border = "1">
            <tr><td>Base</td>
                <td><input name = "base" type = "text" /></td>
            </tr>
            <tr><td>Exponent</td>
                <td><input name = "expo" type = "text" /></td>
            </tr>
```

```
                <tr><td>Result</td>
                    <td><input name = "result" type = "text" />
                    </td>
                </tr>
                <tr><td>
                    <input type = "button" value = "Calculate"
                       onclick = "getExpo()" />
                </td></tr>
            </table>
         </form>
      </body>
   </html>
```


Solution 9.13 - Windows Internet Explorer
C:\ch09solutions\ex09_13.html
Google
Solution 9.13
Page
Tools
My Computer
100%

Base	2
Exponent	5
Result	
Calculate	


Solution 9.13 - Windows Internet Explorer
C:\ch09solutions\ex09_13.html
Google
Solution 9.13
Page
Tools
My Computer
100%

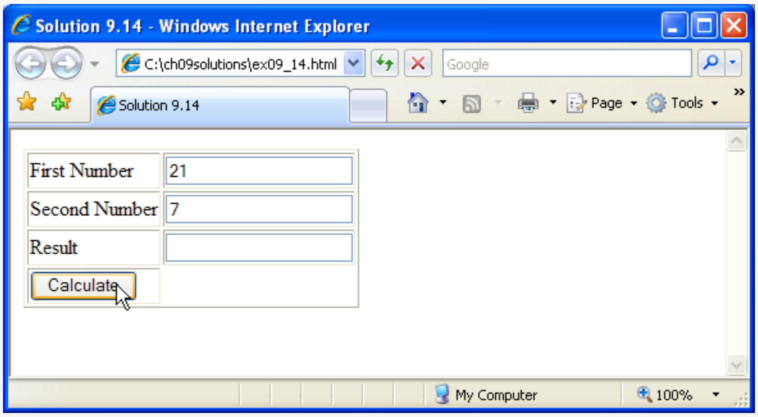
Base	2
Exponent	5
Result	32
Calculate	

9.14 Write a function `multiple` that determines, for a pair of integers, whether the first integer is a multiple of the second. The function should take two integer arguments and return `true` if the first is a multiple of the second, and `false` otherwise. Incorporate this function into a script that inputs a series of pairs of integers (one pair at a time). The XHTML form should consist of two text fields and a button to initiate the calculation. The user should interact with the program by typing numbers in both text fields, then clicking the button.

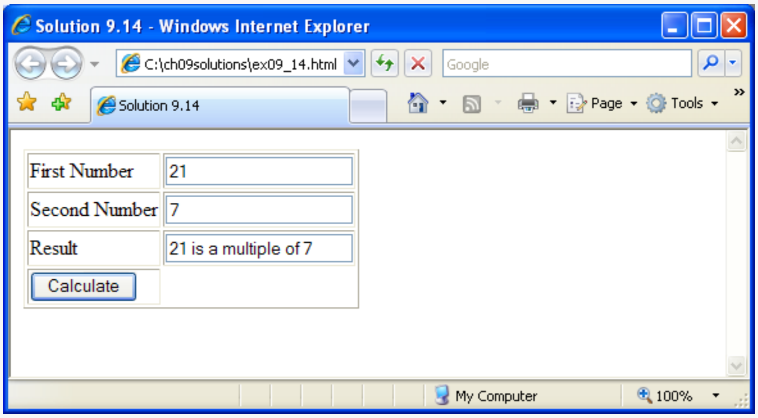
ANS:

```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Exercise 9.14: Solution -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Solution 9.14</title>
      <script type = "text/javascript">
         <!--
         function testMult()
         {
            var num1;
            var num2;
            var result;
            var form = document.getElementById( "myForm" );
            num1 = parseInt( form.num1.value );
            num2 = parseInt( form.num2.value );

            if ( multiple( num1, num2 ) )
               form.result.value =
                  num1 + " is a multiple of " + num2;
            else
               form.result.value =
                  num1 + " is not a multiple of " + num2;
         } // end function testMult

         function multiple( num1, num2 )
         {
            return ( ( num1 % num2 ) == 0 )
         } // end function multiple
         // -->
      </script>
   </head>
   <body>
      <form id = "myForm" action = "">
         <table border = "1">
            <tr><td>First Number</td>
                <td><input name = "num1" type = "text" /></td>
            </tr>
            <tr><td>Second Number</td>
                <td><input name = "num2" type = "text" /></td>
            </tr>
            <tr><td>Result</td>
                <td><input name = "result" type = "text" />
                </td>
            </tr>
            <tr><td>
                <input type = "button" value = "Calculate"
                   onclick = "testMult()" /></td>
            </tr>
         </table>
```

```
        </form>
     </body>
  </html>
```


Solution 9.14 - Windows Internet Explorer
C:\ch09solutions\ex09_14.html
Google
Solution 9.14
Page
Tools
My Computer
100%

First Number 21
Second Number 7
Result
Calculate

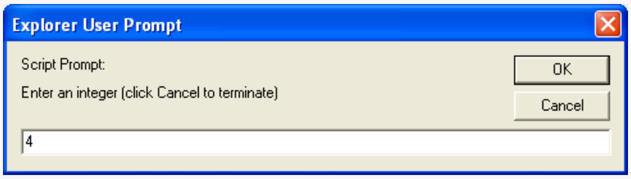

Solution 9.14 - Windows Internet Explorer
C:\ch09solutions\ex09_14.html
Google
Solution 9.14
Page
Tools
My Computer
100%

First Number 21
Second Number 7
Result 21 is a multiple of 7
Calculate

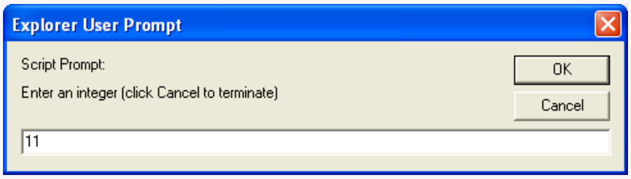
9.15 Write a script that inputs integers (one at a time) and passes them one at a time to function `isEven`, which uses the modulus operator to determine whether an integer is even. The function should take an integer argument and return `true` if the integer is even and `false` otherwise. Use sentinel-controlled looping and a `prompt` dialog.

ANS:

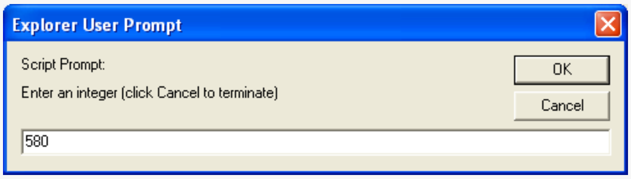
```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Exercise 9.15: Solution -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Solution 9.15</title>
      <script type = "text/javascript">
         <!--
         var input;
         var value;

         input = window.prompt(
            "Enter an integer (click Cancel to terminate)", "0");

         while ( input )
         {
            value = parseInt( input );
            document.write( value + ( isEven( value ) ?
               " is even" : " is odd" ) );
            document.writeln( "<br />" );

            input = window.prompt(
               "Enter an integer (click Cancel to terminate)",
               "0" );
         } // end while

         // this function executes only when called
         function isEven( num )
         {
            return ( num % 2 == 0 );
         } // end function isEven
         // -->
      </script>
   </head>
   <body>
      <p>Click Refresh (or Reload) to run this script again.</p>
   </body>
</html>
```



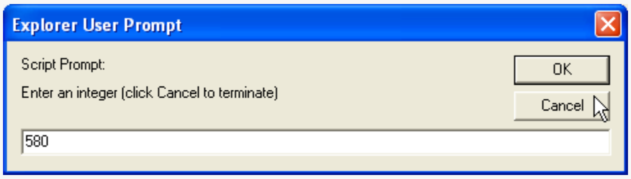
Script Prompt:
Enter an integer (click Cancel to terminate)
4



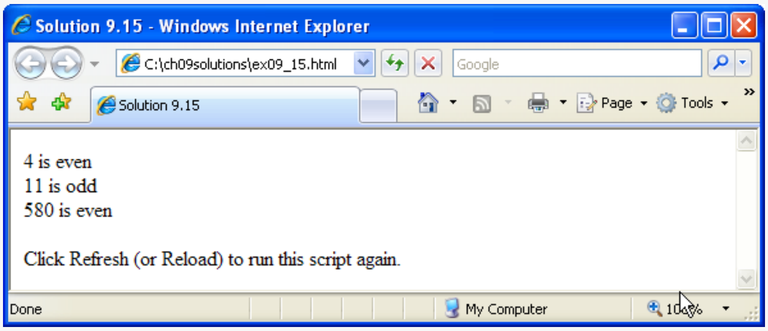
Script Prompt:
Enter an integer (click Cancel to terminate)
11



Script Prompt:
Enter an integer (click Cancel to terminate)
580



Script Prompt:
Enter an integer (click Cancel to terminate)
580


Solution 9.15 - Windows Internet Explorer
C:\ch09solutions\ex09_15.html
Google
Solution 9.15
Page
Tools
Done
My Computer
100%

```
4 is even
11 is odd
580 is even

Click Refresh (or Reload) to run this script again.
```

9.16 Write a function `squareOfAsterisks` that displays a solid square of asterisks whose side is specified in integer parameter `side`. For example, if `side` is `4`, the function displays

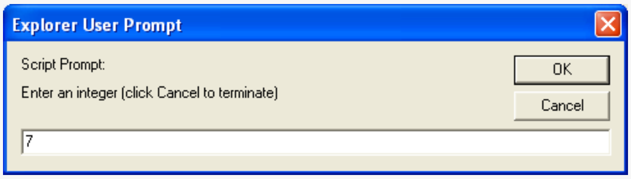
```
****
****
****
****
```

Incorporate this function into a script that reads an integer value for `side` from the user at the keyboard and performs the drawing with the `squareOfAsterisks` function.

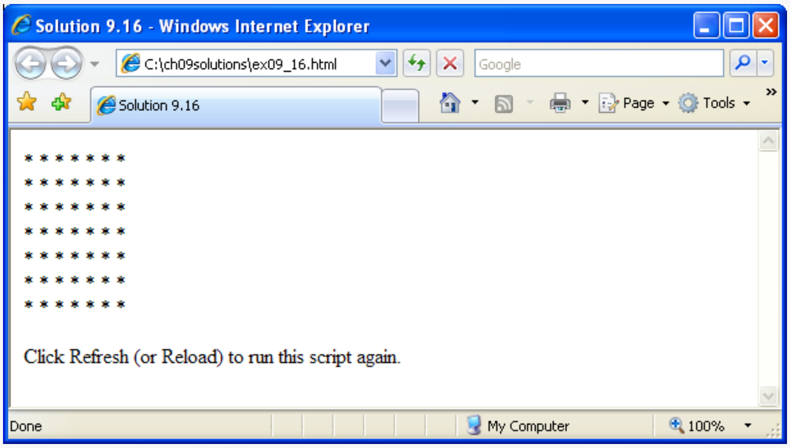
ANS:

```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Exercise 9.16: Solution -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Solution 9.16</title>
      <script type = "text/javascript">
         <!--
         var input;

         input = window.prompt(
            "Enter an integer (click Cancel to terminate)", "0");

         squareOfAsterisks( parseInt( input ) );

         // this function executes only when called
         function squareOfAsterisks( size )
         {
            for ( var a = 1; a <= size * size; ++a )
            {
               document.write( "* " );
```

```
                  if ( a % size == 0 )
                     document.write( "<br />" );
               } // end for
            } // end function squareOfAsterisks
            // -->
         </script>
      </head>
      <body>
         <p>Click Refresh (or Reload) to run this script again.</p>
      </body>
   </html>
```



Script Prompt:
Enter an integer (click Cancel to terminate)
7


Solution 9.16 - Windows Internet Explorer
C:\ch09solutions\ex09_16.html
Google
Solution 9.16
Page
Tools
Done
My Computer
100%

```
*******
*******
*******
*******
*******
*******
*******
```

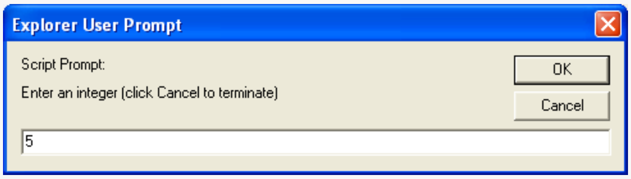
Click Refresh (or Reload) to run this script again.

9.17 Modify the script created in Exercise 9.16 to also promptthe user for a character which will be used to create the square. Thus, if `side` is `5` and `fillCharacter` is `#`, the function should print

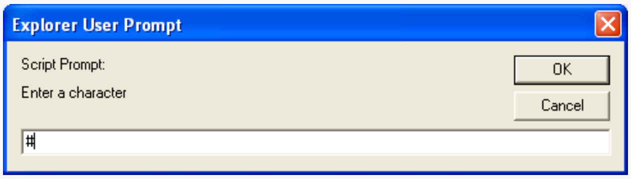
```
#####
#####
#####
#####
#####
```

ANS:

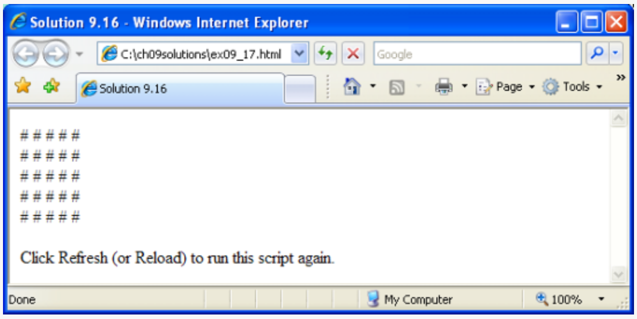
```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Exercise 9.16: Solution -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Solution 9.16</title>
      <script type = "text/javascript">
         <!--
         var sideLength = window.prompt(
            "Enter an integer (click Cancel to terminate)", "0");
         var character = window.prompt( "Enter a character", "*" );

         squareOfCharacters( parseInt( sideLength ), character );

         // this function executes only when called
         function squareOfCharacters( size, fillCharacter )
         {
            for ( var a = 1; a <= size * size; ++a )
            {
               document.write( character + " " );
               if ( a % size == 0 )
                  document.write( "<br />" );
            } // end for
         } // end function squareOfCharacters
         // -->
      </script>
   </head>
   <body>
      <p>Click Refresh (or Reload) to run this script again.</p>
   </body>
</html>
```



Script Prompt:
Enter an integer (click Cancel to terminate)
5



Script Prompt:
Enter a character
#


Solution 9.16 - Windows Internet Explorer
C:\ch09solutions\ex09_17.html
Google
Solution 9.16
Page
Tools
Done
My Computer
100%

```
# # # # #
# # # # #
# # # # #
# # # # #
# # # # #
```

Click Refresh (or Reload) to run this script again.

9.18 Write program segments that accomplish each of the following tasks:

a) Calculate the integer part of the quotient when integer `a` is divided by integer `b`.

b) Calculate the integer remainder when integer `a` is divided by integer `b`.

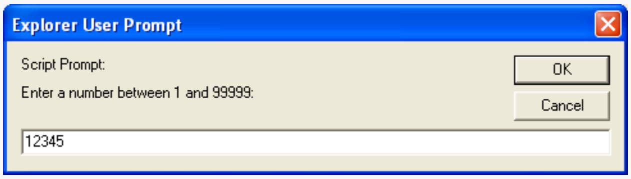
c) Use the program pieces developed in parts (a) and (b) to write a function `displayDigits` that receives an integer between `1` and `99999` and prints it as a series of digits, each pair of which is separated by two spaces. For example, the integer `4562` should be printed as

```
4  5  6  2.
```

d) Incorporate the function developed in part (c) into a script that inputs an integer from a `prompt` dialog and invokes `displayDigits` by passing to the function the integer entered.

ANS:

```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Exercise 9.18: Solution -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Solution 9.18</title>
      <script type = "text/javascript">
         <!--
         var input;

         input = window.prompt(
            "Enter a number between 1 and 99999:", "1" );
         document.writeln( "<pre>" );
         displayDigits( parseInt( input ) );
         document.writeln( "</pre>" );

         // the following functions execute only when called
         // Part A
         function quotient( a, b )
         {
            return Math.floor( a / b );
         } // end function quotient

         // Part B
          function remainder( a, b )
         {
            return a % b;
         } // end function remainder

         // Part C
         function displayDigits( number )
         {
            var divisor = 10000, digit;

            // determine the divisor
            while ( number % divisor == number )
               divisor = quotient( divisor, 10 );

            while ( divisor >= 1 )
            {
               digit = quotient( number, divisor );
               document.write( digit + "   " );
               number = remainder( number, divisor );
               divisor = quotient( divisor, 10 );
            } // end while
         } // end function displayDigits
         // -->
      </script>
   </head>
   <body>
      <p>Click Refresh (or Reload) to run this script again.</p>
   </body>
</html>
```

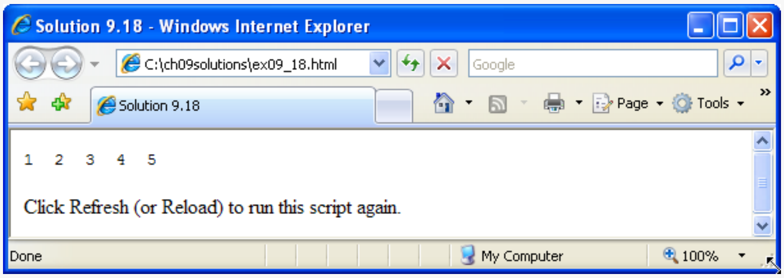


Explorer User Prompt

Script Prompt:

Enter a number between 1 and 99999:

12345



Solution 9.18 - Windows Internet Explorer

C:\ch09solutions\ex09_18.html

```
1  2  3  4  5
```

Click Refresh (or Reload) to run this script again.

9.19 Implement the following functions:

a) Function `celsius` returns the Celsius equivalent of a Fahrenheit temperature, using the calculation

```
C = 5.0 / 9.0 * ( F - 32 );
```

b) Function `fahrenheit` returns the Fahrenheit equivalent of a Celsius temperature, using the calculation

```
F = 9.0 / 5.0 * C + 32;
```

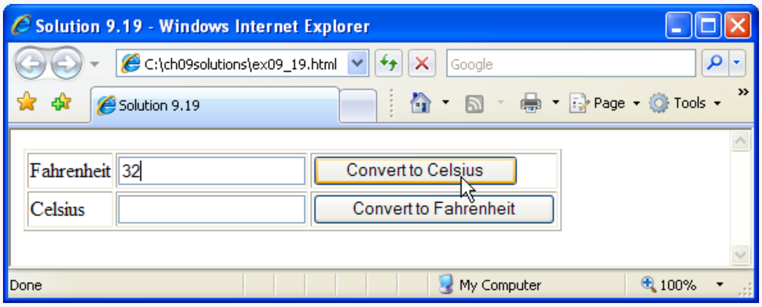
c) Use these functions to write a script that enables the user to enter either a Fahrenheit or a Celsius temperature and displays the Celsius or Fahrenheit equivalent.

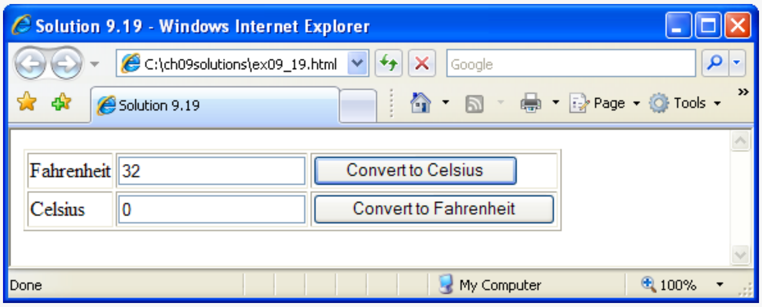
Your XHTML document should contain two buttons—one to initiate the conversion from Fahrenheit to Celsius and one to initiate the conversion from Celsius to Fahrenheit.

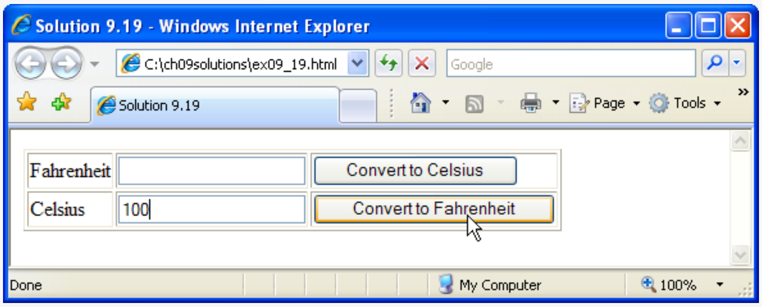
ANS:

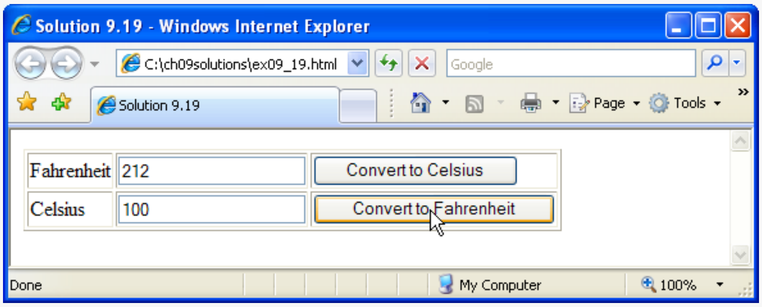
```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Exercise 9.19: Solution -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Solution 9.19</title>
      <script type = "text/javascript">
         <!--
         function convertToCelsius()
         {
            var form = document.getElementById( "myForm" );
            var value = parseInt( form.fText.value );
            form.cText.value = celsius( value );
         } // convertToCelsius

         function celsius( fTemp )
         {
            return ( Math.floor( 5.0 / 9.0 *
               ( fTemp - 32 ) ) );
         } // end function celsius

         function convertToFahrenheit()
         {
            var form = document.getElementById( "myForm" );
            var value = parseInt( form.cText.value );
            form.fText.value = fahrenheit( value );
         } // end function convertToFahrenheit

         function fahrenheit( cTemp )
         {
              return ( Math.floor( 9.0 / 5.0 * cTemp + 32 ) );
         } // end function fahrenheit
         // -->
      </script>
   </head>
   <body>
      <form id = "myForm" action = "">
         <table border = "1">
            <tr><td>Fahrenheit</td>
                <td><input name = "fText" type = "text" /></td>
                <td><input type = "button" value =
                      "Convert to Celsius" onclick =
                      "convertToCelsius()" /></td>
            </tr>
            <tr><td>Celsius</td>
                <td><input name = "cText" type = "text" /></td>
                <td><input type = "button" value =
                      "Convert to Fahrenheit" onclick =
                      "convertToFahrenheit()" /></td>
            </tr>
         </table>
```

```
        </form>
    </body>
</html>
```


Fahrenheit 32 | Convert to Celsius
Celsius | Convert to Fahrenheit


Fahrenheit 32 | Convert to Celsius
Celsius 0 | Convert to Fahrenheit

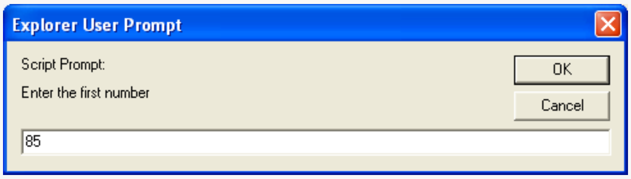

Fahrenheit | Convert to Celsius
Celsius 100 | Convert to Fahrenheit


Fahrenheit 212 | Convert to Celsius
Celsius 100 | Convert to Fahrenheit

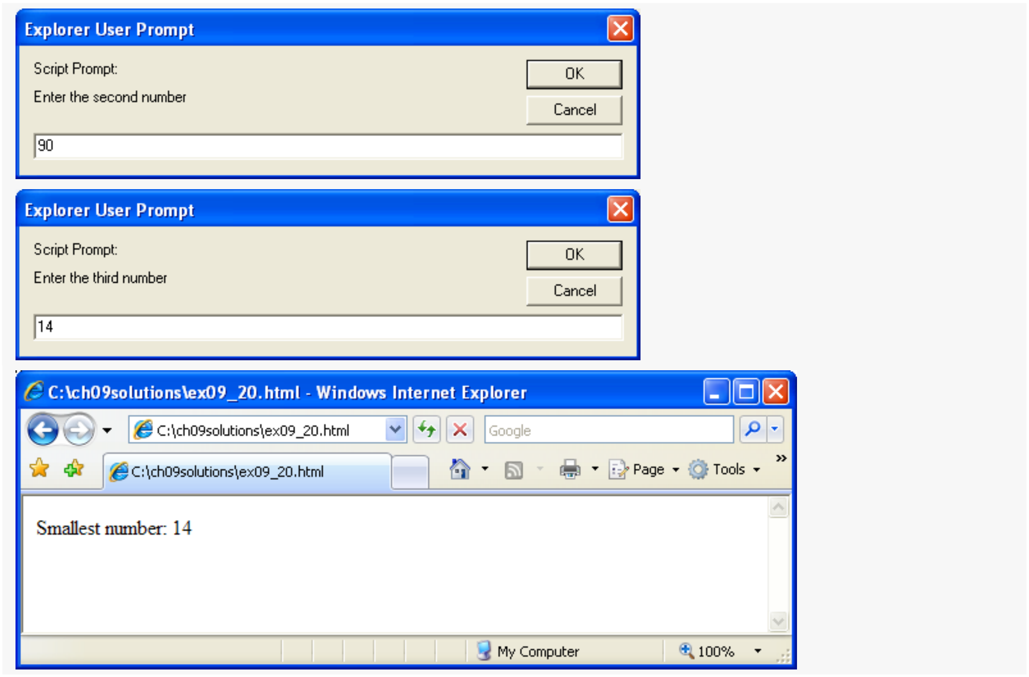
9.20 Write a function `minimum3` that returns the smallest of three floating-point numbers. Use the `Math.min` function to implement `minimum3`. Incorporate the function into a script that reads three values from the user and determines the smallest value.

ANS:

```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Exercise 9.20: Solution -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Solution 9.20</title>
      <script type = "text/javascript">
         <!--
         var number1;
         var number2;
         var number3;
         var min;

         function run()
         {
            number1 = window.prompt( "Enter the first number", "0" );
            number2 = window.prompt( "Enter the second number", "0" );
            number3 = window.prompt( "Enter the third number", "0" );

            min = minimum3( number1, number2, number3 );

            document.write( "Smallest number: " + min );
         } // end function run

         function minimum3( a, b, c )
         {
            var smallest;

            smallest = Math.min( a, b );
            smallest = Math.min( smallest, c );
            return smallest;
         } // end function minimum3
         // -->
      </script>
   </head>
   <body onload = "run()"></body>
</html>
```



Script Prompt:
Enter the first number
85


Explorer User Prompt

Script Prompt:
Enter the second number

90

Explorer User Prompt

Script Prompt:
Enter the third number

14

C:\ch09solutions\ex09_20.html - Windows Internet Explorer

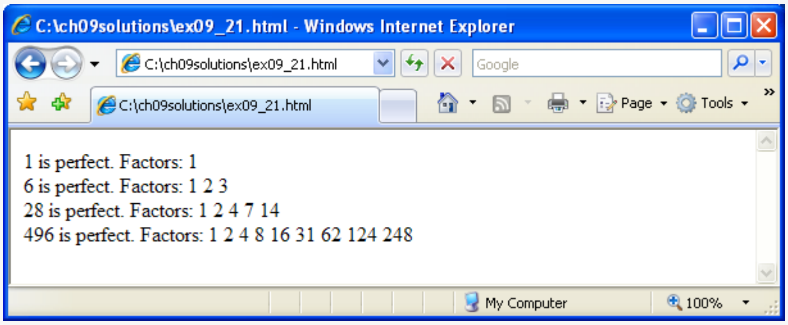
Smallest number: 14

9.21 An integer number is said to be a **perfect number** if its factors, including 1 (but not the number itself), sum to the number. For example, 6 is a perfect number, because 6 = 1 + 2 + 3. Write a function `perfect` that determines whether parameter `number` is a perfect number. Use this function in a script that determines and displays all the perfect numbers between 1 and 1000. Print the factors of each perfect number to confirm that the number is indeed perfect. Challenge the computing power of your computer by testing numbers much larger than 1000. Display the results in a `<textarea>`.

ANS:

```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Exercise 9.21: Solution -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Solution 9.21</title>
      <script type = "text/javascript">
         <!--
         var factors;

         function start()
         {
            for ( var i = 1; i <= 1000; ++i )
               if ( perfect( i ) == true )

```

```
                  document.write( i + " is perfect. Factors: "
                     + factors + "<br />");
         } // end function start

         function perfect( value )
         {
            var factorSum = 1;
            factors = "1";

            for ( var b = 2; b <= Math.floor( value / 2 ); b++ )

               if ( value % b == 0 )
               {
                  factorSum += b;
                  factors += "  " + b;
               } // end if

            if ( factorSum == value )
               return true;

            return false;
         } // end function perfect
         // -->
      </script>
   </head>
   <body onload = "start()">
      <h1>Perfect numbers from 1 to 1000</h1>
   </body>
</html>
```


C:\ch09solutions\ex09_21.html - Windows Internet Explorer
C:\ch09solutions\ex09_21.html
Google
C:\ch09solutions\ex09_21.html
Page
Tools
My Computer
100%

```
1 is perfect. Factors: 1
6 is perfect. Factors: 1 2 3
28 is perfect. Factors: 1 2 4 7 14
496 is perfect. Factors: 1 2 4 8 16 31 62 124 248
```

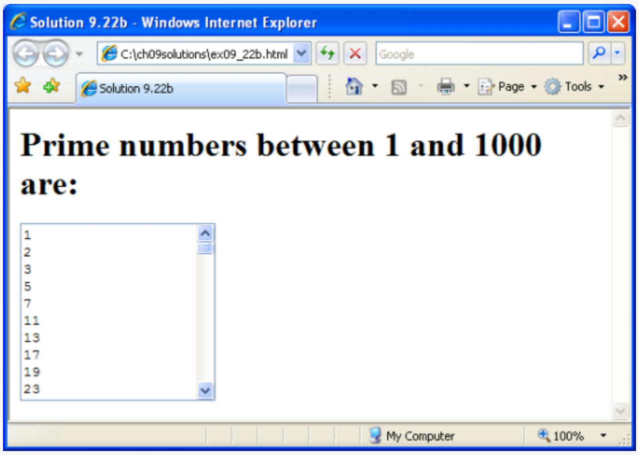
9.22 An integer is said to be prime if it is greater than 1 and divisible only by 1 and itself. For example, 2, 3, 5 and 7 are prime, but 4, 6, 8 and 9 are not.

a) Write a function that determines whether a number is prime.

b) Use this function in a script that determines and prints all the prime numbers between 1 and 10,000. How many of these 10,000 numbers do you really have to test before being sure that you have found all the primes? Display the results in a `<textarea>`.

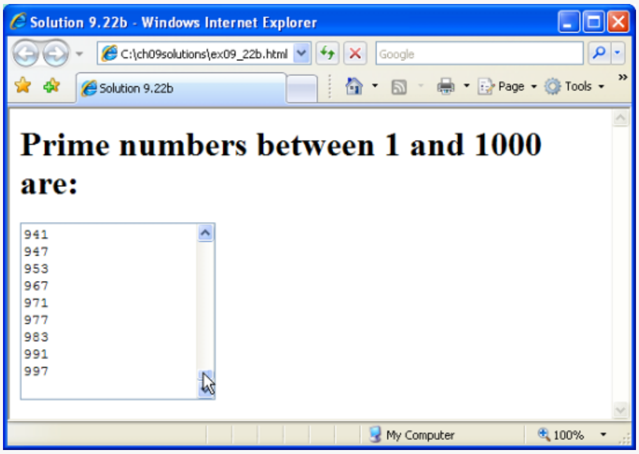
ANS:

```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Exercise 9.22b: Solution -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Solution 9.22b</title>
      <script type = "text/javascript">
         <!--
         document.writeln( "<h1>Prime numbers between 1 and"
            + " 1000 are: <br /></h1>" );
         document.writeln( "<textarea rows = '10'>");

         for ( var m = 1; m <= 1000; m++ )
            if ( prime( m ) )
            {
               document.writeln( m );
            } // end if
            document.writeln( "</textarea>" );

         function prime( n )
         {
            for ( var v = 2; v <= n / 2; v++ )
               if ( n % v == 0 )
                  return false;

            return true;
         } // end function prime
         // -->
      </script>
   </head>
   <body></body>
</html>
```


Solution 9.22b - Windows Internet Explorer
C:\ch09solutions\ex09_22b.html
Google
Solution 9.22b
Page
Tools
My Computer
100%

Prime numbers between 1 and 1000 are:

```
1
2
3
5
7
11
13
17
19
23
```


Solution 9.22b - Windows Internet Explorer
C:\ch09solutions\ex09_22b.html
Solution 9.22b
My Computer
100%

Prime numbers between 1 and 1000 are:

```
941
947
953
967
971
977
983
991
997
```

c) Initially, you might think that *n*/2 is the upper limit for which you must test to see whether a number is prime, but you only need go as high as the square root of *n*. Why? Rewrite the program, and run it both ways. Estimate the performance improvement.

ANS:

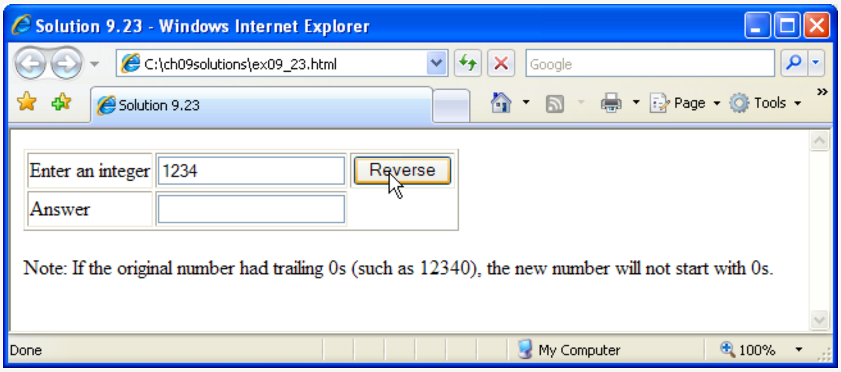
```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Exercise 9.22c: Solution -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Solution 9.22c</title>
      <script type = "text/javascript">
         <!--

         var count = 0;

         document.writeln( "<h1>Prime numbers between 1 and"
            + " 1000 are: <br /></h1>" );
         document.writeln( "<textarea rows = '10'>" );

         for ( var m = 1; m <= 1000; m++ )

            if ( prime( m ) == true )
            {
               ++count;
               document.writeln( m );
            } // end if

         document.writeln("</textarea>");

         function prime( n )
         {
            for ( var v = 2; v <= Math.sqrt( n ); v++ )

               if ( n % v == 0 )
                  return false;

            return true;
         } // end function prime
         // -->
      </script>
   </head>
   <body></body>
</html>
```

The performance is significantly enhanced. The browsers can now handle larger upper limits of numbers to test, such as 10,000, without having to prompt the user for permission to continue.

9.23 Write a function that takes an integer value and returns the number with its digits reversed. For example, given the number 7631, the function should return 1367. Incorporate the function into a script that reads a value from the user.

ANS:

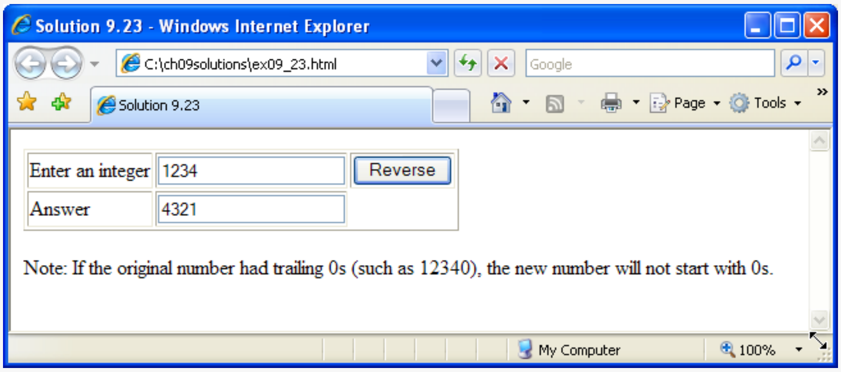
```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Exercise 9.23: Solution -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Solution 9.23</title>
      <script type = "text/javascript">
         <!--
         function reverseNumber()
         {
            var form = document.getElementById( "myForm" );
            var input = parseInt( form.number.value );
            form.answer.value = reverseInt( input );
         } // end function reverseNumber()

         // the following functions execute only when called
         function reverseInt( number )
         {
            var digit;
            var newNumber = 0;
            var multiplier = 1;
            var divisor = determineDivisor( number );

            while ( divisor >= 1 )
            {
               digit = quotient( number, divisor );
               newNumber += digit * multiplier
               number = remainder( number, divisor );
               divisor = quotient( divisor, 10 );
               multiplier *= 10;
            } // end while
            return newNumber;
         } // end function reverseInt

         function determineDivisor( number )
         {
            var divisor = 1;

            while ( quotient( number / 10, divisor ) != 0 )
               divisor *= 10;

            return divisor;
         } // end function determineDivisor

         function quotient( a, b )
         {
            return Math.floor( a / b );
         } // end function quotient

         function remainder( a, b )
         {
            return a % b;
         } // end function remainder
         // -->
      </script>
   </head>

   <body>
   <form id = "myForm" action = "">
      <table border = "1">
         <tr><td>Enter an integer</td>
             <td><input name = "number" type = "text" /></td>
             <td><input type = "button" value = "Reverse"
                onclick = "reverseNumber()" /></td>
         </tr>
         <tr><td>Answer</td>
             <td><input name = "answer" type = "text" /></td>
         </tr>
      </table>
    </form>
   <p>Note: If the original number had trailing 0s (such as
      12340), the new number will not start with 0s.</p>
   </body>
</html>
```


Solution 9.23 - Windows Internet Explorer
C:\ch09solutions\ex09_23.html
Google
Solution 9.23
Page
Tools
Done
My Computer
100%

Enter an integer 1234 Reverse

Answer

Note: If the original number had trailing 0s (such as 12340), the new number will not start with 0s.


Solution 9.23 - Windows Internet Explorer
C:\ch09solutions\ex09_23.html
Google
Solution 9.23
Page
Tools
My Computer
100%

Enter an integer 1234 Reverse

Answer 4321

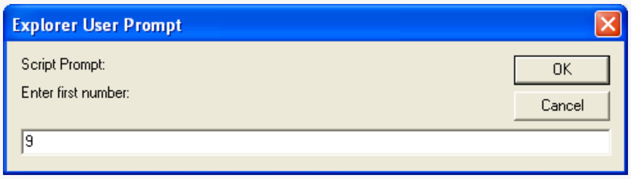
Note: If the original number had trailing 0s (such as 12340), the new number will not start with 0s.

9.24 The **greatest common divisor** (GCD) of two integers is the largest integer that evenly divides each of the two numbers. Write a function `gcd` that returns the greatest common divisor of two integers. Incorporate the function into a script that reads two values from the user.

ANS:

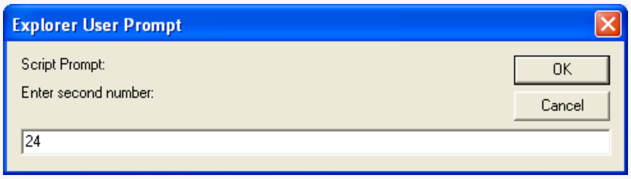
```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Exercise 9.24: Solution -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Solution 9.24</title>
      <script type = "text/javascript">
         <!--
         var num1 = window.prompt( "Enter first number:", "1" );
         var num2 = window.prompt( "Enter second number:", "1" );
         var result = gcd( num1, num2 );
         document.write( "GCD of " + num1 + " and " + num2 +
            " is " + result );

         function gcd( x, y )
         {
            var greatest = 1;
```

```

               for ( var z = 2; z <= Math.min( x, y ); z++ )
                  if ( ( x % z == 0 ) && ( y % z == 0 ) )
                     greatest = z;

               return greatest;
            } // function gcd
            // -->
         </script>
      </head>
      <body></body>
   </html>
```


Explorer User Prompt

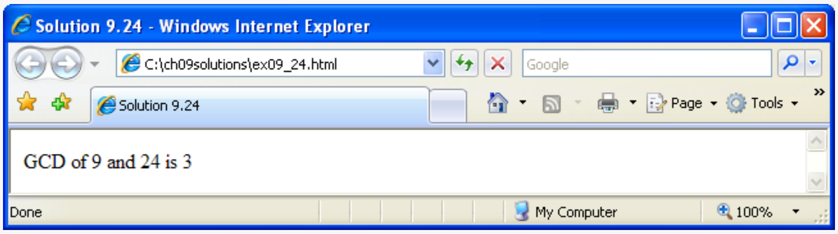
Script Prompt:
Enter first number:

9


Explorer User Prompt

Script Prompt:
Enter second number:

24


Solution 9.24 - Windows Internet Explorer

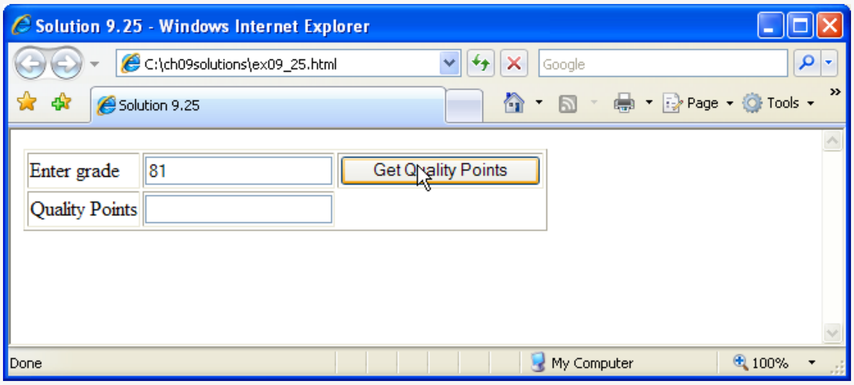
C:\ch09solutions\ex09_24.html

GCD of 9 and 24 is 3

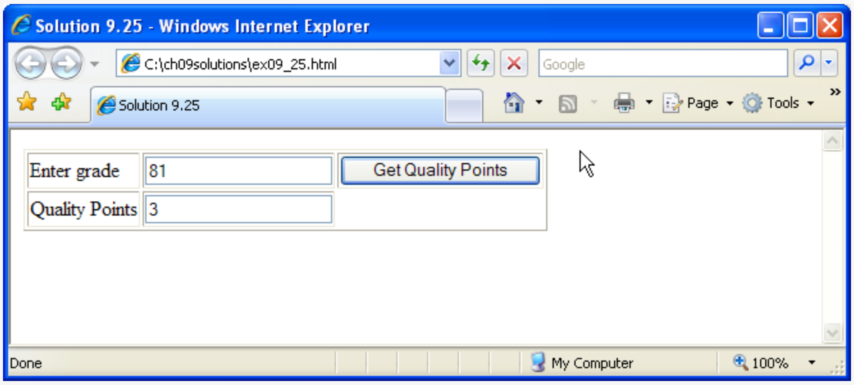
9.25 Write a function `qualityPoints` that inputs a student's average and returns 4 if the student's average is 90–100, 3 if the average is 80–89, 2 if the average is 70–79, 1 if the average is 60–69 and 0 if the average is lower than 60. Incorporate the function into a script that reads a value from the user.

ANS:

```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Exercise 9.25: Solution -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Solution 9.25</title>
      <script type = "text/javascript">
         <!--
         function determineQualityPoints()
         {
            var form = document.getElementById( "myForm" );
            var input = parseInt( form.number.value );

            form.points.value  =
               ( input >= 0 && input <= 100 ) ?
                  qualityPoints( input ) : "Invalid input.";
         } // end function determineQualityPoints

         function qualityPoints( grade )
         {
            if ( grade >= 90 )
               return 4;
            else if ( grade >= 80 )
               return 3;
            else if ( grade >= 70 )
               return 2;
            else if ( grade >= 60 )
               return 1;
            else
               return 0;
         } // end function qualityPoints
         // -->
      </script>
   </head>

   <body>
      <form id = "myForm" action = "">
         <table border = "1">
            <tr><td>Enter grade</td>
                <td><input name = "number" type = "text" />
                </td>
                <td><input type = "button" value =
                   "Get Quality Points" onclick =
                   "determineQualityPoints()" /></td>
            </tr><tr>
                <td>Quality Points</td>
                <td><input type = "text" name =
                   "points" /></td>
            </tr>
         </table>
      </form>
   </body>
</html>
```


Solution 9.25 - Windows Internet Explorer
C:\ch09solutions\ex09_25.html
Google
Solution 9.25
Page
Tools
Done
My Computer
100%

Enter grade 81 Get Quality Points
Quality Points

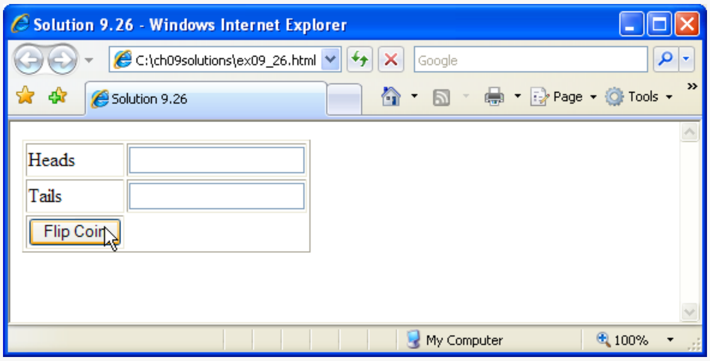

Solution 9.25 - Windows Internet Explorer
C:\ch09solutions\ex09_25.html
Google
Solution 9.25
Page
Tools
Done
My Computer
100%

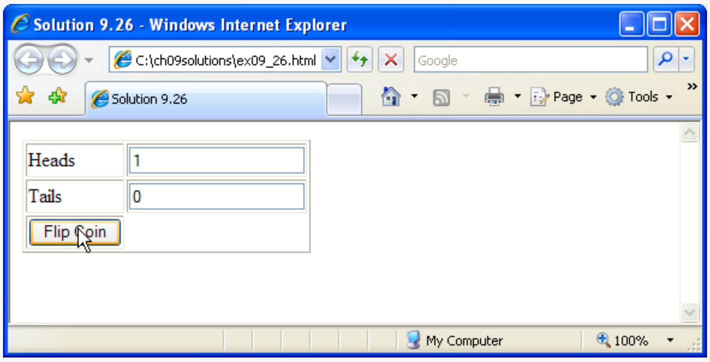
Enter grade 81 Get Quality Points
Quality Points 3

9.26 Write a script that simulates coin tossing. Let the program toss the coin each time the user clicks the **Toss** button. Count the number of times each side of the coin appears. Display the results. The program should call a separate function `flip` that takes no arguments and returns `false` for tails and `true` for heads. [*Note:* If the program realistically simulates the coin tossing, each side of the coin should appear approximately half the time.]

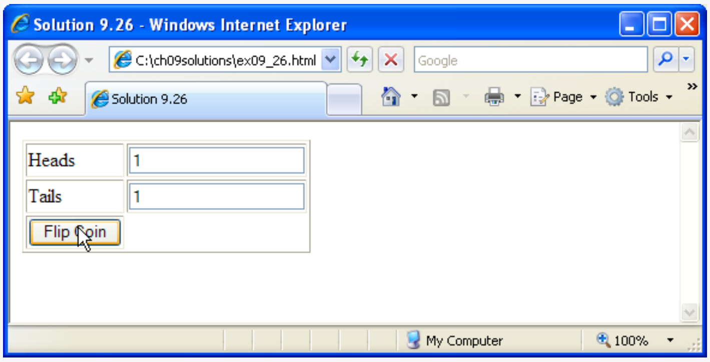
ANS:

```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Exercise 9.26: Solution -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Solution 9.26</title>
      <script type = "text/javascript">
         <!--
         var heads = 0;
         var tails = 0;

         function toss()
         {
            var headsField = document.getElementById( "heads" );
            var tailsField = document.getElementById( "tails" );

            if ( flip() )
            {
               ++heads;
               headsField.value = heads;
            }
            else
            {
               ++tails;
               tailsField.value = tails;
            }
         } // end function toss

         function flip()
         {
            return Math.floor( Math.random() * 2 ) == 1
         } // end function flip
         // -->
      </script>
   </head>

   <body>
      <form action = "">
         <table border = "1">
            <tr><td>Heads</td>
                <td><input id = "heads" type = "text" />
                </td>
            </tr>
            <tr><td>Tails</td>
                <td><input id = "tails" type = "text" />
                </td>
            </tr>
            <tr><td><input type = "button" value = "Flip Coin"
                           onclick = "toss()" /></td>
            </tr>
         </table>
      </form>
   </body>
</html>
```

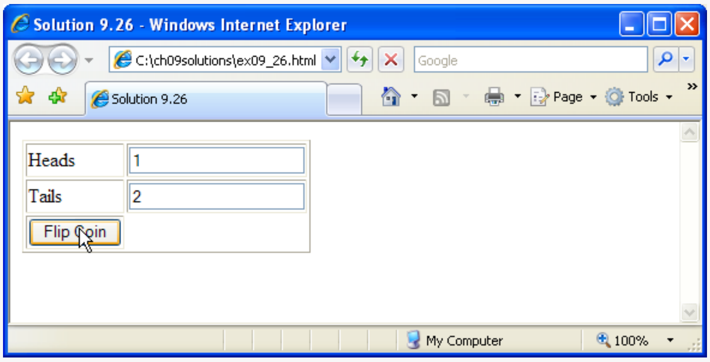

Solution 9.26 - Windows Internet Explorer
C:\ch09solutions\ex09_26.html
Google
Solution 9.26
Page
Tools
Heads
Tails
Flip Coin
My Computer
100%


Solution 9.26 - Windows Internet Explorer
C:\ch09solutions\ex09_26.html
Solution 9.26
My Computer
100%

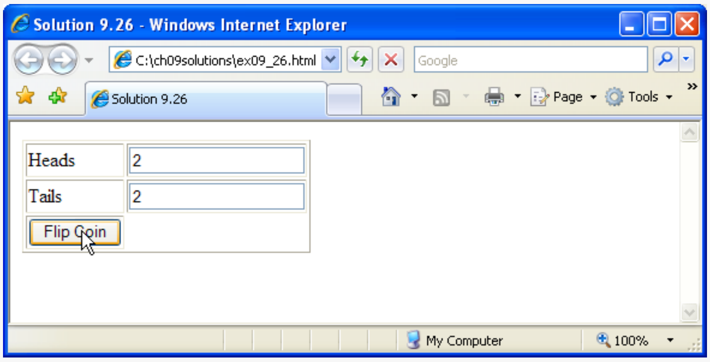
Heads	1
Tails	0
Flip Coin	


Solution 9.26 - Windows Internet Explorer
C:\ch09solutions\ex09_26.html
Solution 9.26
My Computer
100%

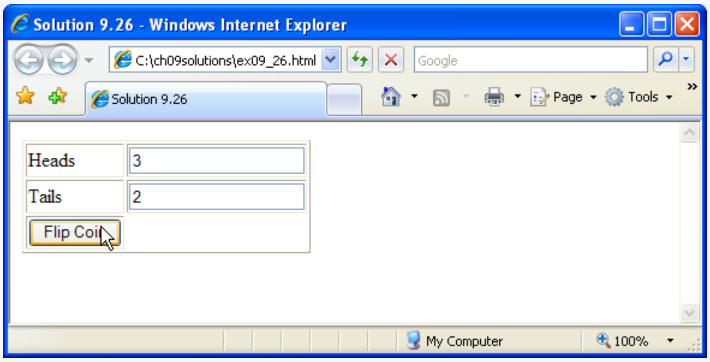
Heads	1
Tails	1
Flip Coin	


Solution 9.26 - Windows Internet Explorer
C:\ch09solutions\ex09_26.html
Solution 9.26
My Computer
100%

Heads	1
Tails	2
Flip Coin	


Solution 9.26 - Windows Internet Explorer
C:\ch09solutions\ex09_26.html
Solution 9.26
My Computer
100%

Heads	2
Tails	2
Flip Coin	


Solution 9.26 - Windows Internet Explorer
C:\ch09solutions\ex09_26.html
Google
Solution 9.26
Page
Tools
My Computer
100%

Heads	3
Tails	2
Flip Coin	

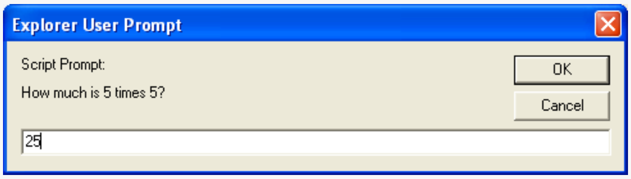
9.27 Computers are playing an increasing role in education. Write a program that will help an elementary-school student learn multiplication. Use `Math.random` to produce two positive one-digit integers. It should then display a question such as

```
How much is 6 times 7?
```

The student then types the answer into a text field. Your program checks the student's answer. If it is correct, display the string `"Very good!"` and generate a new question. If the answer is wrong, display the string `"No. Please try again."` and let the student try the same question again repeatedly until the student finally gets it right. A separate function should be used to generate each new question. This function should be called once when the script begins execution and each time the user answers the question correctly.

ANS:

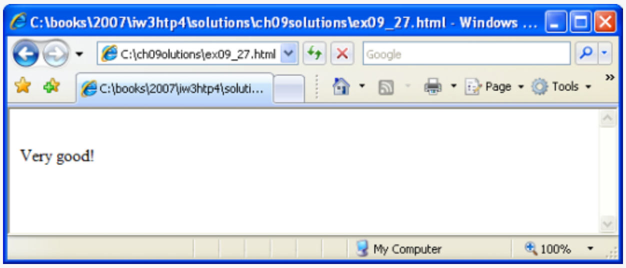
```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Exercise 9.27: Solution -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Solution 9.27</title>
      <script type = "text/javascript">
         <!--
         var number1;
         var number2;
         var answer;

         function run()
         {
            ask();

            while( window.prompt(
               "Would you like to answer another question?" +
                " ( yes or no ) ", "yes" ) == "yes" )
            {
               ask();
            } // end while
         } // end function run

         function ask()
         {
            number1 = Math.floor( 1 + Math.random() * 9 );
            number2 = Math.floor( 1 + Math.random() * 9 );

            while( true )
            {
               answer = window.prompt( "How much is " +
                  number1 + " times " + number2 + "?" );

               if ( parseInt( answer ) != number1 * number2 )
                  document.write( "<br />No. Please try again." );
               else
               {
                  document.write( "<br />Very good!" );
                  return;
               } // end else
            } // end while
         } // end function ask
         // -->
      </script>
   </head>
   <body onload = "run()">
   </body>
</html>
```



Explorer User Prompt

Script Prompt:
How much is 5 times 5?

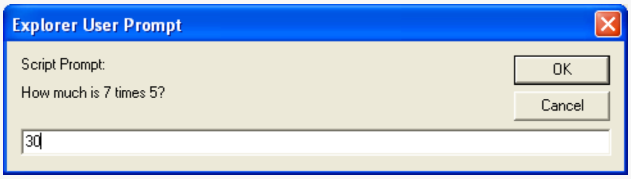
25



C:\books\2007\iw3htp4\solutions\ch09solutions\ex09_27.html - Windows ...

C:\ch09solutions\ex09_27.html
Google
C:\books\2007\iw3htp4\soluti...
Page
Tools

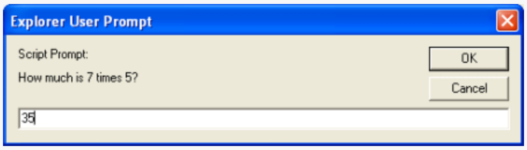
Very good!



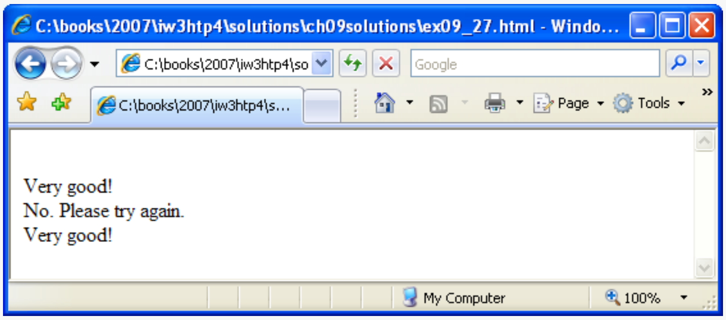
Explorer User Prompt

Script Prompt:
How much is 7 times 5?

30



Script Prompt:
How much is 7 times 5?
35


C:\books\2007\iw3htp4\solutions\ch09solutions\ex09_27.html - Windo...
Google
Page
Tools
My Computer
100%

```
Very good!
No. Please try again.
Very good!
```

9.28 The use of computers in education is referred to as **computer-assisted instruction** (CAI). One problem that develops in CAI environments is student fatigue. This problem can be eliminated by varying the computer's dialogue to hold the student's attention. Modify the program in Exercise 9.27 to print one of a variety of comments for each correct answer and each incorrect answer. The set of responses for correct answers is as follows:

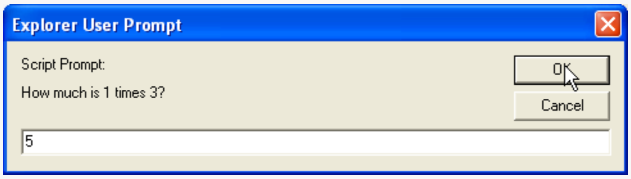
```
Very good!
Excellent!
Nice work!
Keep up the good work!
```

The set of responses for incorrect answers is as follows:

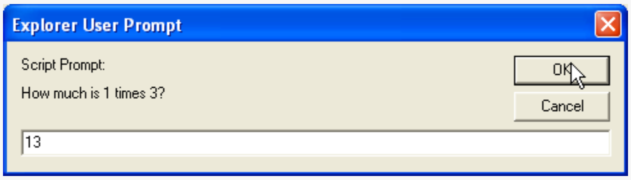
```
No. Please try again.
Wrong. Try once more.
Don't give up!
No. Keep trying.
```

Use random number generation to choose a number from 1 to 4 that will be used to select an appropriate response to each answer. Use a `switch` statement to issue the responses.

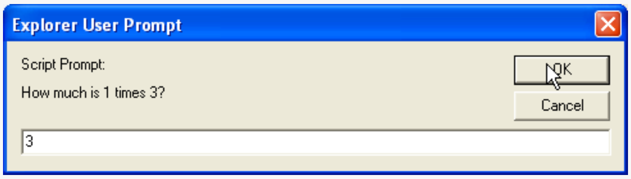
```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Exercise 9.28: Solution -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Solution 9.28</title>
      <script type = "text/javascript">
         <!--
         var number1;
         var number2;
         var answer;

         function run()
         {
            ask();

            while( window.prompt(
               "Would you like to answer another question?" +
                " ( yes or no ) ", "yes" ) == "yes" )
            {
               ask();
            } // end while
         } // end function run

         function ask()
         {
            number1 = Math.floor( 1 + Math.random() * 9 );
            number2 = Math.floor( 1 + Math.random() * 9 );

            while( true )
            {
               answer = window.prompt( "How much is " +
                  number1 + " times " + number2 + "?" );

               if ( parseInt( answer ) != number1 * number2 )
               {
                   response = Math.floor( 1 + Math.random() * 4 );

                  switch ( response )
                  {
                     case 1:
                        document.write( "No. Please try again." );
                        break;
                     case 2:
                        document.write( "Wrong. Try once more.");
                        break;
                     case 3:
                        document.write( "Don't give up!");
                        break;
                     case 4:
                        document.write( "No. Keep trying.");
                        break;
                  } // end switch
                  document.write( "<br />" );

               } // end if
               else
               {
                  response = Math.floor( 1 + Math.random() * 4 );

                  switch ( response )
                  {
                     case 1:
                        document.write( "Very good!");
                        break;
                     case 2:
                        document.write( "Excellent!");
                        break;
                     case 3:
                        document.write( "Nice work!");
                        break;
                     case 4:
                        document.write( "Keep up the good work!" );
                        break;
                  } // end switch
                  document.write( "<br />" );
                  return;
               } // end else
            } // end while
         } // end function ask
         // -->
      </script>
   </head>
   <body onload = "run()">
   </body>
</html>
```



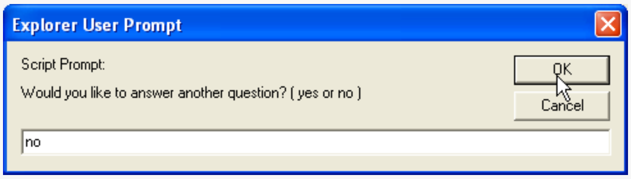
Script Prompt:
How much is 1 times 3?
5



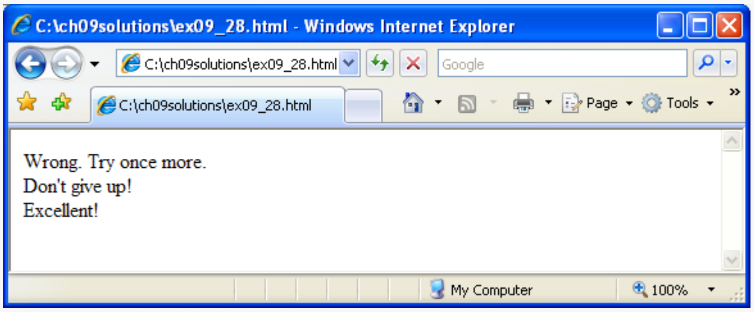
Script Prompt:
How much is 1 times 3?
13



Script Prompt:
How much is 1 times 3?
3



Script Prompt:
Would you like to answer another question? (yes or no)
no



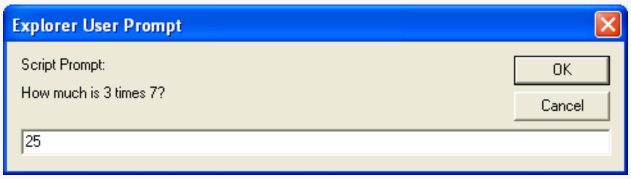
```
Wrong. Try once more.
Don't give up!
Excellent!
```

9.29 More sophisticated computer-assisted instruction systems monitor the student's performance over a period of time. The decision to begin a new topic is often based on the student's success with previous topics. Modify the program in Exercise 9.28 to count the number of correct and incorrect responses typed by the student. After the student answers 10 questions, your program should calculate the percentage of correct responses. If the percentage is lower than 75 percent, print `Please ask your instructor for extra help`, and reset the program so another student can try it.

ANS:

```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Exercise 9.29: Solution -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Solution 9.29</title>
      <script type = "text/javascript">
         <!--
         var number1;
         var number2;
         var answer;

         function run()
         {
            while ( true )
            {
               var correct = 0;

               for (var i = 0; i < 10; i++ )
                  correct += ask();

               var getHelp = "";

               if ( correct <= 7 )
                  getHelp = "Please see your instructor for extra help"

               alert( "Total correct: " + correct + " " + getHelp );
               var tenMore = window.prompt(
                  "Would you like to answer ten more? (yes or no)",
                  "yes");
               if ( tenMore == "no" )
                  return;
            } // end while
         } // end function run

         function ask()
         {
            number1 = Math.floor( 1 + Math.random() * 9 );
            number2 = Math.floor( 1 + Math.random() * 9 );

            while ( true )
            {
               answer = window.prompt( "How much is " +
                  number1 + " times " + number2 + "?" );

               if ( parseInt( answer ) != number1 * number2 )
               {
                  response = Math.floor( 1 + Math.random() * 4 );

                  switch ( response )
                  {
                     case 1:
                        alert( "No." );
                        break;
                     case 2:
                        alert( "Wrong. Try another one." );
                        break;
                     case 3:
                        alert( "Don't give up!" );
                        break;
                     case 4:
                        alert( "No. Keep trying." );
                        break;
                  } // end switch
                  return 0;
               } // end if
               else
               {
                  response = Math.floor( 1 + Math.random() * 4 );

                  switch ( response )
                  {
                     case 1:
                        alert( "Very good!" );
                        break;
                     case 2:
                        alert( "Excellent!" );
                        break;
                     case 3:
                        alert( "Fantastic!" );
                        break;
                     case 4:
                        alert( "Keep up the good work!" );
                        break;
                  } // end switch
```

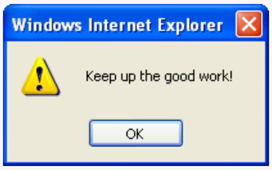
```
                  return 1;
               } // end else
            } // end while
         } // end function ask
         // -->
      </script>
   </head>
   <body onload = "run()">
   </body>
</html>
```



Explorer User Prompt

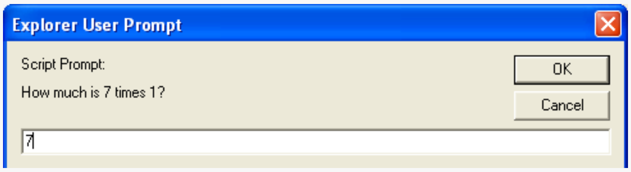
Script Prompt:
How much is 3 times 7?

25



Windows Internet Explorer

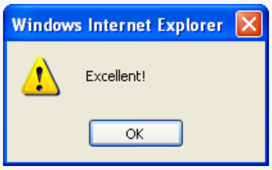
Keep up the good work!



Explorer User Prompt

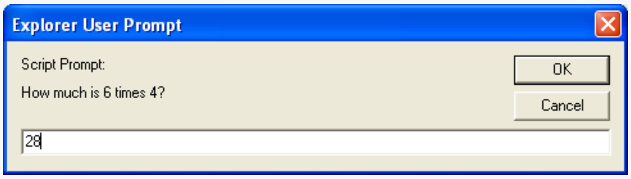
Script Prompt:
How much is 7 times 1?

7



Windows Internet Explorer

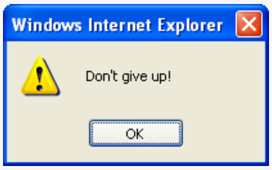
Excellent!



Explorer User Prompt

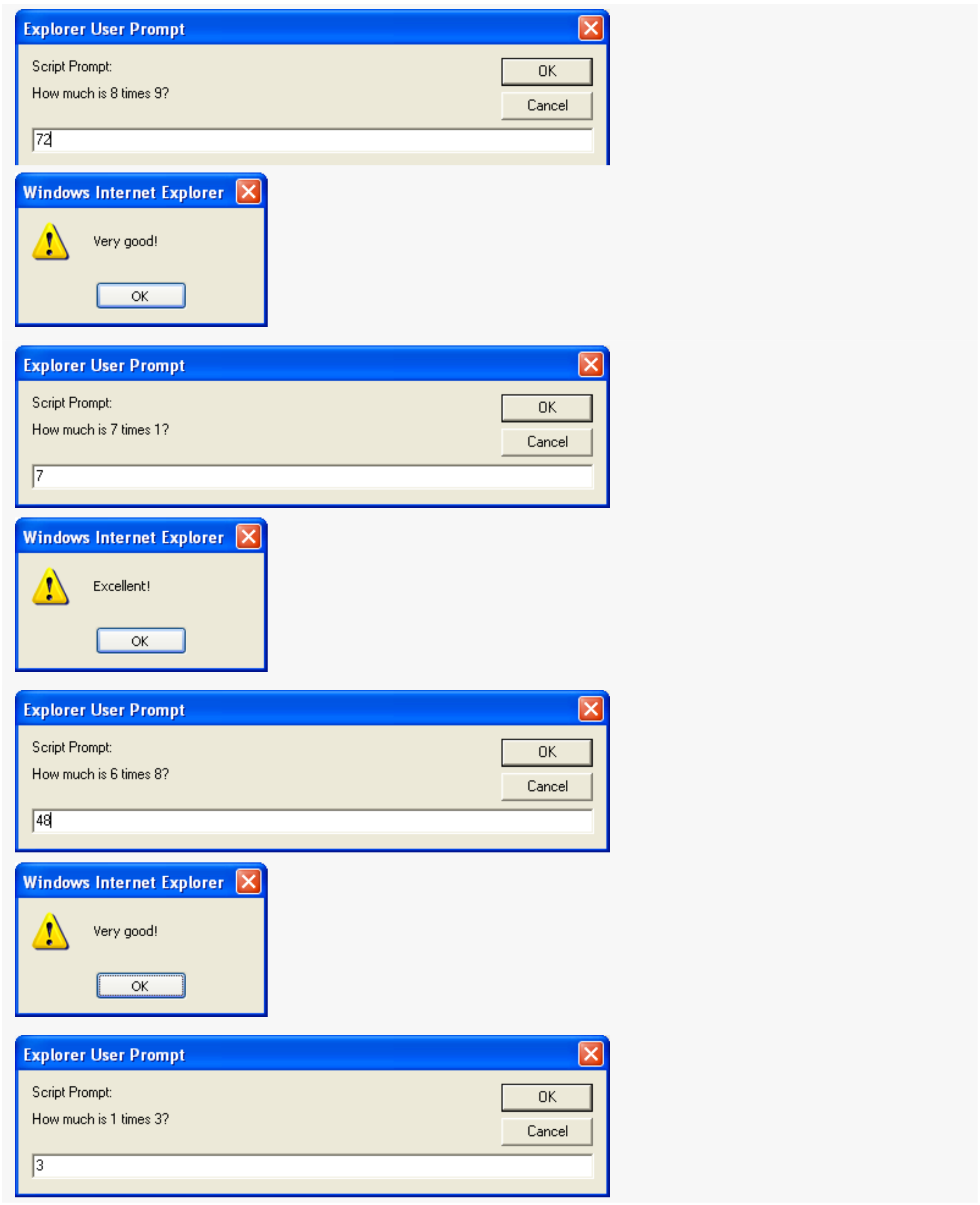
Script Prompt:
How much is 6 times 4?

28



Windows Internet Explorer

Don't give up!


Explorer User Prompt

Script Prompt:

How much is 8 times 9?

72

Windows Internet Explorer

Very good!

Explorer User Prompt

Script Prompt:

How much is 7 times 1?

7

Windows Internet Explorer

Excellent!

Explorer User Prompt

Script Prompt:

How much is 6 times 8?

48

Windows Internet Explorer

Very good!

Explorer User Prompt

Script Prompt:

How much is 1 times 3?

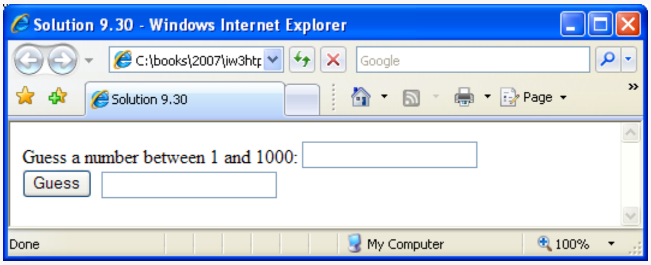
3

9.30 Write a script that plays a "guess the number" game as follows: Your program chooses the number to be guessed by selecting a random integer in the range 1 to 1000. The script displays the prompt `Guess a number between 1 and 1000` next to a text field. The player types a first guess into the text field and clicks a button to submit the guess to the script. If the player's guess is incorrect, your program should display `Too high. Try again.` or `Too low. Try again.` to help the player "zero in" on the correct answer and should clear the text field so the user can enter the next guess. When the user enters the correct answer, display `Congratulations. You guessed the number!` and clear the text field so the user can play again. [*Note*: The guessing technique employed in this problem is similar to a **binary search**, which we discuss in Chapter 10, JavaScript: Arrays.]

ANS:

```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Exercise 9.30: Solution -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Solution 9.30</title>
      <script type = "text/javascript">
         <!--
         function check()
         {
            var hintField = document.getElementById( "hint" );
            var targetField = document.getElementById( "target" );
            var guessField = document.getElementById( "userguess" );

            if ( parseInt( guessField.value ) > targetField.value )
               hintField.value = "Too high. Try again.";
            else if ( parseInt( guessField.value ) < targetField.value )
               hintField.value = "Too low. Try again.";
            else
            {
               hintField.value = "You guessed it!";
               changevalue();
            } // end else
         } // end function check

         function changevalue()
         {
            document.getElementById( "target" ).value =
               Math.floor( 1 + Math.random() * 1000 );
         } // end function changevalue
         // -->
      </script>
   </head>

   <body onload = "changevalue()">
      <form id = "guess" action = "">
         <div>
            Guess a number between 1 and 1000:
            <input type = "text" id = "userguess" /><br />
            <input type = "button" value = "Guess"
               onclick = "check()" />
            <input type = "hidden" id = "target" value = "0" />
            <label>
               <input id = "hint" type = "text" />
            </label>
         </div>
      </form>
```

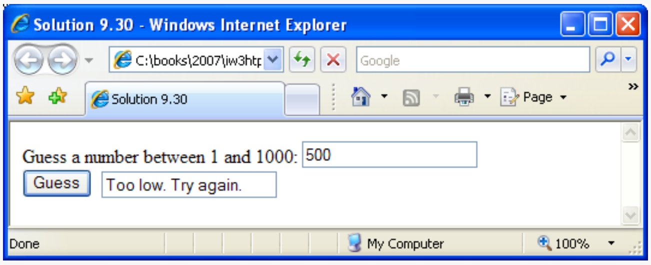
```
    </body>
</html>
```



Solution 9.30 - Windows Internet Explorer

Guess a number between 1 and 1000:

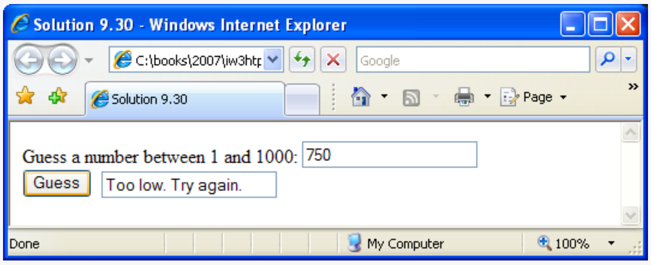
Guess



Solution 9.30 - Windows Internet Explorer

Guess a number between 1 and 1000: 500

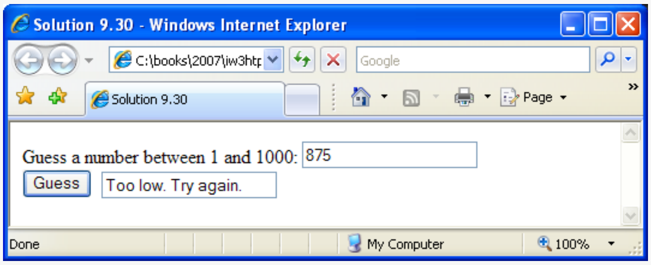
Guess Too low. Try again.



Solution 9.30 - Windows Internet Explorer

Guess a number between 1 and 1000: 750

Guess Too low. Try again.



Solution 9.30 - Windows Internet Explorer

Guess a number between 1 and 1000: 875

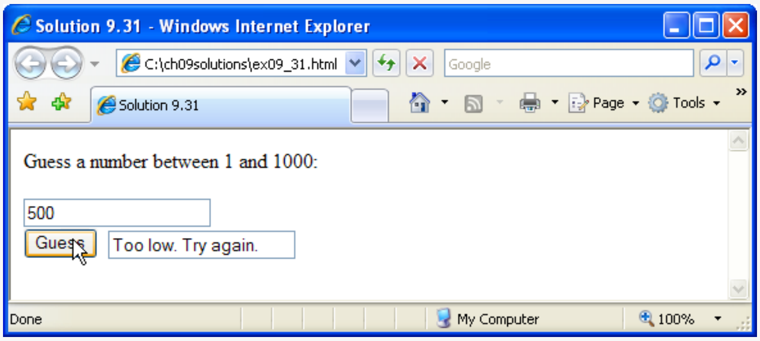
Guess Too low. Try again.

9.31 Modify the program of Exercise 9.30 to count the number of guesses the player makes. If the number is 10 or fewer, display `Either you know the secret or you got lucky!` If the player guesses the number in 10 tries, display `Ahah! You know the secret!` If the player makes more than 10 guesses, display `You should be able to do better!` Why should it take no more than 10 guesses? Well, with each good guess, the player should be able to eliminate half of the numbers. Now show why any number 1 to 1000 can be guessed in 10 or fewer tries.

ANS:

```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Exercise 9.31: Solution -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Solution 9.31</title>
      <script type = "text/javascript">
         <!--
         var numGuesses = 0;

         function check()
         {
            var form = document.getElementById( "guess" );
            if ( parseInt( form.userguess.value ) >
                  form.target.value )
               form.hint.value = "Too high. Try again.";
            else if ( parseInt( form.userguess.value ) <
                  form.target.value )
               form.hint.value = "Too low. Try again.";
            else {
               form.hint.value =
                  "You guessed it!";
               if ( parseInt( numGuesses ) < 10 )
                  window.alert( "Either you know the secret or"
                     + " you got lucky!");
               if ( parseInt( numGuesses ) == 10 )
                  window.alert( "Ahah! You know the secret!" );
               if ( parseInt( numGuesses ) > 10 )
                  window.alert(
                     "You should be able to do better!" );

               guess.userguess.value = "";
               window.status = "Done";
               changevalue();
            } // end else
            numGuesses++;
         } // end function check

         function changevalue()
         {
            document.getElementById( "guess" ).target.value =
               Math.floor( 1 + Math.random() * 1000 );
         } // end function changevalue
         // -->
      </script>
   </head>

   <body onload = "changevalue()">
      <form id = "guess" action = "">
         <div>
```

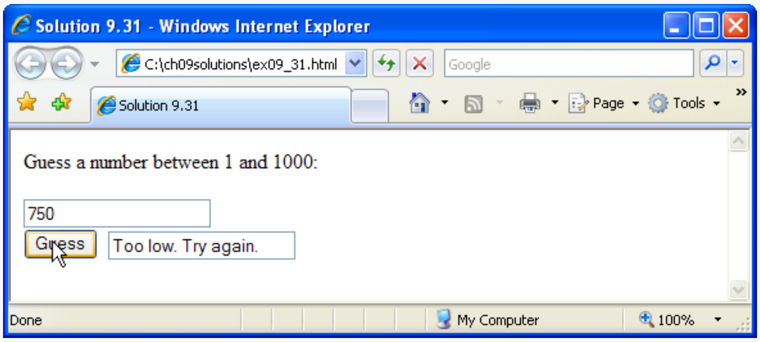
```
                <p>Guess a number between 1 and 1000:</p>
                <input type = "text" name = "userguess" /><br />
                <input type = "button" value = "Guess"
                   onclick = "check()" />
                <input type = "hidden" name = "target" value = "0" />
                <label>
                   <input name = "hint" type = "text" />
                </label>
             </div>
          </form>
       </body>
    </html>
```


Solution 9.31 - Windows Internet Explorer
C:\ch09solutions\ex09_31.html
Google
Solution 9.31
Page
Tools
Done
My Computer
100%

Guess a number between 1 and 1000:

500

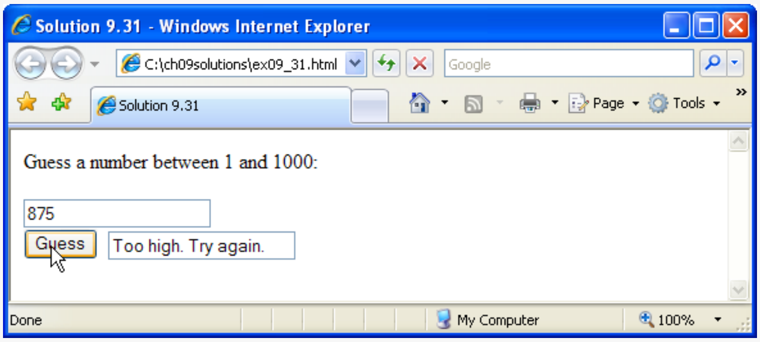
Guess | Too low. Try again.


Solution 9.31 - Windows Internet Explorer
C:\ch09solutions\ex09_31.html
Google
Solution 9.31
Page
Tools
Done
My Computer
100%

Guess a number between 1 and 1000:

750

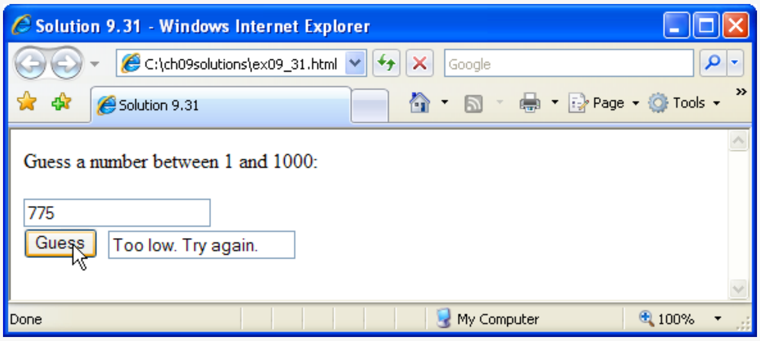
Guess | Too low. Try again.


Solution 9.31 - Windows Internet Explorer
C:\ch09solutions\ex09_31.html
Google
Solution 9.31
Page
Tools
Done
My Computer
100%

Guess a number between 1 and 1000:

875

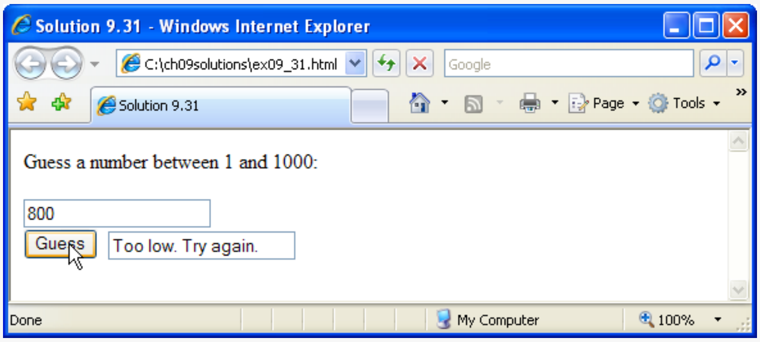
Guess | Too high. Try again.


Solution 9.31 - Windows Internet Explorer
C:\ch09solutions\ex09_31.html
Done
My Computer
100%

Guess a number between 1 and 1000:

775

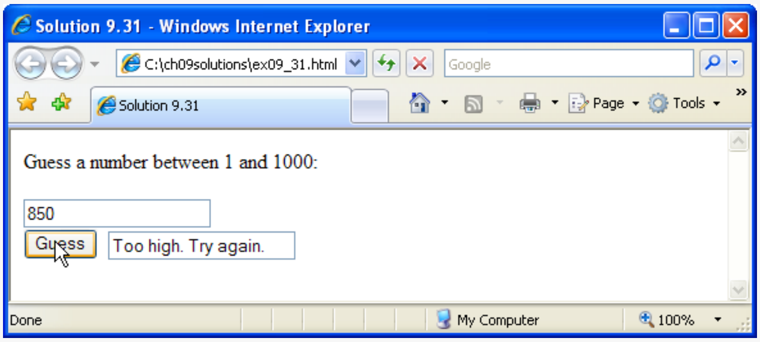
Guess Too low. Try again.


Solution 9.31 - Windows Internet Explorer
C:\ch09solutions\ex09_31.html
Done
My Computer
100%

Guess a number between 1 and 1000:

800

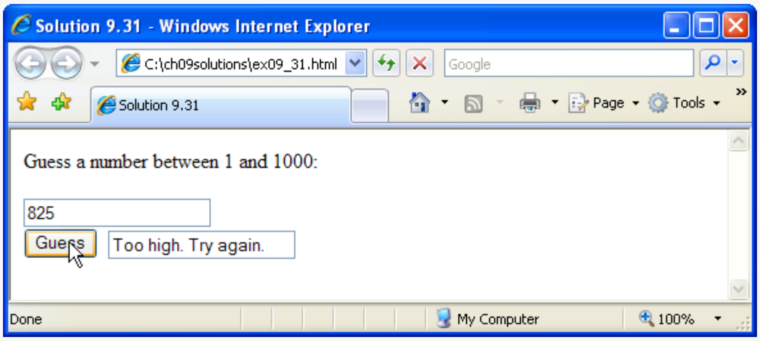
Guess Too low. Try again.


Solution 9.31 - Windows Internet Explorer
C:\ch09solutions\ex09_31.html
Done
My Computer
100%

Guess a number between 1 and 1000:

850

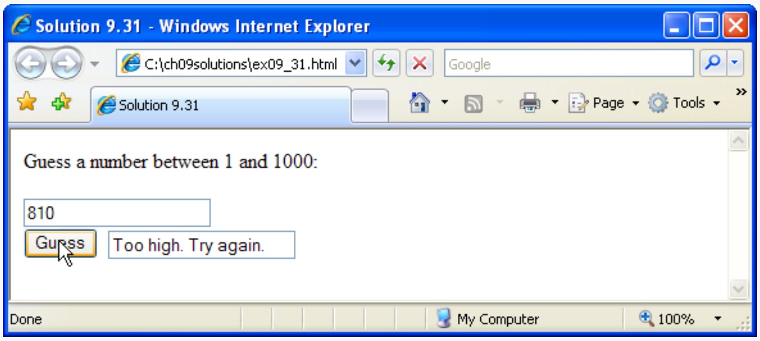
Guess Too high. Try again.


Solution 9.31 - Windows Internet Explorer
C:\ch09solutions\ex09_31.html
Done
My Computer
100%

Guess a number between 1 and 1000:

825

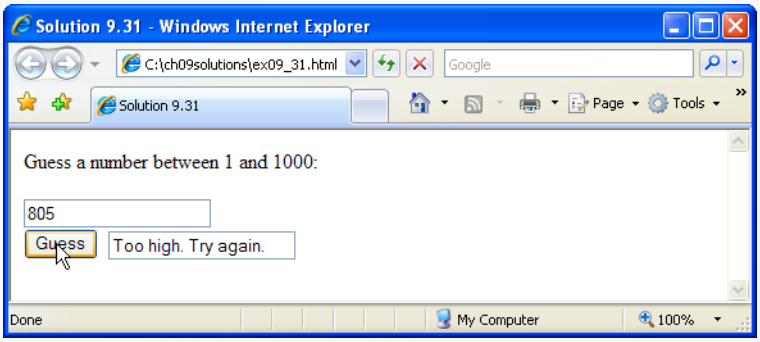
Guess Too high. Try again.


Solution 9.31 - Windows Internet Explorer
C:\ch09solutions\ex09_31.html
Google
Solution 9.31
Page
Tools
Done
My Computer
100%

Guess a number between 1 and 1000:

810

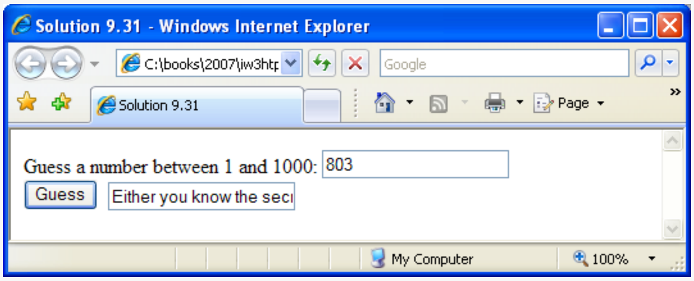
Guess Too high. Try again.


Solution 9.31 - Windows Internet Explorer
C:\ch09solutions\ex09_31.html
Google
Solution 9.31
Page
Tools
Done
My Computer
100%

Guess a number between 1 and 1000:

805

Guess Too high. Try again.



Guess a number between 1 and 1000: 803

Guess Either you know the seci

With each guess, we can eliminate one-half of the possible guessing pool if we guess the number exactly halfway into the current guessing range. So after the first guess, there are still 500 possibilites; after the second, there are 250, etc. The number of remaining possibilities can be generalized with the expression 1000 * (1/2) $^{\text{(number of guesses completed)}}$. So after 9 guesses, there are fewer than 2 posssible answers. This can be rounded down to 1, so by using this binary guessing technique, the answer can always be obtained in 10 or fewer guesses.

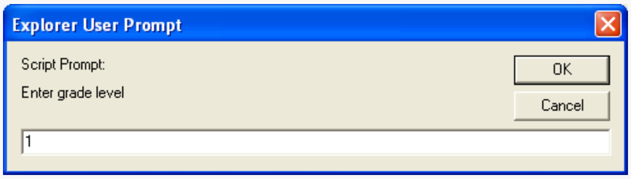
9.32 Exercises 9.27 through 9.29 developed a computer-assisted instruction program to teach an elementary-school student multiplication. This exercise suggests enhancements to that program.

a) Modify the program to allow the user to enter a grade-level capability. A grade level of 1 means to use only single-digit numbers in the problems, a grade level of 2 means to use numbers as large as two digits, and so on.

ANS:

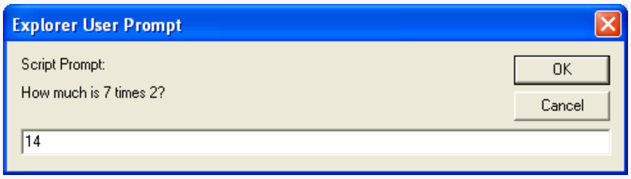
```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Exercise 9.32a: Solution -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Solution 9.32a</title>
      <script type = "text/javascript">
         <!--
         var number1;
         var number2;
         var answer;
         var scaleFactor;

         function run()
         {
            var gradeLevel = window.prompt( "Enter grade level" );
            scaleFactor = Math.pow( 10, gradeLevel ) - 1;

            ask();

            while ( window.prompt(
               "Would you like to answer another question?" +
                " ( yes or no ) ", "yes" ) == "yes" )
            {
               ask();
            } // end while
         } // end function run

         function ask()
         {
            number1 = Math.floor( 1 + Math.random() * scaleFactor );
            number2 = Math.floor( 1 + Math.random() * scaleFactor );

            while( true )
            {
               answer = window.prompt( "How much is " +
                  number1 + " times " + number2 + "?" );

               if ( parseInt( answer ) != number1 * number2 )
                  document.write( "<br />No. Please try again." );
               else
               {
                  document.write( "<br />Very good!" );
                  return;
               } // end else
            } // end while
         } // end function ask
         // -->
      </script>
   </head>
```

```
    <body onload = "run()">
    </body>
</html>
```



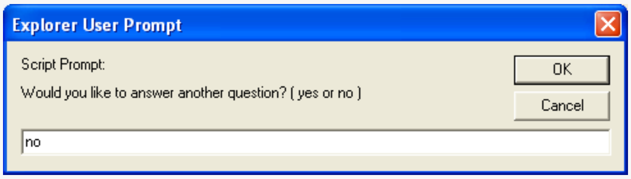
Script Prompt:
Enter grade level

```
1
```



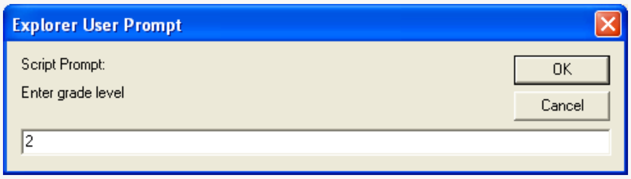
Script Prompt:
How much is 7 times 2?

```
14
```



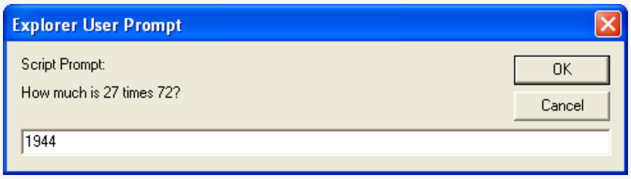
Script Prompt:
Would you like to answer another question? (yes or no)

```
no
```



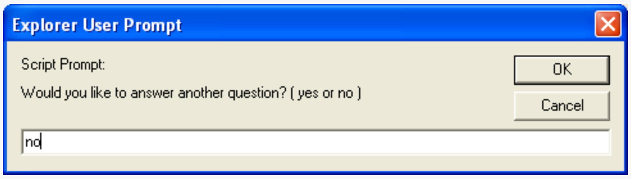
Script Prompt:
Enter grade level

```
2
```



Script Prompt:
How much is 27 times 72?

```
1944
```



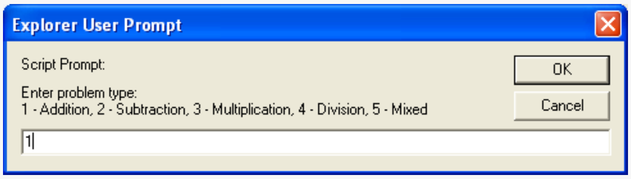
Script Prompt:
Would you like to answer another question? (yes or no)

```
no
```

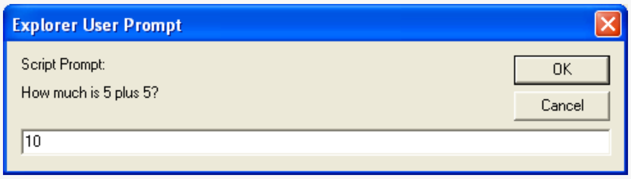
b) Modify the program to allow the user to pick the type of arithmetic problems he or she wishes to study. An option of 1 means addition problems only, 2 means subtraction problems only, 3 means multiplication problems only, 4 means division problems only and 5 means to intermix randomly problems of all these types.

ANS:

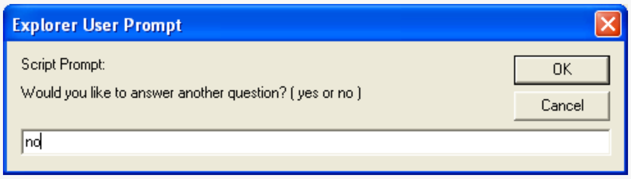
```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Exercise 9.32b: Solution -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Solution 9.32b</title>
      <script type = "text/javascript">
         <!--
         var number1;
         var number2;
         var answer;
         var operation;
         var operationWord;
         var correctAnswer;
         var randomOperator = false;

         function run()
         {
            operation = parseInt( window.prompt(
                                  "Enter problem type: \n" +
                                  "1 - Addition, " +
                                  "2 - Subtraction, " +
                                  "3 - Multiplication, " +
                                  "4 - Division, " +
                                  "5 - Mixed" ) );
            ask();

            while ( window.prompt(
               "Would you like to answer another question?" +
               " ( yes or no ) ", "yes" ) == "yes" )
            {
               ask();
            } // end while
         } // end function run

         function ask()
         {
            number1 = Math.floor( 1 + Math.random() * 9 );
            number2 = Math.floor( 1 + Math.random() * 9 );

            if ( !setOperation( operation ) )
               return;

            while( true )
            {
               if ( randomOperator )
                  setOperation( 5 ); // set a random operation

               answer = window.prompt( "How much is " +
                  number1 + operationWord + number2 + "?" );

               if ( parseInt( answer ) != correctAnswer )
                  document.write( "<br />No. Please try again." );
               else
               {
                  document.write( "<br />Very good!" );
                  return;
               } // end else
            } // end while
         } // end function ask

         function setOperation( op )
         {
            switch ( op )
            {
            case 1:
               correctAnswer = number1 + number2;
               operationWord = " plus ";
               break;
            case 2:
               correctAnswer = number1 - number2;
               operationWord = " minus ";
               break;
            case 3:
               correctAnswer = number1 * number2;
               operationWord = " times ";
               break;
            case 4:
               correctAnswer = number1;
               number1 = number1 * number2;
               operationWord = " divided by ";
               break;
            case 5:
               randomOperator = true;
               setOperation( Math.floor( 1 + Math.random() * 4 ) );
               break;
            default:
               alert( "Operation type must be 1-5." );
               return false;
            } // end switch
            return true;
         } // end function setOperation
         // -->
      </script>
   </head>
   <body onload = "run()">
   </body>
</html>
```



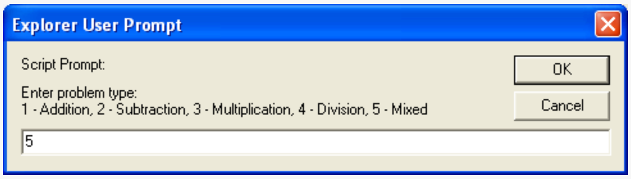
```
Script Prompt:
Enter problem type:
1 - Addition, 2 - Subtraction, 3 - Multiplication, 4 - Division, 5 - Mixed
1
```



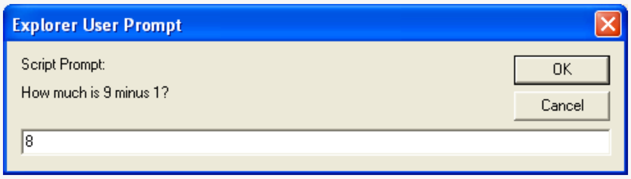
```
Script Prompt:
How much is 5 plus 5?
10
```



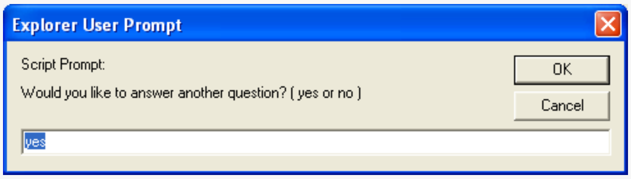
```
Script Prompt:
Would you like to answer another question? ( yes or no )
no
```



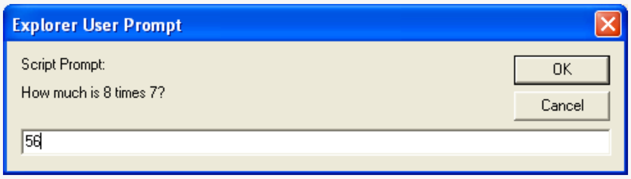
```
Script Prompt:
Enter problem type:
1 - Addition, 2 - Subtraction, 3 - Multiplication, 4 - Division, 5 - Mixed
5
```



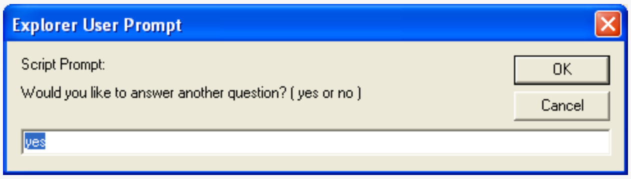
```
Script Prompt:
How much is 9 minus 1?
8
```



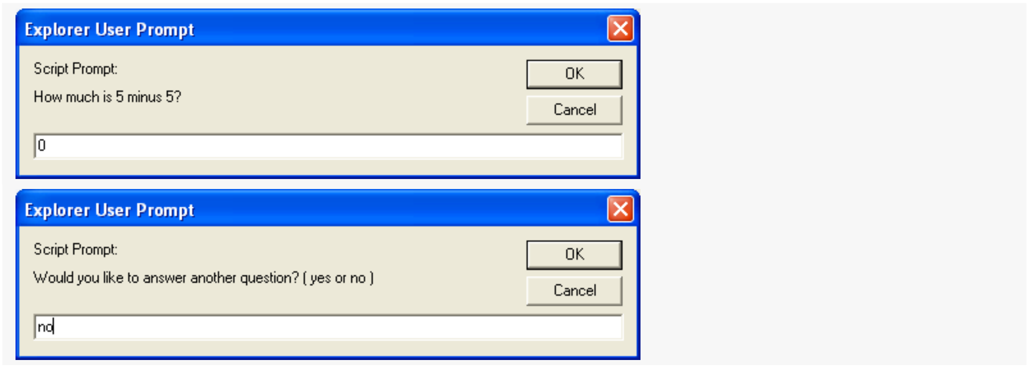
```
Script Prompt:
Would you like to answer another question? ( yes or no )
yes
```



```
Script Prompt:
How much is 8 times 7?
56
```



```
Script Prompt:
Would you like to answer another question? ( yes or no )
yes
```


Explorer User Prompt

Script Prompt:
How much is 5 minus 5?

0

Explorer User Prompt

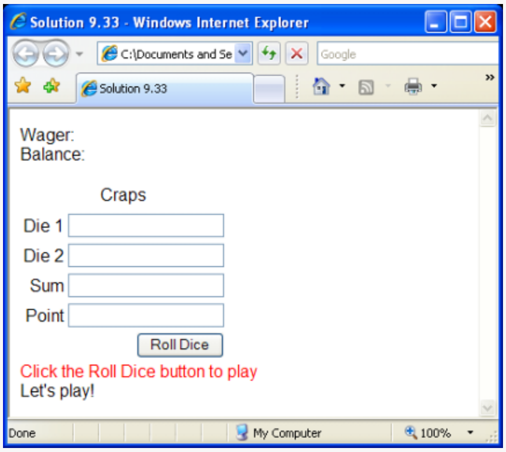
Script Prompt:
Would you like to answer another question? (yes or no)

no

9.33 Modify the craps program in Fig. 9.6 to allow wagering. Initialize variable `bankBalance` to 1000 dollars. Prompt the player to enter a `wager`. Check that the `wager` is less than or equal to `bankBalance` and, if not, have the user reenter `wager` until a valid `wager` is entered. After a valid `wager` is entered, run one game of craps. If the player wins, increase `bankBalance` by `wager`, and print the new `bankBalance`. If the player loses, decrease `bankBalance` by `wager`, print the new `bankBalance`, check whether `bankBalance` has become zero and, if so, print the message `Sorry. You busted!` As the game progresses, print various messages to create some chatter, such as `Oh, you're going for broke, huh?` or `Aw c'mon, take a chance!` or `You're up big. Now's the time to cash in your chips!`. Implement the chatter as a separate function that randomly chooses the string to display.

ANS:

```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Exercise 9.33: Solution -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Solution 9.33</title>
      <style type = "text/css">
         table   { text-align: right }
         body    { font-family: arial, sans-serif }
         div.red { color: red }
      </style>
      <script type = "text/javascript">
         <!--
         // variables used to test the state of the game
         var WON = 0;
         var LOST = 1;
         var CONTINUE_ROLLING = 2;

         // other variables used in program
         var firstRoll = true; // true if current roll is first
         var sumOfDice = 0; // sum of the dice
         var myPoint = 0; // point if no win/loss on first roll
         var gameStatus = CONTINUE_ROLLING; // game not over yet
         var balance = 1000; // the current balance
         var wager; // the player's wager

         // process one roll of the dice
         function play()
         {
            chatter();

            // get the point field on the page
            var point = document.getElementById( "pointfield" );

            // get the status div on the page
            var statusDiv = document.getElementById( "status" );
            if ( firstRoll ) // first roll of the dice
            {
               do
               {
                  wager = parseInt( window.prompt( "Enter a wager:" ) );
               }
               while ( wager > balance );

               balance -= wager;
               document.getElementById( "wager" ).innerHTML = "Wager: "
                  + wager;
               document.getElementById( "balance" ).innerHTML =
                  "Balance: " + balance;
               sumOfDice = rollDice();

               switch ( sumOfDice )
               {
                  case 7: case 11: // win on first roll
                     gameStatus = WON;
                     // clear point field
                     point.value = "";
                     break;
                  case 2: case 3: case 12: // lose on first roll
                     gameStatus = LOST;
                     // clear point field
                     point.value = "";
                     break;
                  default: // remember point
                     gameStatus = CONTINUE_ROLLING;
                     myPoint = sumOfDice;
                     point.value = myPoint;
                     firstRoll = false;
               } // end switch
            } // end if
            else
            {
               sumOfDice = rollDice();

               if ( sumOfDice == myPoint ) // win by making point
                  gameStatus = WON;
               else
                  if ( sumOfDice == 7 )    // lose by rolling 7
                     gameStatus = LOST;
            } // end else

            if ( gameStatus == CONTINUE_ROLLING )
               statusDiv.innerHTML = "Roll again";
            else
            {
               if ( gameStatus == WON )
               {
                  statusDiv.innerHTML = "Player wins. " +
                     "Click Roll Dice to play again.";
                  balance += 2 * wager;
                  document.getElementById( "balance" ).innerHTML =
                     "Balance: " + balance;
               }
               else
               {
                  statusDiv.innerHTML = "Player loses. " +
                     "Click Roll Dice to play again.";

                  if ( balance <= 0 )
                  {
                     alert( "You busted! Play again." );
                     balance = 1000;
                     document.getElementById( "balance" ).innerHTML =
                     "Balance: " + balance;
                  }
               }
               firstRoll = true;
            } // end else
         } // end function play

         // roll the dice
         function rollDice()
         {
            var die1;
            var die2;
            var workSum;

            die1 = Math.floor( 1 + Math.random() * 6 );
            die2 = Math.floor( 1 + Math.random() * 6 );
            workSum = die1 + die2;

            document.getElementById( "die1field" ).value = die1;
            document.getElementById( "die2field" ).value = die2;
            document.getElementById( "sumfield" ).value = workSum;

            return workSum;
         } // end function rollDice

         function chatter()
         {
            var chatterBox = document.getElementById( "chatter" )

            if ( balance < 500 )
               chatterBox.innerHTML = "You're going broke!";
            else if ( balance > 1500 )
               chatterBox.innerHTML = "You're doing well!";
            else if ( balance > 3000 )
               chatterBox.innerHTML = "You'd better cash in!";
            else
               chatterBox.innerHTML = "You haven't lost big yet...";
         }
         // -->
      </script>
   </head>
   <body>
      <div id = "wager">Wager: </div><div id = "balance">Balance: </div>
      <form action = "">
         <table>
         <caption>Craps</caption>
         <tr><td>Die 1</td>
            <td><input id = "die1field" type = "text" />
            </td></tr>
         <tr><td>Die 2</td>
            <td><input id = "die2field" type = "text" />
            </td></tr>
         <tr><td>Sum</td>
            <td><input id = "sumfield" type = "text" />
            </td></tr>
         <tr><td>Point</td>
            <td><input id = "pointfield" type = "text" />
            </td></tr>
         <tr><td /><td><input type = "button" value = "Roll Dice"
           onclick = "play()" /></td></tr>
         </table>
         <div id = "status" class = "red">
            Click the Roll Dice button to play</div>
         <div id = "chatter">Let's play!</div>
      </form>
   </body>
</html>
```

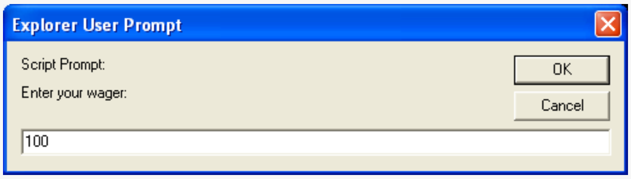


Wager:
Balance:

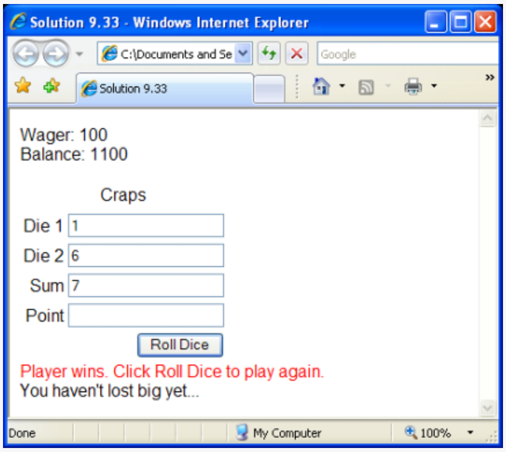
Craps

Die 1
Die 2
Sum
Point
Roll Dice

Click the Roll Dice button to play
Let's play!



Script Prompt:
Enter your wager:
100

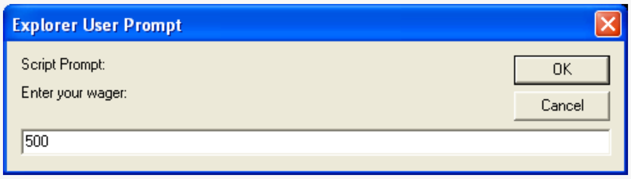


Wager: 100
Balance: 1100

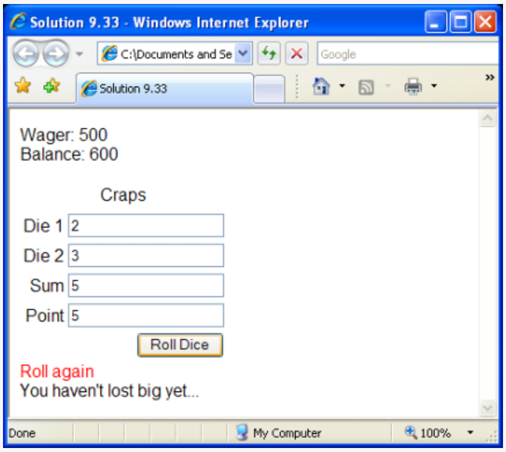
Craps

Die 1 1
Die 2 6
Sum 7
Point

Player wins. Click Roll Dice to play again.
You haven't lost big yet...



Script Prompt:
Enter your wager:
500

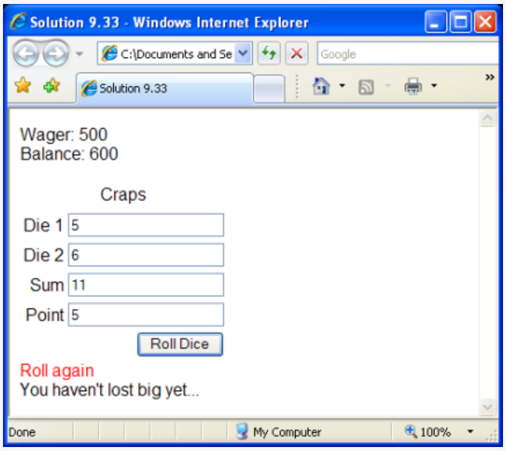


Wager: 500
Balance: 600

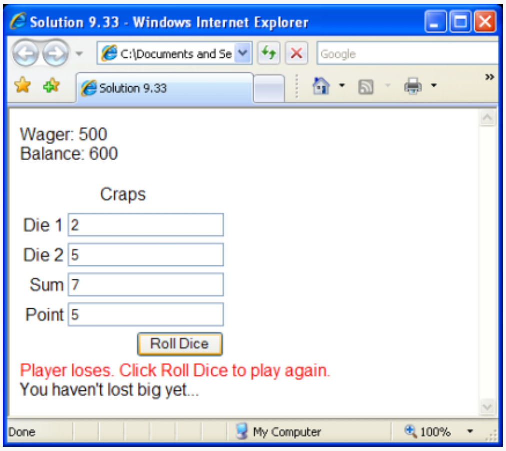
Craps

Die 1 2
Die 2 3
Sum 5
Point 5

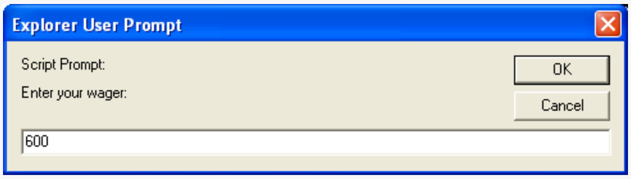
Roll again
You haven't lost big yet...



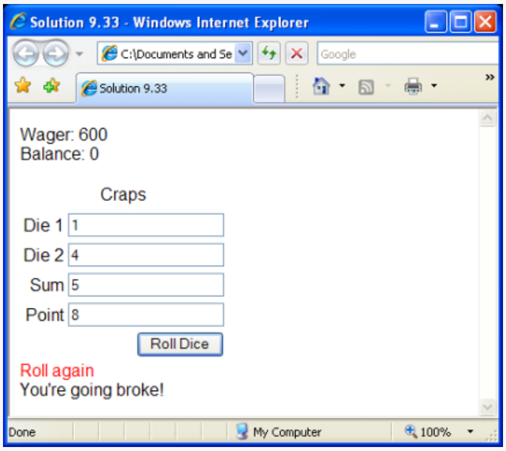
```
Wager: 500
Balance: 600

Craps
Die 1  5
Die 2  6
Sum    11
Point  5

Roll again
You haven't lost big yet...
```



```
Wager: 500
Balance: 600

Craps
Die 1  2
Die 2  5
Sum    7
Point  5

Player loses. Click Roll Dice to play again.
You haven't lost big yet...
```



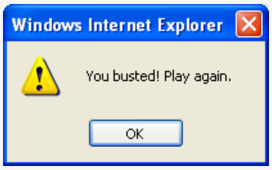
```
Script Prompt:
Enter your wager:
600
```


Wager: 600
Balance: 0

Craps

Die 1 1
Die 2 4
Sum 5
Point 8

Roll again
You're going broke!


You busted! Play again.

9.34 Write a recursive function `power( base, exponent )` that, when invoked, returns

$$base^{exponent}$$

for example, `power( 3, 4 ) = 3 * 3 * 3 * 3`. Assume that `exponent` is an integer greater than or equal to 0. The recursion step would use the relationship

$$base^{exponent} = base \cdot base^{exponent - 1}$$

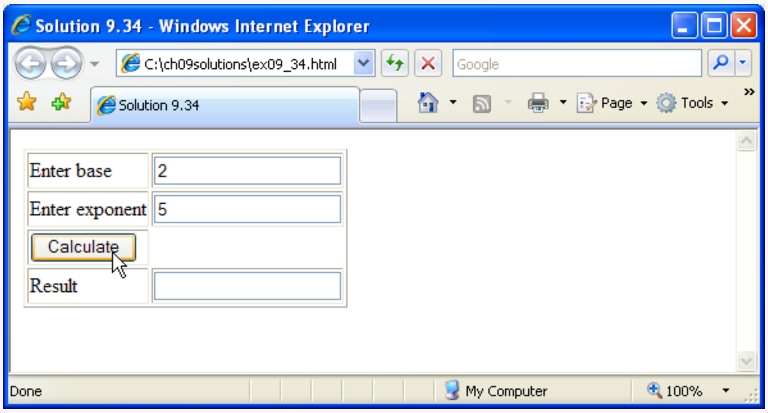
and the terminating condition occurs when `exponent` is equal to 0, because

$$base^{0} = 1$$

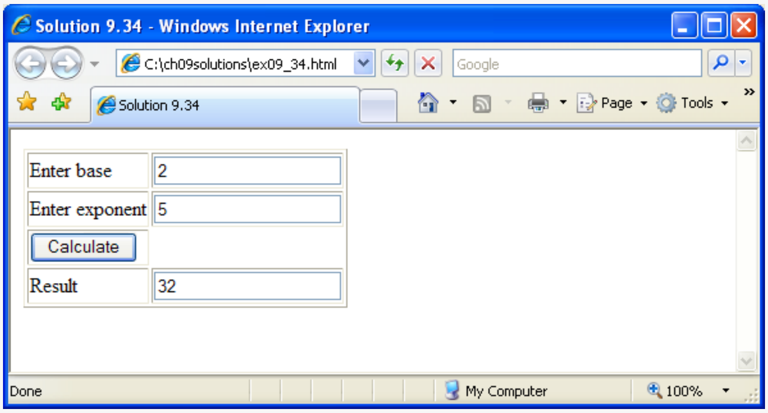
Incorporate this function into a script that enables the user to enter the `base` and `exponent`.

ANS:

```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Exercise 9.34: Solution -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Solution 9.34</title>
      <script type = "text/javascript">
         <!--
         function getPower()
         {
            var form = document.getElementById( "myForm" );
            var base = parseInt( form.base.value );
            var exponent = parseInt(form.exponent.value );
            form.result.value = power( base, exponent );
         } // end function getPower

         function power( b, exp )
         {
            if ( exp == 0 )
               return 1;
            else
               return b * power( b, exp - 1 );
         } // end function power
         // -->
      </script>
   </head>

   <body>
      <form id = "myForm" action = "">
         <table border = "1">
            <tr><td>Enter base</td>
                <td><input name = "base" type = "text" />
                </td>
            </tr>
            <tr><td>Enter exponent</td>
                <td><input name = "exponent" type = "text" />
                </td>
            </tr>
            <tr><td><input type = "button" value = "Calculate"
               onclick = "getPower()" /></td>
            </tr>
            <tr><td>Result</td>
                <td><input name = "result" type = "text" />
                </td>
            </tr>
         </table>
      </form>
   </body>
</html>
```


Solution 9.34 - Windows Internet Explorer
C:\ch09solutions\ex09_34.html
Google
Solution 9.34
Page
Tools
Done
My Computer
100%

Enter base	2
Enter exponent	5
Calculate	
Result	

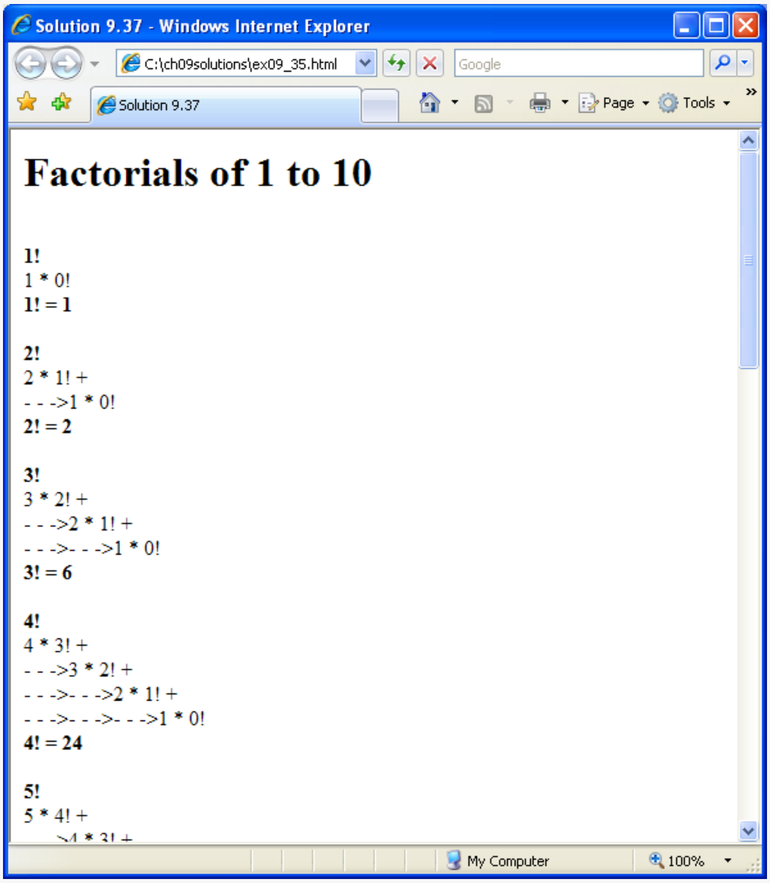

Solution 9.34 - Windows Internet Explorer
C:\ch09solutions\ex09_34.html
Google
Solution 9.34
Page
Tools
Done
My Computer
100%

Enter base	2
Enter exponent	5
Calculate	
Result	32

9.35 *(Visualizing Recursion)* It is interesting to watch recursion in action. Modify the factorial function in Fig. 9.11 to display its local variable and recursive-call parameter. For each recursive call, display the outputs on a separate line and add a level of indentation. Do your utmost to make the outputs clear, interesting and meaningful. Your goal here is to design and implement an output format that helps a person understand recursion better. You may want to add such display capabilities to the many other recursion examples and exercises throughout the text.

ANS:

```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Exercise 9.35: Solution -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Solution 9.35</title>
      <script type = "text/javascript">
         <!--
         var indent = "";

         document.writeln( "<h1>Factorials of 1 to 10</h1>" );

         for ( var i = 1; i <= 10; i++ )
         {
            document.writeln( "<br /><strong>" + i +
               "!</strong><br />" );
            var fact = factorial(i);
            document.writeln( "<strong>" + i + "! = " +
               parseInt(fact) + "</strong><br />" );
            indent = "";
         } // end for
         //recursive definition of function factorial
         function factorial( number )
         {
            if ( number <= 1 )  // base case
            {
               document.writeln( indent + parseInt(number) +
                  " * " + parseInt( number - 1 ) +
                  "! <br />" );
               return "1";
            } // end if
            else
            {
               document.writeln( indent + parseInt(number) +
                  " * " + parseInt( number - 1 ) +
                  "! + <br />" );
               indent += "- - ->";
               return number * factorial( number - 1 );
            } // end else
         } // end function factorial
      //-->
      </script>
   </head>
   <body></body>
</html>
```

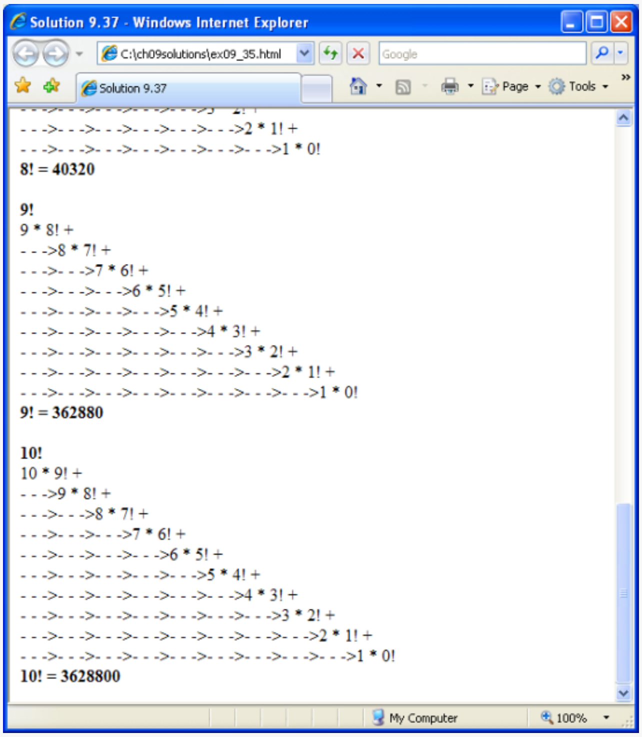


Solution 9.37 - Windows Internet Explorer

C:\ch09solutions\ex09_35.html

Factorials of 1 to 10

```
1!
1 * 0!
1! = 1

2!
2 * 1! +
- - ->1 * 0!
2! = 2

3!
3 * 2! +
- - ->2 * 1! +
- - ->- - ->1 * 0!
3! = 6

4!
4 * 3! +
- - ->3 * 2! +
- - ->- - ->2 * 1! +
- - ->- - ->- - ->1 * 0!
4! = 24

5!
5 * 4! +
```



```
- - ->- - ->- - ->- - ->- - ->- - ->2 * 1! +
- - ->- - ->- - ->- - ->- - ->- - ->- - ->1 * 0!
8! = 40320

9!
9 * 8! +
- - ->8 * 7! +
- - ->- - ->7 * 6! +
- - ->- - ->- - ->6 * 5! +
- - ->- - ->- - ->- - ->5 * 4! +
- - ->- - ->- - ->- - ->- - ->4 * 3! +
- - ->- - ->- - ->- - ->- - ->- - ->3 * 2! +
- - ->- - ->- - ->- - ->- - ->- - ->- - ->2 * 1! +
- - ->- - ->- - ->- - ->- - ->- - ->- - ->- - ->1 * 0!
9! = 362880

10!
10 * 9! +
- - ->9 * 8! +
- - ->- - ->8 * 7! +
- - ->- - ->- - ->7 * 6! +
- - ->- - ->- - ->- - ->6 * 5! +
- - ->- - ->- - ->- - ->- - ->5 * 4! +
- - ->- - ->- - ->- - ->- - ->- - ->4 * 3! +
- - ->- - ->- - ->- - ->- - ->- - ->- - ->3 * 2! +
- - ->- - ->- - ->- - ->- - ->- - ->- - ->- - ->2 * 1! +
- - ->- - ->- - ->- - ->- - ->- - ->- - ->- - ->- - ->1 * 0!
10! = 3628800
```

Solution 9.37 - Windows Internet Explorer
C:\ch09solutions\ex09_35.html
Google
Solution 9.37
Page
Tools
My Computer
100%

9.36 What does the following function do?

```
// Parameter b must be a positive
// integer to prevent infinite recursion

function mystery( a, b )
{
   if ( b == 1 )
      return a;
   else
      return a + mystery( a, b - 1 );
}
```

ANS: The function performs the multiplication of `a` and `b`, where `b` must be greater than or equal to 1.

9.37 Find the error in the following recursive function, and explain how to correct it:

```
function sum( n )
{
   if ( n == 0 )
      return 0;
   else
      return n + sum( n );
}
```

ANS: The value `n` does not decrement toward the `( n == 0 )` base case, creating an infinite loop. Fix this problem by calling `return n + sum( n - 1 )` instead.