

# CGS 3175: Internet Applications Fall 2007

## Introduction To JavaScript – Part 4

Instructor : Dr. Mark Llewellyn  
markl@cs.ucf.edu  
HEC 236, 407-823-2790  
<http://www.cs.ucf.edu/courses/cgs3175/fall2007>

School of Electrical Engineering and Computer Science  
University of Central Florida



## Things to Try Yourself

26. Modify the example XHTML document on page 20 so that it uses a function to print Tiffany's name.



## Things to Try Yourself - #26 Solution

```
<?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">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title> Writing a JavaScript function - Practice Problem #26 </title>
<script type="text/javascript">
    function writeTiffany() {
        document.write("Tiffany");
    }
</script>
</head>
<body>
<h1> Thanks for using JavaScript  </h1>
<h2>
<script type="text/javascript">
/* <![CDATA[ */
    document.write("<br /> Welcome ");
    writeTiffany();
    document.write(" !!<br />");
/* ]]> */
</script>
</h2>
</body>
</html>
```



# More On JavaScript Functions

- The functions we've seen so far have been functions which required no parameters. In other words, we did not need to send any information to the function in order for the function to accomplish its task. (Recall that when a function is invoked (called) it simply performs the task it was designed to accomplish.)
- In Practice Problem #26 the function printed Tiffany's name. Suppose that we wanted the function to be able to print any visitor's name. To accomplish this, we would need to ask the user to enter their name and then send their name as a **parameter** to the function.
- Just like the variables we've already seen in JavaScript, a parameter is also a variable, but this variable belongs only to the function in which it is defined.
- Let's rewrite Practice Problem #26 using a function with a parameter in which we will send the function the visitor's name.

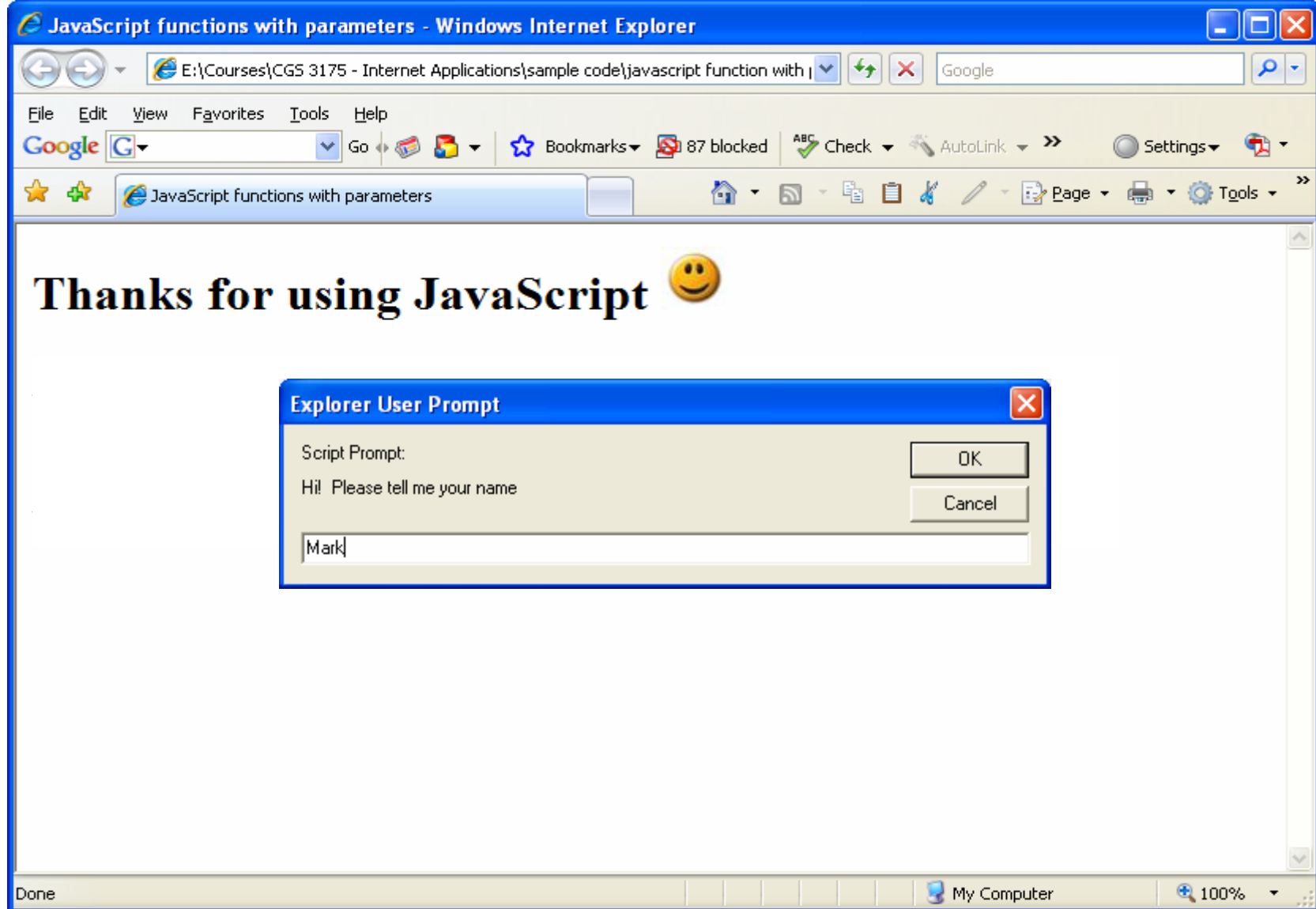


# More On JavaScript Functions

```
<?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">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title> JavaScript functions with parameters </title>
<script type="text/javascript">
  function writeVisitorName(name) {
    document.write(" " + name + " ");
  }
</script>
</head>
<body>
<h1> Thanks for using JavaScript  </h1>
<h2>
<script type="text/javascript">
/*  <![CDATA[  */
  var userName;
  userName = prompt("Hi!  Please tell me your name");
  document.write("<br /> Welcome ");
    writeVisitorName(userName);
  document.write(" !!<br /><br /> <br />");
  document.write("Welcome to our Web site...We hope you enjoy your stay ");
  writeVisitorName(userName);
  document.write("!<br />");
/*  ]]>  */
</script>
</h2>
</body>
</html>
```



# More On JavaScript Functions



# More On JavaScript Functions



## Creating A Slide Show Using JavaScript

- To create a slide show on the Web you preload a set of images, which are then played on demand as the visitor clicks forward and backward buttons.
- We'll add captions to our slides and introduce some more JavaScript features in this example.





# Creating A Slide Show Using JavaScript

```
<?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">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>My First Slide Show - Softball Images 2006-2007 </title>
<style type="text/css">
<!-- body { margin: 0px 10% 0px 10% }
-->
</style>
<script type="text/javascript">
/* The image objects are stored in an array.*/
var mySlides = new Array();
mySlides[1] = new Image();
mySlides[2] = new Image();
mySlides[3] = new Image();
mySlides[4] = new Image();
mySlides[5] = new Image();
mySlides[6] = new Image();
mySlides[7] = new Image();
mySlides[8] = new Image();

mySlides[1].src = "Pan Am Gold.jpg";
mySlides[2].src = "Finch 1.jpg";
mySlides[3].src = "Mendoza 1.jpg";
mySlides[4].src = "Cat 1.jpg";
mySlides[5].src = "Caitlin Lowe.jpg";
mySlides[6].src = "Mowatt 1.jpg";
mySlides[7].src = "Lovie Jung 1.jpg";
mySlides[8].src = "three great pitchers.jpg";
```



# Creating A Slide Show Using JavaScript

```
/* The text captions are stored in another array.*/
var myCaptions = new Array();
myCaptions[1] = "Team USA - Pan Am Gold Medalists.";
myCaptions[2] = "Jennie Finch - Team USA pitcher.";
myCaptions[3] = "Jessica Mendosa - Team USA - right fielder.";
myCaptions[4] = "Cat Osterman - Team USA pitcher.";
myCaptions[5] = "Caitlin Lowe - Team USA and Arizona Wildcat.";
myCaptions[6] = "Taryne Mowatt - University of Arizona Wildcats - pitcher.";
myCaptions[6] += "    ESPN Female Athlete of the Year 2007.";
myCaptions[7] = "Lovie Jung - Team USA - Sliding int 2nd base during 2007 World Cup.";
myCaptions[8] = "Three great pitchers: Jennie Finch, Alica Hollowell, and Cat Osterman.";

var slidenumber = 1;
var totalslides = mySlides.length - 1;

function showSlide(direction){
    if (direction == "next"){
        (slidenumber == totalslides) ? slidenumber = 1 : slidenumber++;
    }else{
        (slidenumber == 1) ? slidenumber = totalslides : slidenumber--;
    }
    if (document.images){
        document.slideframe.src = mySlides[slidenumber].src;
        document.slidecontrols.caption.value = myCaptions[slidenumber];
        document.slidecontrols.currentslide.value = slidenumber;
    }
}
</script>
</head>
```



# Creating A Slide Show Using JavaScript

```
<body style="background-color:white">
<div>
<h2>Softball Images 2006-2007</h2>


<!-- NOTE: This XHTML document will not validate under Strict DTD because the name
attribute of the form
        element has been deprecated, however, current browser support suggests to
ignore this validation
        error as the form may not be scriptable under some browsers without a name
attribute.
-->
<form id="slidecontrols" name="slidecontrols" action="#" method = "post">
<p><textarea id="caption" rows="2" cols="50">Our slide show begins with Team USA - Pan Am
Gold Medalists</textarea><br />
<br />
<input type="button" id="previous" name="previous" value="Previous Slide"
onclick="showSlide('previous');" />
<input type="button" id="next" name="next" value="Next Slide" onclick="showSlide('next');"
/>
<br /> <br />
Slide Number:
<input type="text" value="1" id="currentslide" name="currentslide" size="4" /></p>
</form>
</div>
</body>
</html>
```




My First Slide Show - Softball Images 2006-2007 - Mozilla Firefox

File Edit View History Bookmarks Tools Help

file:///E:/Courses/CGS%203175%20... Google

Customize Links Free Hotmail Windows Marketplace Windows Media Windows

## Softball Images 2006-2007



Our slide show begins with Team USA - Pan Am Gold Medalists

Previous Slide Next Slide

Slide Number:

Done



My First Slide Show - Softball Images 2006-2007 - Mozilla Firefox

File Edit View History Bookmarks Tools Help

file:///E:/Courses/CGS%203175%20... Google

Customize Links Free Hotmail Windows Marketplace Windows Media Windows

## Softball Images 2006-2007



Jennie Finch - Team USA pitcher.

Previous Slide Next Slide

Slide Number:

Done



My First Slide Show - Softball Images 2006-2007 - Mozilla Firefox

File Edit View History Bookmarks Tools Help

file:///E:/Courses/CGS%203175%2C Google

Customize Links Free Hotmail Windows Marketplace Windows Media Windows

## Softball Images 2006-2007



Taryne Mowatt - University of Arizona Wildcats - pitcher. ESPN Female Athlete of the Year 2007.

Previous Slide Next Slide

Slide Number:

Done



# How The Slide Show Script Works

- The browser loads the `<head>` and stores eight new image objects in an array called `mySlides` beginning with array element 1 (we are not using array element 0, but it is there). The `src` property of each image is then filled with a jpeg image file. After this step all of the images are preloaded onto the visitors computer.
- The next thing that happens is all of the image captions are loaded into an array called `myCaptions`, again beginning with array position 1 and corresponding to the correct images in the `mySlides` array. Notice the use of the add-by-value operator (`+=`) to store the long caption for the sixth image.
- A global variable, `slidenum` is created to hold the number of the current slide. A second global variable, `totalslides`, is created to hold the number of slides in the presentation. Since array element 0 is not used, we need to subtract 1 from the length of the array to determine the number of slides in the presentation.
- The function `showSlide(direction)` is stored in memory and will perform its function when called later.



# How The Slide Show Script Works

- The <body> fo the document includes an image called `slideframe` and a from called `slidecontrols`. Wihtin the form is a textarea called `caption` set to display 2 rows and 50 characters (columns) of text. An initial text string informs the visitor that the slide show begins with a picture of the 2007 Pan Am gold medal team from the USA.
- The form includes two buttons, the first button calls the `showSlide()` function and sends it the value `previous`. The second button calls the `showSlide()` function and sends it the value `next`.
- The `showSlide()` function in the <head> takes the value of the parameter it receives (either `previous` or `next`) and places it in a local (temporary) variable named `direction`. If `direction` has a value of `next`, the function looks to see whether the current slide number (stored in the global variable `slidenum`) is the last slide in the show. If it is, `slidenum` is reset to 1, Otherwise, `slidenum` is incremented by 1. If `direction` is `previous`, the function checks to see if the current slide number is the first slide in the show. If it is, `slidenum` is set to the total number of slides. Otherwise, `slidenum` is decremented by 1.

```
if (direction == "next"){
    (slidenum == totalslides) ? slidenum = 1 : slidenum++;
}else{
    (slidenum == 1) ? slidenum = totalslides : slidenum--;
}
```





## How The Slide Show Script Works

- Finally, the `showSlide()` function checks to make sure that the browser supports the `images` array. It then uses the current value of `slidenum` to determine which element of the `mySlides` array and corresponding element of the `myCaptions` array to display. The `slidenum` is also displayed in the text field called `currentslide` so that the visitor will see the number of the current slide.
- This slide show technique is useful for building pages where you want the visitor to proceed through a series of images by clicking buttons. Corporate presentations, travel logs, instructional demonstrations are just some of the possible applications.



# XHTML Tables

- Tables are used to format content into rows and columns. In the past they have been used to format forms, but as you have learned in this class, using the CSS box model and page layout techniques are the better way to perform this task. However, sometimes you just need to display a table within a page and XHTML tables are the easiest way to do this.
- In this last section of notes, we'll examine the table elements in XHTML.
- Tables in XHTML work much the same way they do in a spreadsheet or word processor application and resemble a matrix (grid).
- The entire table in XHTML is surrounded by the start table `<table>` tag and the end table tag `</table>`. You choose how many rows and columns you need for your table.
- The `<caption>` element is an optional element, and when present, must come directly after the `<table>` tag. There can only be one caption element per table. The `<caption>` element is used to describe the data in the table.



# XHTML Tables

- The **table heading** element `<th>` is used to label the rows and columns of a table. This element is an optional element for tables, as some tables do not require headings. In most browsers the `<th>` element automatically centers the content of the element within the table's cell and makes the text bold.
- The main body of a table is made up of rows and columns, like a matrix. In XHTML, you construct your tables one row at a time. Each row begins with an opening **table row** `<tr>` tag. Each column within that row contains the open and end tags for the **table data** `<td>` element. After the last `<td>` element is closed for the last column, the row ends with the ending `</tr>` tag. Shown below is the definition for a row that contains three columns.

```
<tr>  
    <td> column 1 </td>  
    <td> column 2 </td>  
    <td> column 3 </td>  
</tr>
```

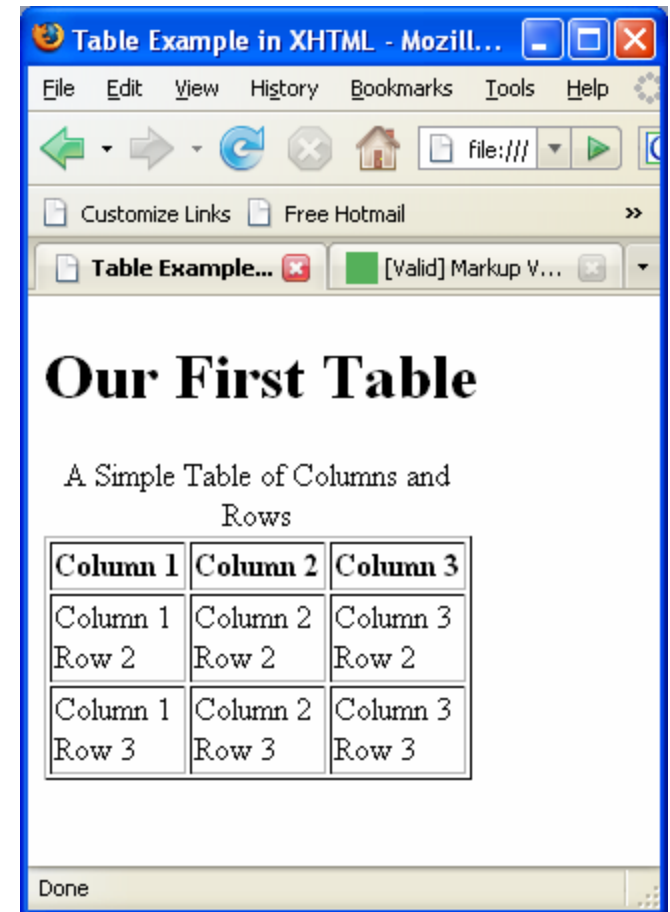


```

<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Table Example in XHTML</title>
</head>
<body>
<div style="align:center"><h1>Our First Table</h1></div>
<!-- Begin Table -->
<table border="1">
  <caption>A Simple Table of Columns and Rows</caption>
  <!-- Begin First Row -->
    <tr>
      <th>Column 1</th>
      <th>Column 2</th>
      <th>Column 3</th>
    </tr>
  <!-- End First Row -->
  <!-- Begin Second Row -->
    <tr>
      <td>Column 1 <br /> Row 2</td>
      <td>Column 2 <br /> Row 2</td>
      <td>Column 3 <br /> Row 2</td>
    </tr>
  <!-- End Second Row -->
  <!-- Begin Third Row -->
    <tr>
      <td>Column 1 <br /> Row 3</td>
      <td>Column 2 <br /> Row 3</td>
      <td>Column 3 <br /> Row 3</td>
    </tr>
  <!-- End Third Row -->
</table>
<!-- End Table -->
</body>
</html>

```

# XHTML Table – Example 1



# Formatting Tables

- Labeling table sections with `<thead>`, `<tbody>`, and `<tfoot>`.
  - The `<thead>`, `<tbody>`, and `<tfoot>` can be used to define logical sections of a table. These elements are used to group the various rows in a table into a header (`<thead>`), body (`<tbody>`), and footer (`<tfoot>`) section. While not heavily used in practice today these elements will become more important as new user agents become more dependent on document structure, so you want to be sure to use them when necessary.
  - These elements are optional, but when used must appear in the following order: `<thead>`, `<tfoot>`, `<tbody>`. The `<tfoot>` element must appear after the ending tag for the `</thead>` and before the open tag for the `<tbody>` element, even though its content will be displayed at the bottom of the table in a browser.
- The various table elements have a number of attributes that can be used to customize the look and layout of tables, rows, and cells. As with other elements, XHTML Strict does not allow all of the formatting attributes that Transitional and Frameset allow. The table on the next page illustrates the most common attributes that can be used with the `<table>`, `<tr>` and `<td>` elements.



# <table> Element Attributes

<u>Name</u>	<u>Description and Values</u>
<a href="#">summary</a>	Text description of the table. Useful for nonvisual browsers.
<a href="#">width</a>	Sets the width of the table. Values: Percentage or pixels
<a href="#">border</a>	Sets the width of the border around the table. Values: A value of 0 makes the border invisible. An integer value greater than 0 will result in a border of that number of pixels.
<a href="#">cellpadding</a>	Sets the amount of space between the border of the table cell and the data contained in the cell. Values: Percentage or pixels
<a href="#">cellspacing</a>	Sets the amount of space between cells. Values: Percentage or pixels
<a href="#">frame</a>	Defines which sides of the table will be displayed. Values: above, below, border, box, lhs, bsides, rhs, vsides, void
<a href="#">rules</a>	Defines which rule lines will be displayed. Values: all, cols, groups, none, rows



# <td> And <th> Element Attributes

<u>Name</u>	<u>Description and Values</u>
<a href="#">align</a>	Horizontal alignment of data in a cell Values: left, center, right, justified
<a href="#">valign</a>	Vertical alignment of data in a cell Values: top, middle, bottom
<a href="#">rowspan</a>	Number of rows a cell span Values: integer greater than 1 and less than or equal to the total number of rows in the table
<a href="#">colspan</a>	Number of columns a cell span Values: integer greater than 1 and less than or equal to the total number of columns in the table
<a href="#">abbr</a>	Used for an abbreviated version of the content of the cell
<a href="#">axis</a>	Used to assign a cell to a category group
<a href="#">headers</a>	List of cells that provide header information for the current cell based on the values of the id attributes of the header cells. This list is space delimited.
<a href="#">scope</a>	Provides information about which cells the current header cell provides header information for Values: col, colspan, row, rowspan



# <tr> Element Attributes

<u>Name</u>	<u>Description and Values</u>
<code>align</code>	Horizontal alignment of data in all cells in a row Values: left, center, right, justified
<code>valign</code>	Vertical alignment of data in all cells in a row Values: top, middle, bottom





## XHTML Table – Examples

- For our second table example, we'll have content that spans multiple rows and columns.
- The third table example illustrates a nested table (a table within a table).
- As with any skill, the best way to master the skill is to practice, practice, practice, so I encourage you to try and develop some additional tables on your own.



# XHTML Table – Example 2

```
<?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">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Spanning Multiple Table Rows and Columns with XHTML Tables</title>
  <style type="text/css">
    .teal_bg {
      background-color: #99FFFF }
    .yellow_bg {
      background-color: #FFFF00 }
    caption {
      font-weight: bold;
      font-size: 14pt;
      text-align: center;
      color: #000099;
    }
  </style>
</head>
<body>
<!-- Begin Table -->
<table border="1" cellpadding="5" class="teal_bg">
<caption>Saltwater Aquarium Invoice</caption>
  <thead>
    <!-- Begin Header Row -->
    <tr>
      <th rowspan="2">Item</th>
      <th colspan="2">Purchase Details</th>
      <th rowspan="2">Total Price</th>
    </tr>
    <tr>
      <th>Price</th>
      <th>Quantity</th>
    </tr>
  </thead>
</table>
```



```

</tr>
</thead>
<tfoot>
  <tr align="center">
    <td colspan="4"><small>Thank you for shopping with us.</small></td>
  </tr>
<!-- End First Row -->
</tfoot>
<tbody>
<!-- Begin First Item -->
  <tr>
    <th>Blue Angel Fish</th>
    <td align="center">$19.95</td>
    <td align="center">2</td>
    <td align="center">$39.90</td>
  </tr>
<!-- End First Item -->
<!-- Begin Second Item -->
  <tr>
    <th>Sailfin Tang Fish</th>
    <td align="center">$34.95</td>
    <td align="center">1</td>
    <td align="center">$34.95</td>
  </tr>
<!-- End Second Item -->
<!-- Begin Third Item -->
  <tr>
    <th>Clown Fish</th>
    <td align="center">$3.95</td>
    <td align="center">4</td>
    <td align="center">$15.80</td>
  </tr>

```



```

<!-- End Third Item -->
<!-- Begin Invoice Total -->
  <tr class="yellow_bg">
    <td colspan="3"><b>TOTAL</b></td>
    <td align="center"><b>$89.75</b></td>
  </tr>
<!-- End Invoice Total -->
</tbody>
</table>
</body>
</html>

```

The screenshot shows a web browser window with the title "Spanning Multiple Table Rows and Columns with XHTML ...". The browser's address bar shows "file:///E:/Cours". The main content of the page is a table titled "Saltwater Aquarium Invoice".

Item	Purchase Details		Total Price
	Price	Quantity	
Blue Angel Fish	\$19.95	2	\$39.90
Sailfin Tang Fish	\$34.95	1	\$34.95
Clown Fish	\$3.95	4	\$15.80
<b>TOTAL</b>			<b>\$89.75</b>

Below the table, there is a text box that says "Thank you for shopping with us." The browser's status bar at the bottom shows "Done".



## XHTML Table – Example 3

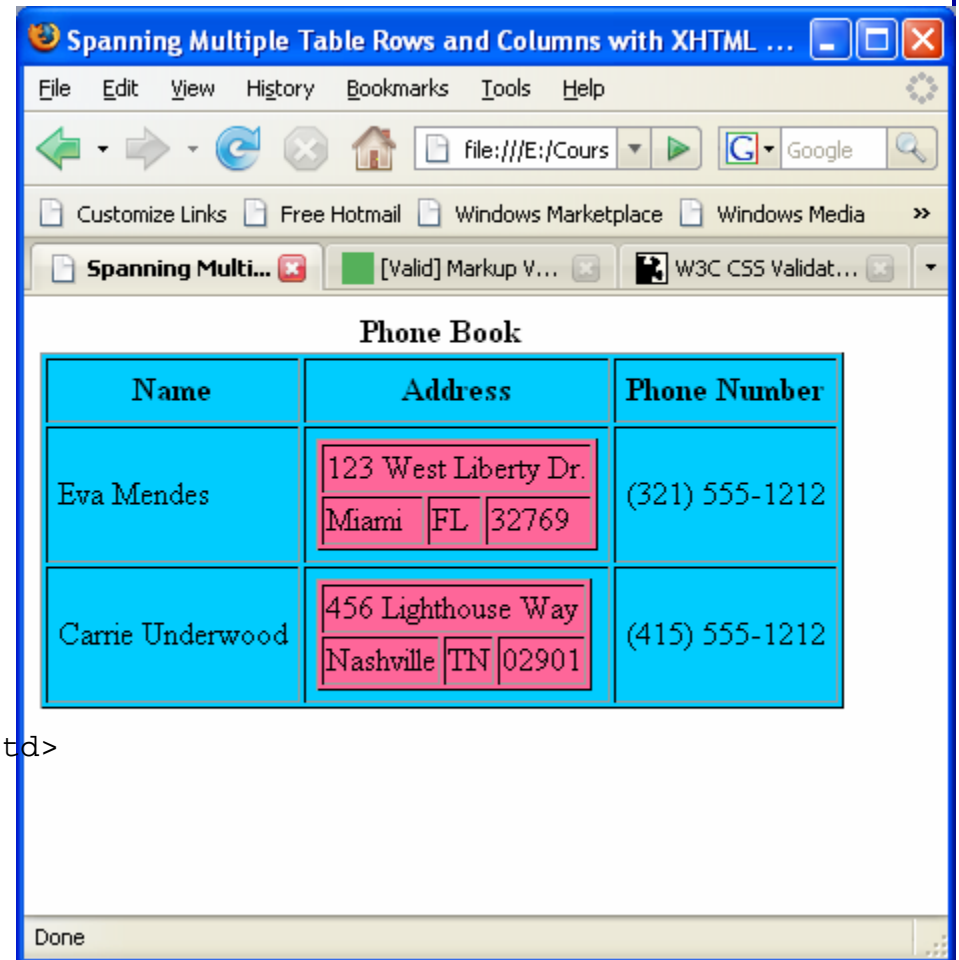
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Spanning Multiple Table Rows and Columns with XHTML Tables</title>
  <style type="text/css">
    .outer {
      background-color: #00CCFF;
    }
    .inner {
      background-color: #FF6699;
    }
  </style>
</head>
<body>
<!-- Begin Outer Table -->
<table border="1" cellpadding="5" class="outer">
  <caption><strong>Phone Book</strong></caption>
  <!-- Begin Header Row -->
  <tr>
    <th>Name</th>
    <th>Address</th>
    <th>Phone Number</th>
  </tr>
  <!-- End Header Row -->
  <!-- Begin First Row -->
  <tr>
    <td>Eva Mendes</td>
    <td>
      <!-- Begin Inner Table -->
      <table border="1" class="inner">
        <tr>
          <td colspan="3">123 West Liberty Dr.</td>
        </tr>
      </table>
    </td>
  </tr>
</table>
</body>
</html>
```



```

        </tr>
        <tr>
        <td>Miami</td>
        <td>FL</td>
        <td>32769</td>
        </tr>
    </table>
<!-- End Inner Table -->
</td>
<td>(321) 555-1212</td>
</tr>
<!-- End First Row -->
<!-- Begin Second Row -->
<tr>
<td>Carrie Underwood</td>
<td>
<!-- Begin Inner Table -->
    <table border="1" class="inner">
    <tr>
    <td colspan="3">456 Lighthouse Way</td>
    </tr>
    <tr>
    <td>Nashville</td>
    <td>TN</td>
    <td>02901</td>
    </tr>
    </table>
    <!-- End Inner Table -->
    </td>
    <td>(415) 555-1212</td>
    </tr>
<!-- End Second Row -->
</table></body></html>

```



## Things to Try Yourself

27. Modify the JavaScript example on page 5 that uses a function that returns a value and change the function (and the XHTML document) to convert Celsius temperatures to Fahrenheit temperatures. The conversion you will need is  $((9/5) * \text{temp}) + 32$ .



# Solution To Practice Problem #27

```
<?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">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title> JavaScript function that returns a value </title>
<script type="text/javascript">
  function fahrenheitToCelsius(tempInF) {
    return (Math.round((5/9)*(tempInF-32)));
  }
</script>
</head>
<body>
<h1><span style="color:blue"> Welcome to the CGS 3175 Temperature Converter</span>

</h1>
<h2>
<script type="text/javascript">
/* <![CDATA[ */
  var temperature;
  temperature = prompt("Please enter the temperature in degrees Fahrenheit.");
  document.write("<br /> <br />");
  document.write("A temperature of " + temperature + " degrees F is equivalent to ");
  document.write(fahrenheitToCelsius(temperature));
  document.write(" degrees C.<br />");
/* ]]> */
</script>
</h2>
</body>
</html>
```

Clip art file available  
on the Web.



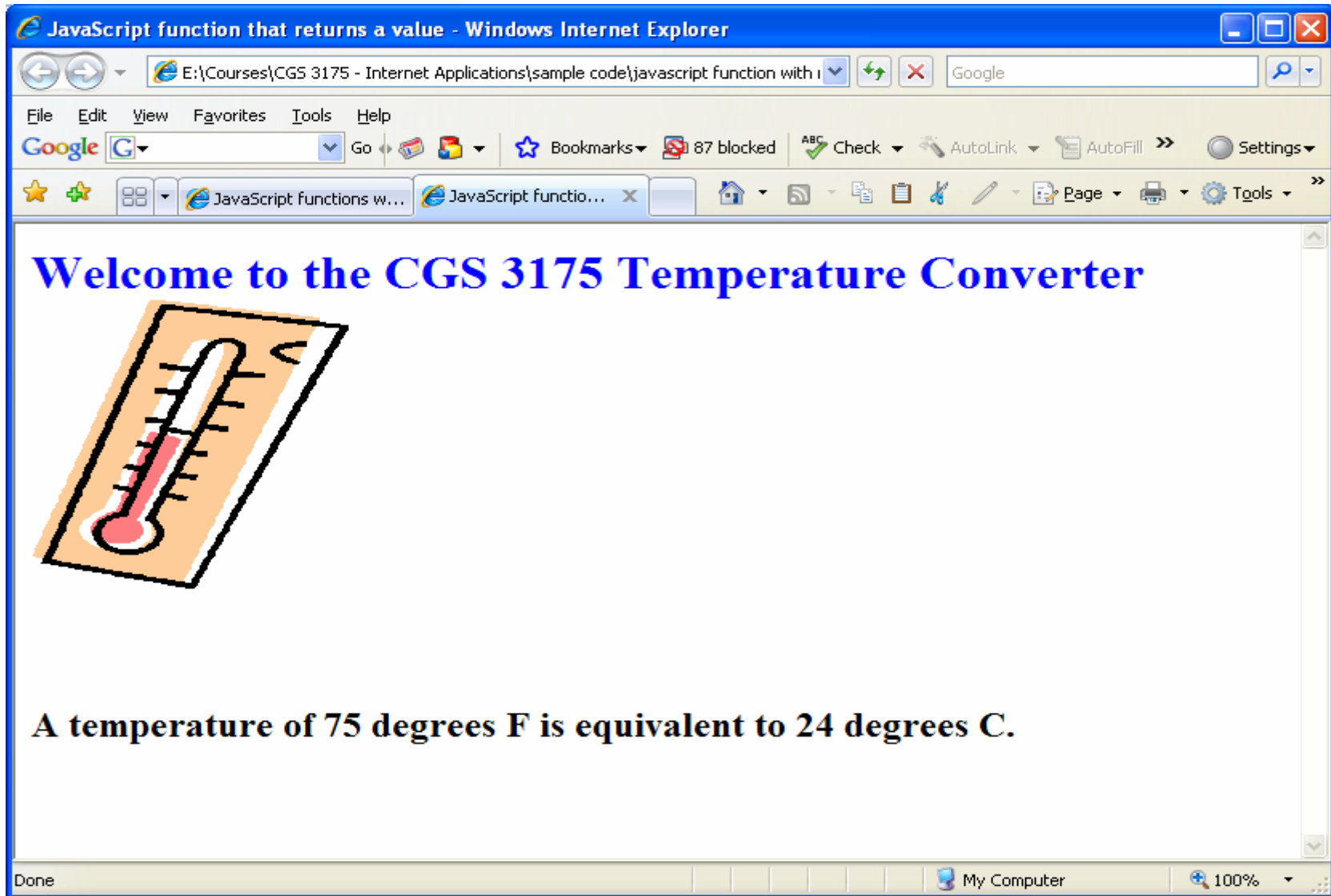


# Solution To Practice Problem #27

The screenshot shows a Windows Internet Explorer browser window titled "JavaScript function that returns a value - Windows Internet Explorer". The address bar displays the file path "E:\Courses\CGS 3175 - Internet Applications\sample code\javascript function with i". The browser's menu bar includes "File", "Edit", "View", "Favorites", "Tools", and "Help". The toolbar contains various icons for navigation and utility. The main content area displays the text "Welcome to the CGS 3175 Temperature Converter" in blue. To the left of the text is a graphic of a thermometer. An "Explorer User Prompt" dialog box is overlaid on the page, containing the text "Script Prompt: Please enter the temperature in degrees Fahrenheit." and a text input field with the value "75". The dialog box has "OK" and "Cancel" buttons. The browser's status bar at the bottom shows "Done", "My Computer", and "100%".




# Solution To Practice Problem #27



The screenshot shows a Windows Internet Explorer browser window. The title bar reads "JavaScript function that returns a value - Windows Internet Explorer". The address bar contains the file path "E:\Courses\CGS 3175 - Internet Applications\sample code\javascript function with i" and the search engine "Google". The menu bar includes "File", "Edit", "View", "Favorites", "Tools", and "Help". The toolbar shows "Go", "Bookmarks", "87 blocked", "Check", "AutoLink", "AutoFill", and "Settings". The main content area displays the text "Welcome to the CGS 3175 Temperature Converter" in blue, followed by a cartoon illustration of a thermometer. Below the illustration, the text reads "A temperature of 75 degrees F is equivalent to 24 degrees C." The status bar at the bottom shows "Done", "My Computer", and "100%".

**Welcome to the CGS 3175 Temperature Converter**



**A temperature of 75 degrees F is equivalent to 24 degrees C.**

