CGS 3175: Internet Applications
Fall 2007

Introduction To JavaScript – Part 3

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

25. Modify any of the example XHTML documents that illustrate the intrinsic events to try some of the intrinsic events that were not illustrated in the notes such as `onmouseup`.

Things to Try Yourself

```
<?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>Triggering Scripts - onmouseover  </title>
<style type="text/css">
<!--  body {background-color:#CCFFCC;  }
      div {border:1px solid black; width:300px; }
-->
</style>
</head>
<body>
<div>
<p onmouseover="alert('Today is '+ Date() ) ">Move mouse here for the current
time.</p>
</div>
<p>The rest of the page goes here...</p>
</body>
</html>
```

Writing Valid JavaScript Code

- Throughout the semester we have always validated our XHTML documents against the strict data type definition (Strict-DTD) to ensure that our XHTML documents were well-formed.
- Some JavaScript statements contain symbols such as the less-than symbol (<), the greater-than symbol (>), and the ampersand (&). As you become a more sophisticated JavaScript programmer, you will begin to use many of the features contained in the JavaScript language and will undoubtedly encounter the need to use these symbols. Unfortunately, these symbols can prevent XHTML documents from passing validation (particularly under the Strict-DTD).
  - Note that there is less of a problem with this when using the Transitional-DTD, but we do not want to relax our standards.
- This is not a problem at all when using HTML, because any statements inside a `<script>` element are interpreted as character data instead of markup.
  - A section of a document that is not interpreted as markup is referred to as character data, or CDATA.
- If you were to validate an HTML document that contained a `<script>` element, the document would validate successfully because the validator would ignore the script section and not attempt to interpret the text and symbols in the JavaScript statements as HTML or attributes.

Writing Valid JavaScript Code

- In contrast, with XHTML documents, the statements in a `<script>` element are treated as parsed character data, or PCDATA, which identifies a section of a document that is interpreted as markup.

- This means that if you attempt to validate an XHTML document that contains a `<script>` element, it may fail to validate.

  – Note that an XHTML document will not necessarily fail to validate under Strict-DTD just because it contains a `<script>` element. In fact, any of the examples that have appeared in the JavaScript notes thus far, will validate successfully. However, the right sequence of symbols inside the `<script>` element may cause the document not to validate.

Writing Valid JavaScript Code

- To avoid this potential problem, you can do one of two things.
- One option is to move all JavaScript code into an external file (with a .js extension) as we saw in Part 1 and will see in more detail later in this section of notes. This of course prevents the validator from attempting to parse the JavaScript statements.
- The second option, and will be a requirement for embedded JavaScript, is to enclose the JavaScript within a <script> element within a CDATA section.
- The syntax for a CDATA section of an XHTML document is as follows:

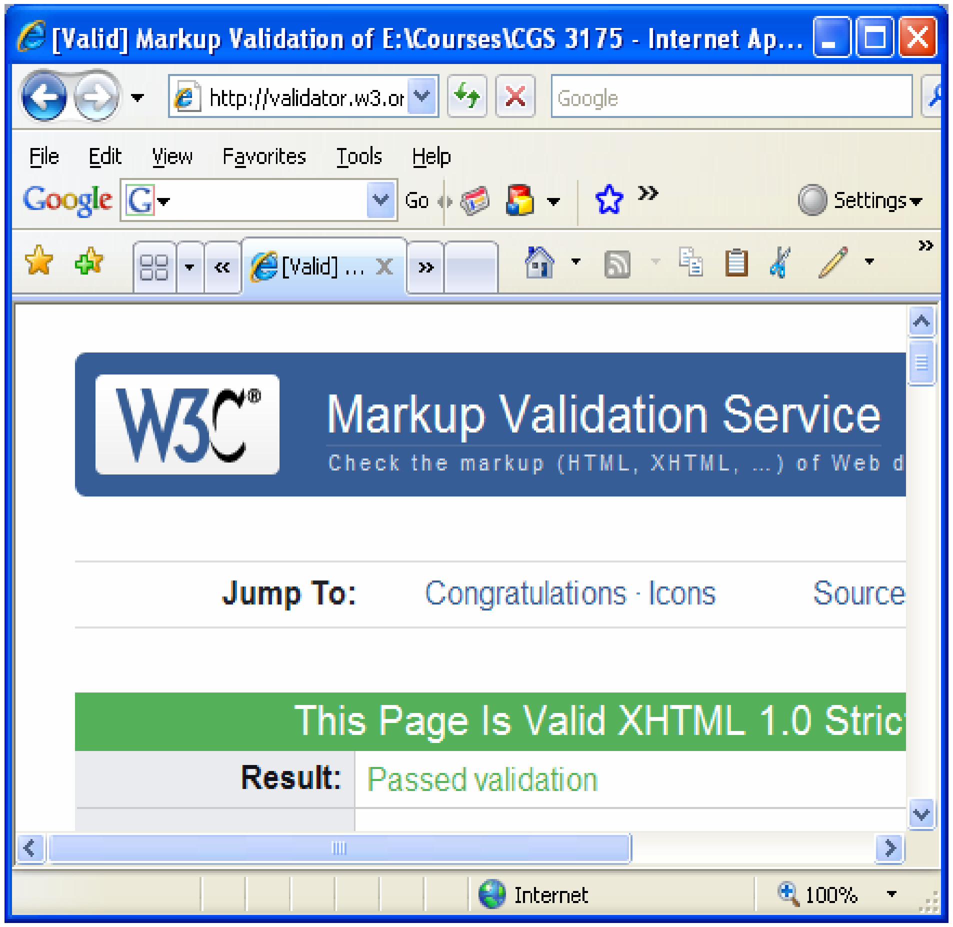
```
/*  <![CDATA [   */
  statements to mark as CDATA
/*  ] ] >   */
```

- Note that the block comments on the opening and closing portions of the CDATA section prevent the JavaScript interpreter from attempting to parse the `<![CDATA[` and `]]>` lines as JavaScript!
- The example on the following page illustrates a CDATA section in an XHTML document. From here on, for embedded JavaScript we'll use this format to ensure validation.

Writing Valid JavaScript Code

```
<?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>Valid JavaScript for compatibility </title>
</head>
<body>
<script type="text/javascript">
/*  <![CDATA[  */
   document.write("<br /> This page illustrates the proper way to embed a
JavaScript script into an <br />");
   document.write("XHTML document.  Since JavaScript statements can contain
symbols such as <, >, and & which <br />");
   document.write("used in XHTML and can cause the XHTML validator to
improperly interpret the JavaScript statements <br />");
   document.write("as markup.  For this reason, the JavaScript needs to be
interpreted as character data, so it is placed in <br />");
   document.write("CDATA section.<br />");
/* ]]>  */
</script>
</body>
</html>
```

Writing Valid JavaScript Code


[Valid] Markup Validation of E:\Courses\CGS 3175 - Internet Ap...
http://validator.w3.or
File Edit View Favorites Tools Help
Internet
100%

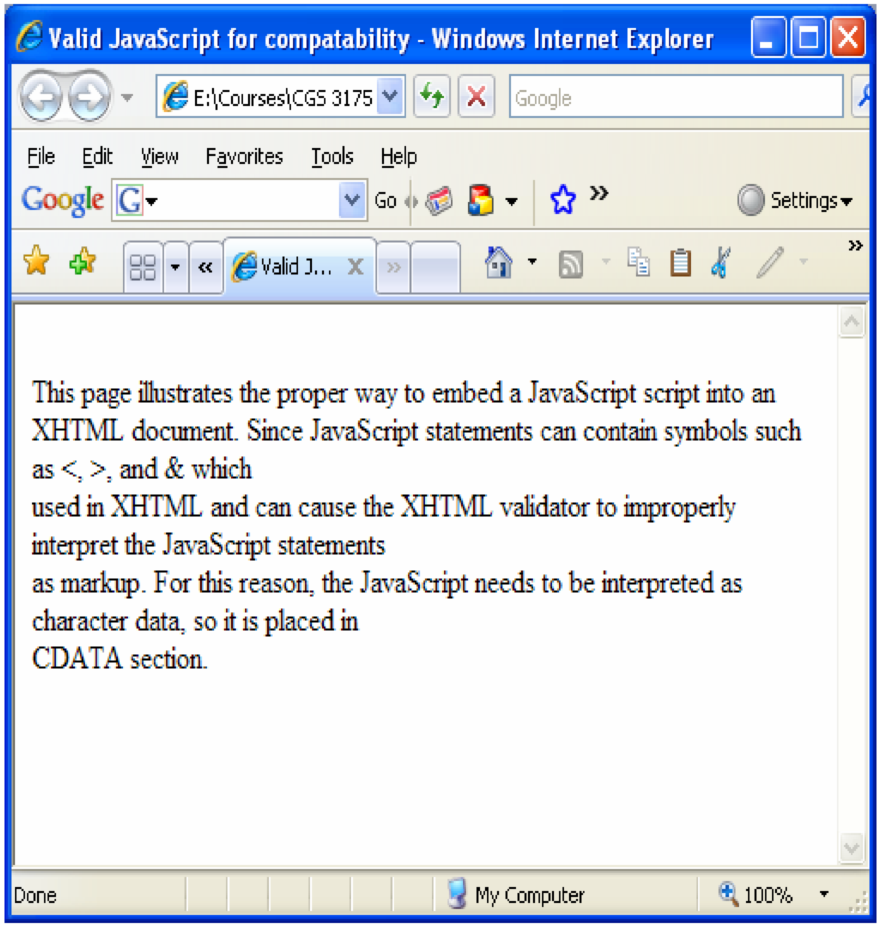
Markup Validation Service

Check the markup (HTML, XHTML, ...) of Web d

Jump To: Congratulations · Icons Source

This Page Is Valid XHTML 1.0 Stric

Result: Passed validation


Valid JavaScript for compatability - Windows Internet Explorer
E:\Courses\CGS 3175
File Edit View Favorites Tools Help
Done
My Computer
100%

This page illustrates the proper way to embed a JavaScript script into an XHTML document. Since JavaScript statements can contain symbols such as <, >, and & which
used in XHTML and can cause the XHTML validator to improperly interpret the JavaScript statements
as markup. For this reason, the JavaScript needs to be interpreted as character data, so it is placed in
CDATA section.

Creating A JavaScript Library

- As we saw in Part 1 of the JavaScript notes, it is quite common to create a library (a file) of JavaScript scripts which provides any of your Web pages access to the scripts without having to repeat the writing of the scripts in either the head or body sections of each document.
- Unless the JavaScript code you intend to use in a document is very short or specific to only one page, it is usually preferred to place the scripts in a library file for the following reasons:
  - Your document will be neater. Lengthy JavaScript code in a document can be confusing and makes understanding ("reading") and maintaining the XHTML that more difficult. You might not be able to tell at a glance where the XHTML code ends and the JavaScript code begins.
  - The JavaScript code can be shared among multiple Web pages. For example, an e-commerce site may contain several pages that allow a user to order an item. Each such page displays a different item but can use the same JavaScript code to gather order information. Instead of recreating the JavaScript order information code within each document, the various pages can share a central JavaScript source file. Sharing a single source file reduces the requirements for disk space and reduces system overhead since only one copy of the same code needs to be in memory.
  - JavaScript libraries hide JavaScript code from incompatible browsers. If your document contains JavaScript code, an incompatible browser displays that code as if it were standard text. In contrast, if the code is contained in a library, the incompatible browser simply ignores it.

Creating A JavaScript Library

- While JavaScript libraries are quite common, it is also quite common to see both libraries and embedded JavaScript code in Web documents, so you need to be familiar with both forms.

- Recall that the `<script>` tag can appear within the <head> tag and/or the `<body>` tag.

- As we will see shortly, the more common form of a script to be included in a library is a function. The following example illustrates the effect of using a JavaScript library without functions.

```
/* This is a JavaScript library of scripts  */
//SCRIPT #1
//this script writes the date and time onto a page
    document.write("</p align='right'> Today is: <i>" + Date() + "</i></p>");
    document.write("<br />");
//SCRIPT #2
//this script returns today's date
   var currentTime = new Date()
   var month = currentTime.getMonth() + 1
   var day = currentTime.getDate()
   var year = currentTime.getFullYear()
   document.write("Today is: " + month + "/" + day + "/" + year)
   document.write("<br />");
//SCRIPT #3
//this script returns the current time
    var currentTime = new Date()
    var hours = currentTime.getHours()
    var minutes = currentTime.getMinutes()
    if (minutes < 10)
        minutes = "0" + minutes
        document.write("The time is " + hours + ":" + minutes + " ")
    if(hours > 11){
        document.write("PM")
    } else {
        document.write("AM")
    }
    document.write("<br />");
//SCRIPT #4
//this script simply writes a message
   document.write(“<br /> Hello there!  I'm a JavaScript script executing on your behalf. <br />");
   document.write("Please note that since I am not a function and thus not called directly <br />");
   document.write("that all of the other scripts in my library have executed before me, since I <br />");
   document.write("appear last in the library <br />");
```

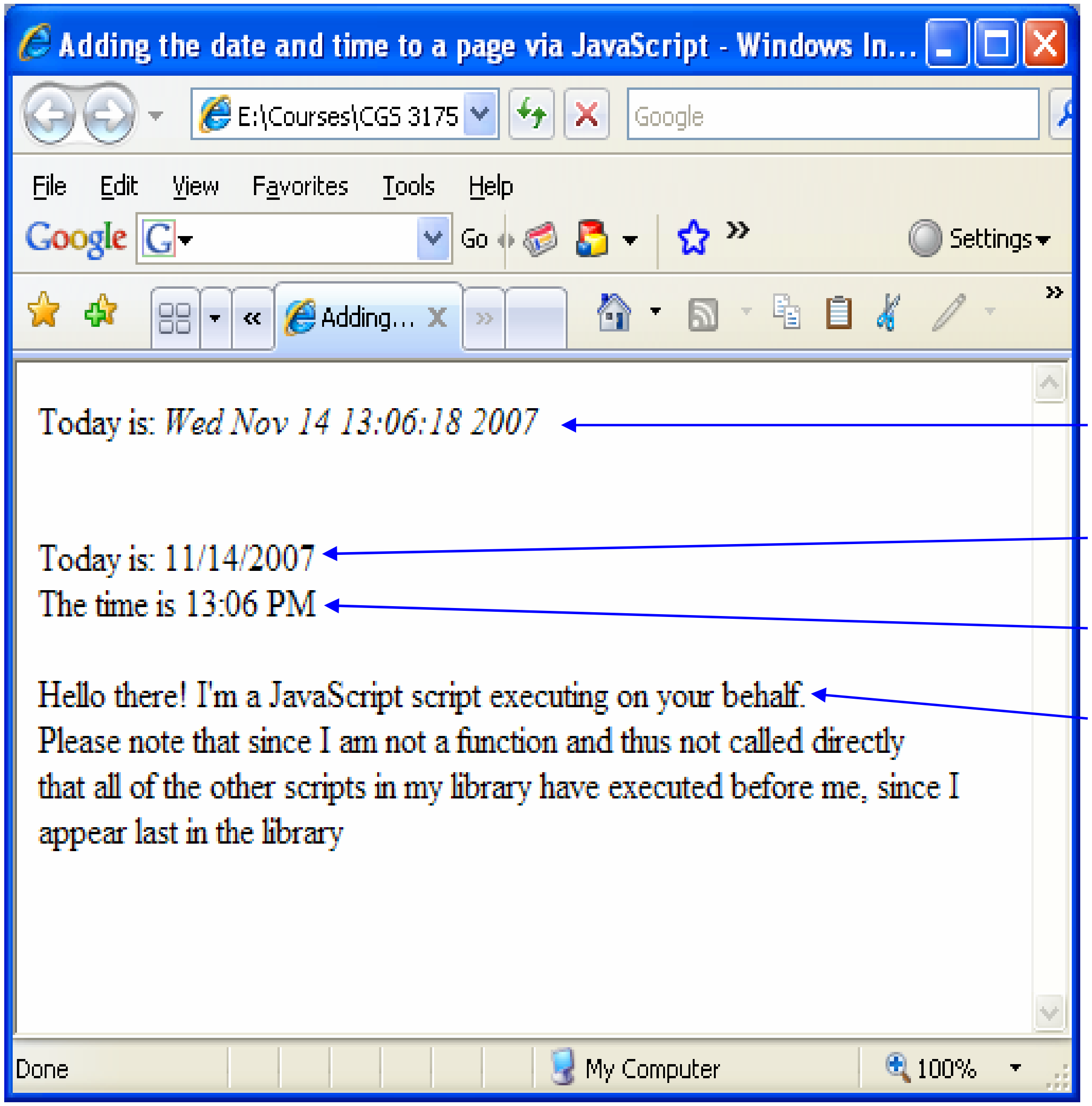
A JavaScript Library

Remember that a JavaScript library has a “.js” file extension

Execution Using A JavaScript Library

This sample XHTML document does nothing except load and run the scripts in the JavaScript library named `myscriptlibrary2.js`

```
<?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>Adding the date and time to a page via JavaScript  </title>
</head>
<body>
<script type="text/javascript"  src="myscriptlibrary2.js"></script>
</body>
</html>
```


Adding the date and time to a page via JavaScript - Windows In...
E:\Courses\CGS 3175
Google
File Edit View Favorites Tools Help
Go
Settings
Adding...
Done
My Computer
100%

Today is: *Wed Nov 14 13:06:18 2007*

Today is: 11/14/2007
The time is 13:06 PM

Hello there! I'm a JavaScript script executing on your behalf. Please note that since I am not a function and thus not called directly that all of the other scripts in my library have executed before me, since I appear last in the library

Execution Using A JavaScript Library

Remember that a JavaScript library has a ".js" file extension

Execution of SCRIPT #1

Execution of SCRIPT #2

Execution of SCRIPT #3

Execution of SCRIPT #4

We'll see shortly how to use functions to "call" and execute only one of these scripts or change the order of execution or whatever we want to do with them.

More JavaScript - Variables

- The values a program stores in computer memory are called variables. Technically speaking, a variable is actually a specific address in the computer memory. The data stored in a variable often changes.
  - Think of a variable like a book bag or back pack. You can put any book you want in the bag and retrieve it later for use. The books in your bag this semester will probably not be the same ones in your bag next semester.
- Many programming languages, such as Java and C++ have a very large set of rules that apply to variables. JavaScript is very loose in how variables can be used.

Naming Variables

- The name you assign to a variable is called an identifier. Although technically different, you can use the terms variable and identifier interchangeably.
- JavaScript defines the following rules for naming a variable:
  - Identifiers must begin with an uppercase or lowercase ASCII letter, dollar sign ($) or underscore(_). (Older browsers will not accept $.)
  - Numbers can appear in the identifier, but not as the first character.
  - No spaces are allowed in the identifier.
  - You cannot use any reserved word as an identifier (see next page.)
- Some examples:

  Valid identifiers: `Angela, num1, _newt, $amount, debi`

  Invalid identifiers: `Didi Thurau, 16_Nov, *69`
- Variable names should be descriptive not cryptic. Convention dictates that variable names begin with a lowercase letter and each additional word in the identifier begins with an uppercase letter. Some examples of conventional variable names are: `productDate, myLastName, birthDate,` and `myLastLapTime`.

Reserved Words In JavaScript

abstract	else	instanceof	switch
boolean	enum	int	synchronized
break	export	interface	this
byte	extends	long	throw
case	false	native	throws
catch	final	new	transient
char	finally	null	true
class	float	package	try
const	for	private	typeof
continue	function	protected	var
debugger	goto	public	void
default	if	return	volatile
delete	implements	short	while
do	import	static	with
double	in	super	

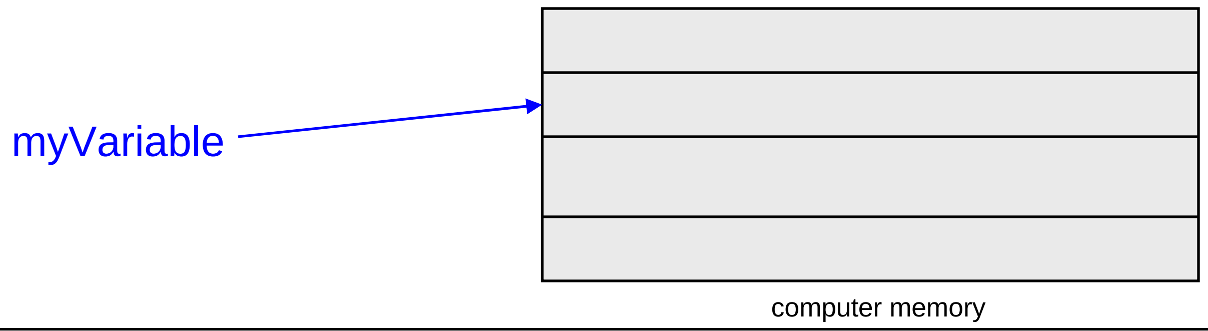
Using Variables

- Before you can use a variable, you need to declare it (basically it means create it). While there are different techniques to create variables, we'll stick with the most common and simplest form which uses the reserved word `var`.

- For example, to create a variable named `myVariable`, you need to write this statement:

```
var myVariable;
```

- All this statement does is make some memory location be accessible to your code whenever you refer to it by this name.



Using Variables

- Declaring a variable just sets aside memory for it, it does not assign any value to the memory.

- Often you want to give the memory location, hence the variable, some initial value. The shorthand notation for this occurs at the same time as the declaration as follows:

```
var myVariable = value;
```

- Examples:

```
var myName = “Mark”;  //literal string
var roomNumber = 104; //a number
var myNum = “69”;  //literal string
```

Writing A Variable's Value To A Web Page

- While some variables will be used only internally to the JavaScript, others will need to be sent to the Web page for display.

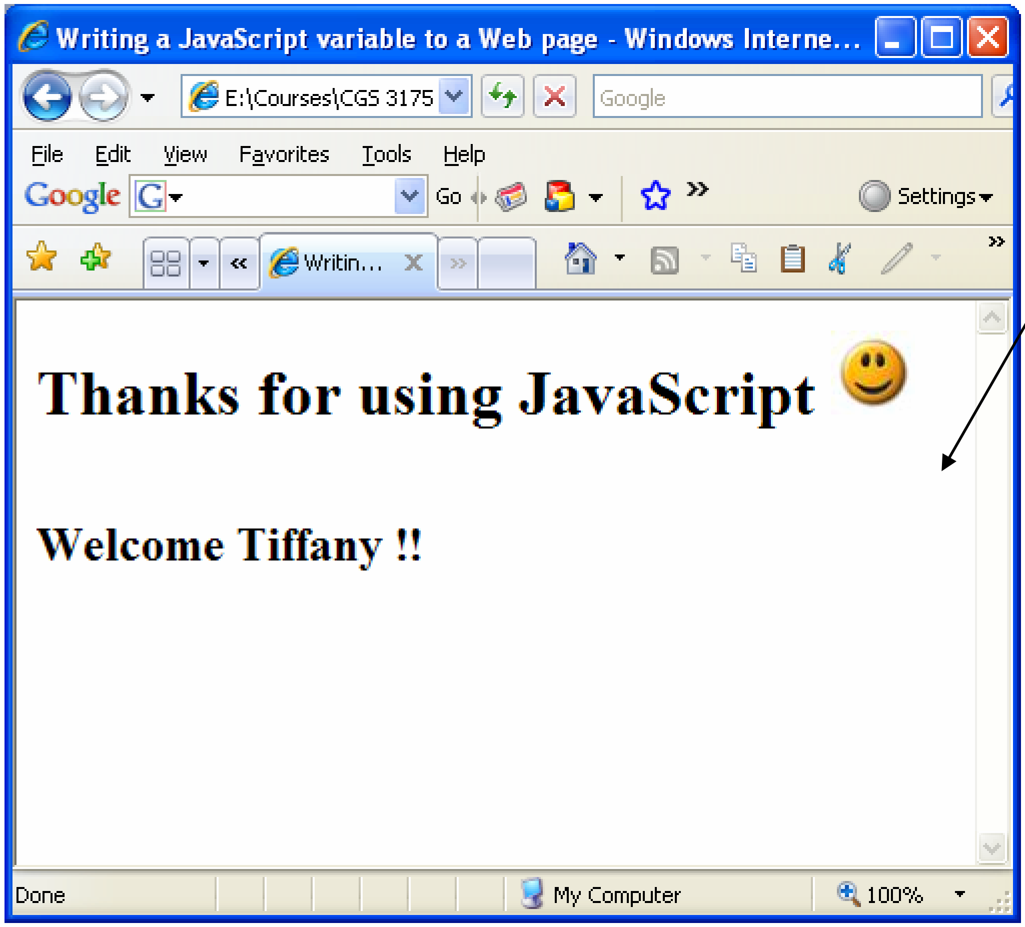
- This is quite simple in JavaScript and is basically no different than what we have already been doing with our simple scripts that have been printing text (strings) to the Web page.

- The next page illustrates a simple XHTML document with an embedded JavaScript that uses a variable which is sent to the Web page for rendering by the browser.

Writing A Variable’s Value To A Web Page - Example

```
<?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 variable to a Web page  </title>
</head>
<body>
<h1> Thanks for using JavaScript <img src="smiley1.jpg" alt="a smiley face"
/> </h1>
<h2>
<script type="text/javascript">
/*  <![CDATA[  */
        var userName = "Tiffany";
        document.write("<br /> Welcome ");
        document.write(userName);
        document.write(" !!<br />");
/* ]]>  */
</script>
</h2>
</body>
</html>
```

Writing A Variable's Value To A Web Page - Example


Writing a JavaScript variable to a Web page - Windows Interne...
E:\Courses\CGS 3175
Google
File Edit View Favorites Tools Help
Go
Settings
Writin...
Done
My Computer
100%

Thanks for using JavaScript

Welcome Tiffany !!

This works great if all of the visitors to your site are named Tiffany, otherwise it won't be too useful!

Assigning Variable Values Using A Prompt

- To make your Web page more interactive, you obviously need some way to receive values from the visitor (we've already done this to some extent using only XHTML with forms).
- In the previous example, things worked well if all the visitors to our Web page were named Tiffany. If your name happens to be Debi, the page doesn't really seem too personal!
- What we need to do is prompt the visitor to tell us their name, then we can assign that to a variable and use the value whenever it seems appropriate.
- The `prompt()` method is a method of the window object (just like the `alert()` method that we've already used in the intrinsic event examples to display the date and time). Normally the prompt() method is used in conjunction with a variable, so that the incoming data is stored in the variable.

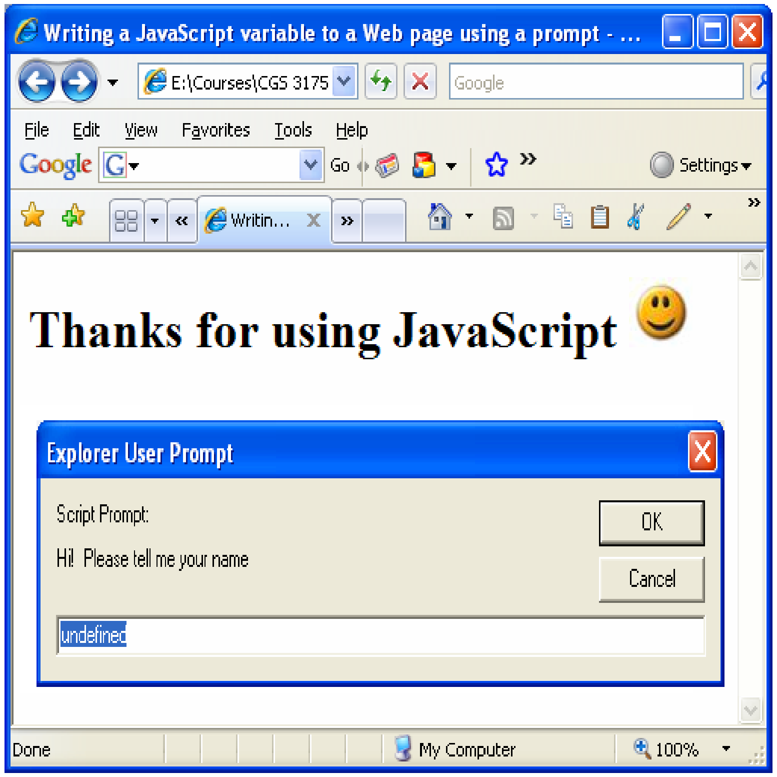
```
someVariable = prompt("prompt message");
```

- The following example develops a modified version of the previous example using a prompt.

Assigning Variable Values Using A Prompt - Example

```
<?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 variable to a Web page using a prompt  </title>
</head>
<body>
<h1> Thanks for using JavaScript <img src="smiley1.jpg" alt="a smiley face" />
</h1>
<h2>
<script type="text/javascript">
/*  <![CDATA[  */
        var userName = "";
        userName = prompt("Hi!  Please tell me your name");
        document.write("<br /> Welcome ");
        document.write(userName);
        document.write(" !!<br /><br /> <br />");
        document.write("Welcome to our Web site...We hope you enjoy your stay "
+ userName + "!<br />");
/* ]]>  */
</script>
</h2>
</body>
</html>
```

Example – Internet Explorer Version


Writing a JavaScript variable to a Web page using a prompt - ...

E:\Courses\CGS 3175

Thanks for using JavaScript

Explorer User Prompt

Script Prompt:

Hi! Please tell me your name

undefined

OK

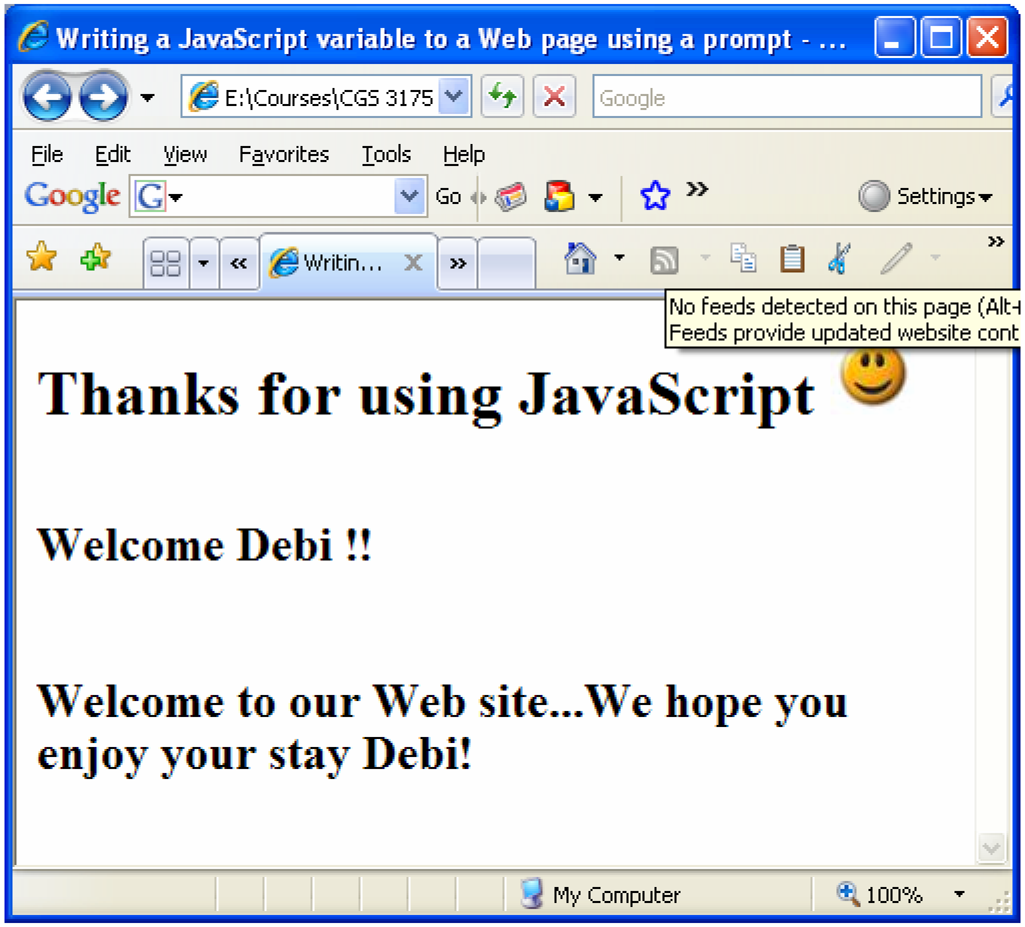
Cancel

Done

My Computer

100%

Initial page


Writing a JavaScript variable to a Web page using a prompt - ...

E:\Courses\CGS 3175

No feeds detected on this page (Alt+
Feeds provide updated website cont

Thanks for using JavaScript

Welcome Debi !!

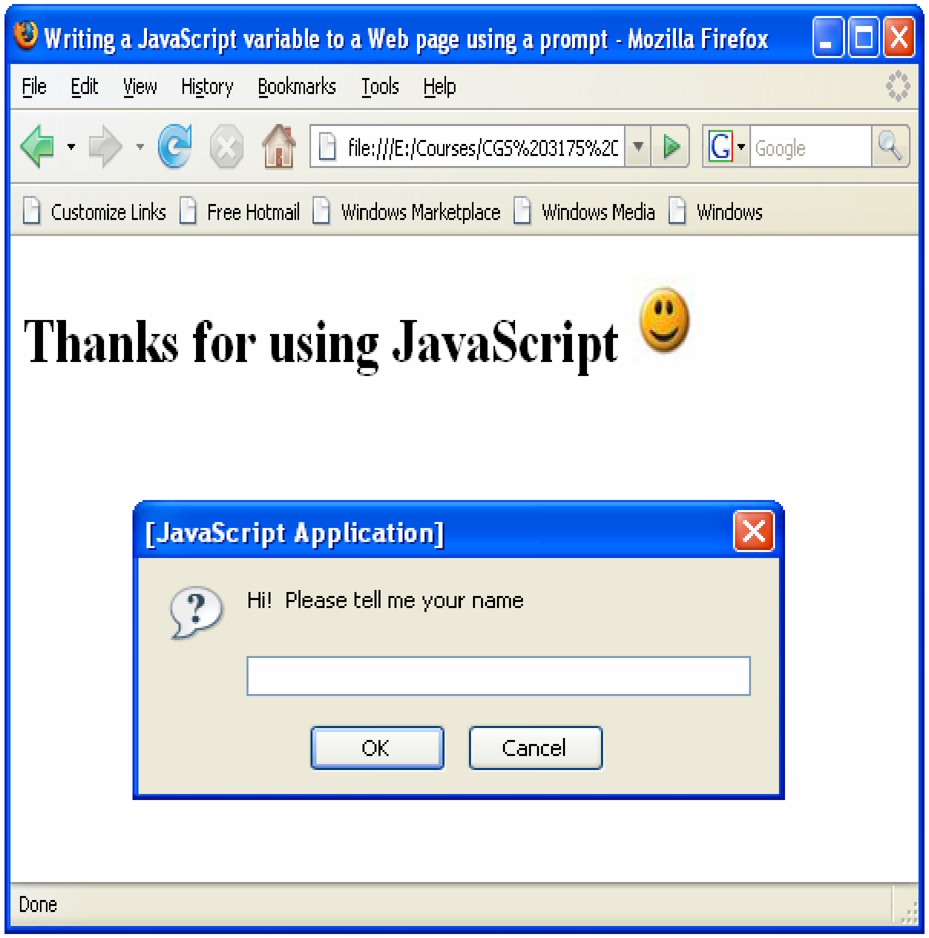
Welcome to our Web site...We hope you enjoy your stay Debi!

My Computer

100%

After visitor clicks “OK” in prompt window

Example – FireFox Version

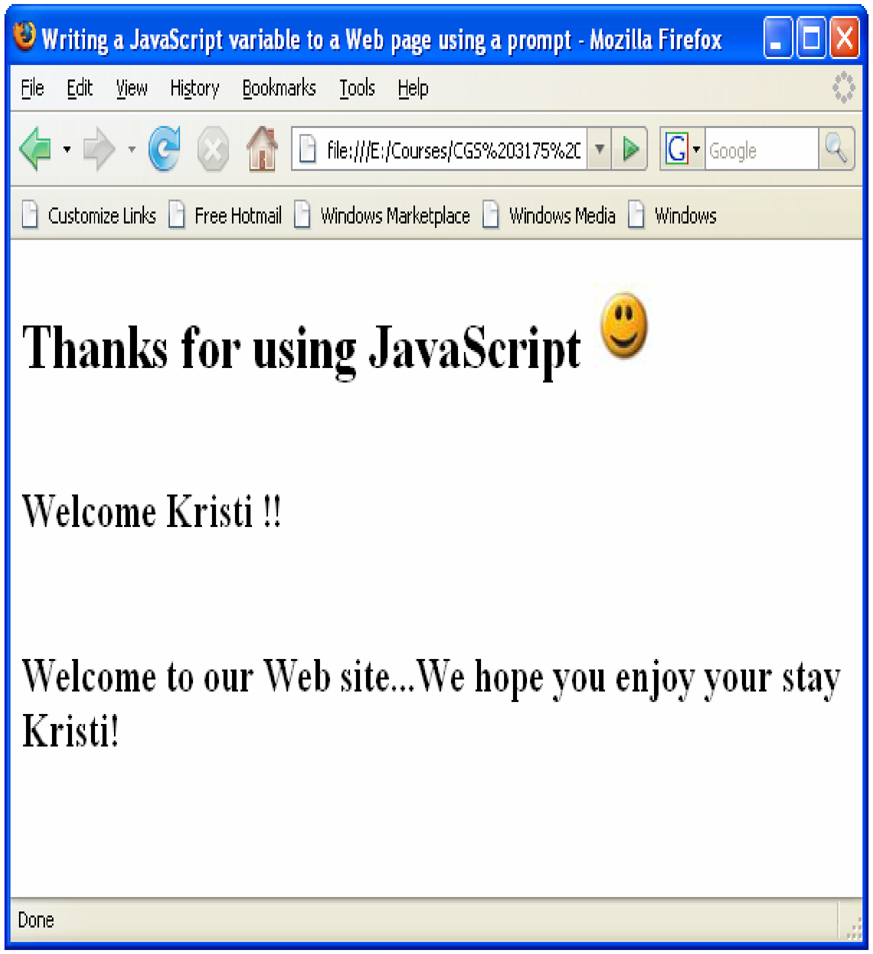

Writing a JavaScript variable to a Web page using a prompt - Mozilla Firefox
File Edit View History Bookmarks Tools Help
file:///E:/Courses/CGS%203175%2C
Google
Customize Links Free Hotmail Windows Marketplace Windows Media Windows
OK
Cancel
Done

Thanks for using JavaScript

[JavaScript Application]

Hi! Please tell me your name

Initial page


Writing a JavaScript variable to a Web page using a prompt - 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
Done

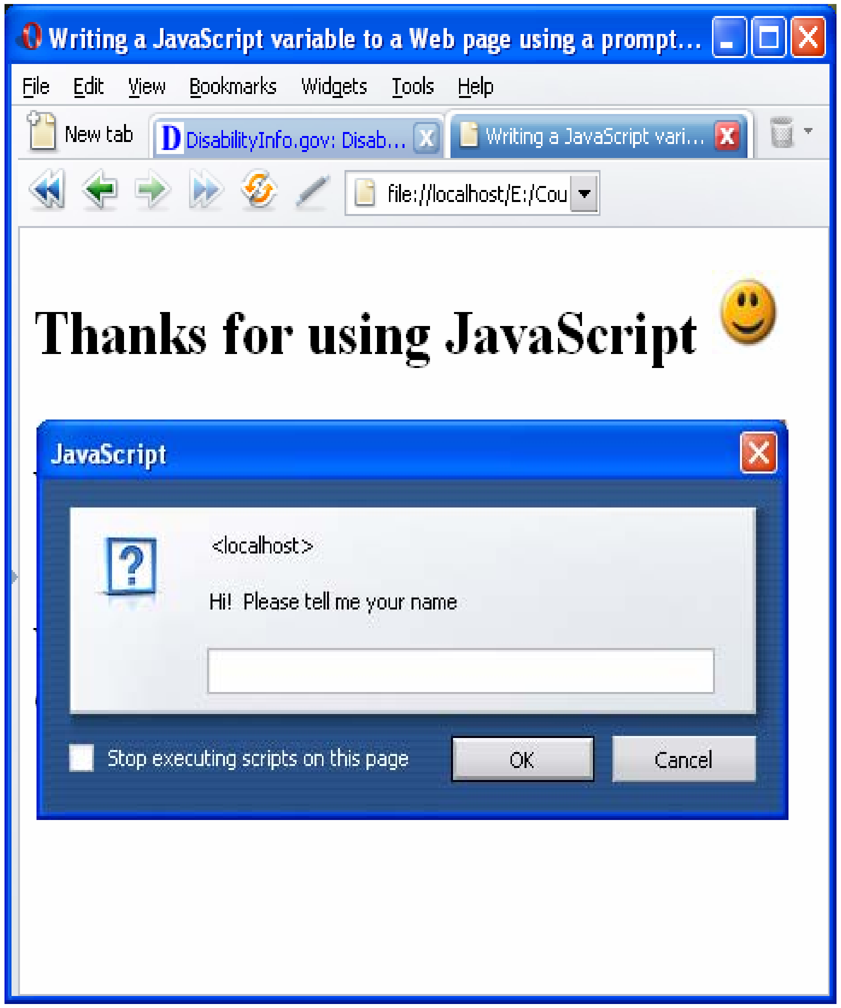
Thanks for using JavaScript

Welcome Kristi !!

Welcome to our Web site...We hope you enjoy your stay Kristi!

After visitor clicks "OK" in prompt window

Example – Opera Version



Writing a JavaScript variable to a Web page using a prompt...

File Edit View Bookmarks Widgets Tools Help

New tab | DisabilityInfo.gov: Disab... | Writing a JavaScript vari...

file://localhost/E:/Cou

Thanks for using JavaScript

JavaScript

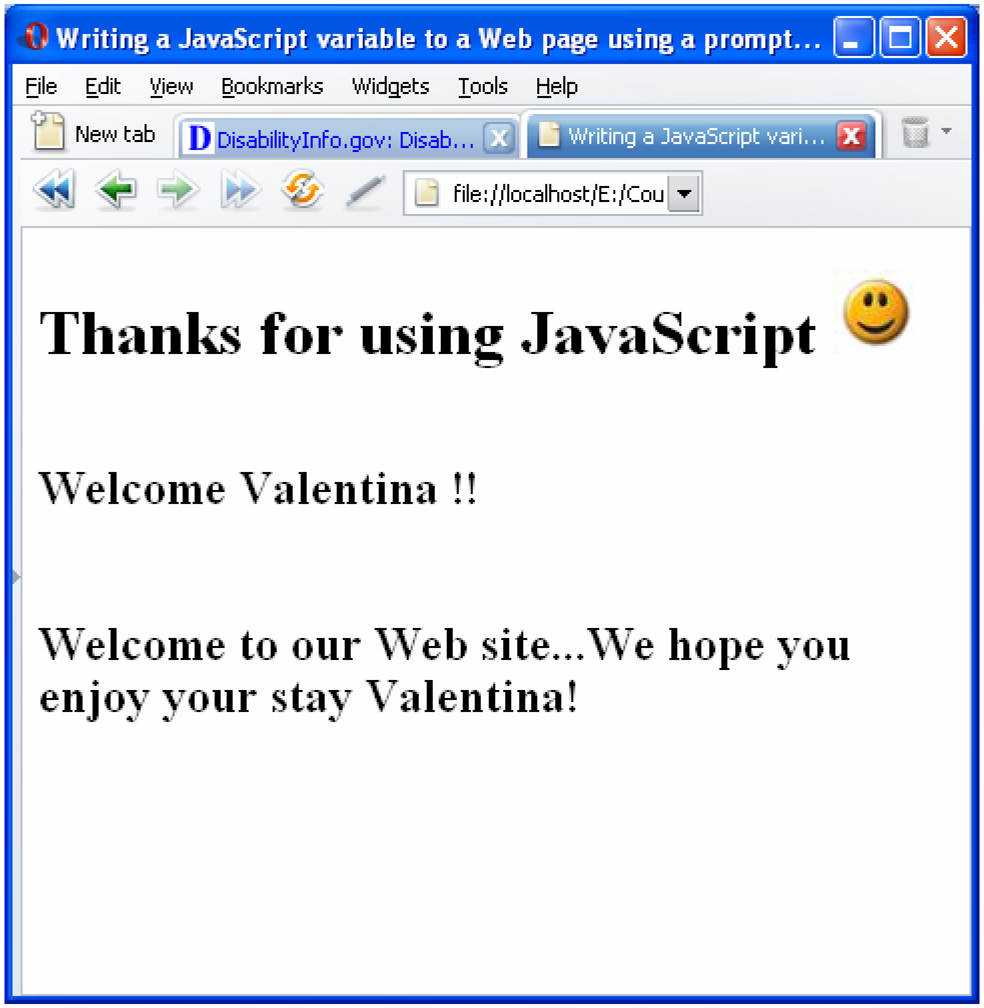
<localhost>

Hi! Please tell me your name

Stop executing scripts on this page

OK Cancel

Initial page



Writing a JavaScript variable to a Web page using a prompt...

File Edit View Bookmarks Widgets Tools Help

New tab | DisabilityInfo.gov: Disab... | Writing a JavaScript vari...

file://localhost/E:/Cou

Thanks for using JavaScript

Welcome Valentina !!

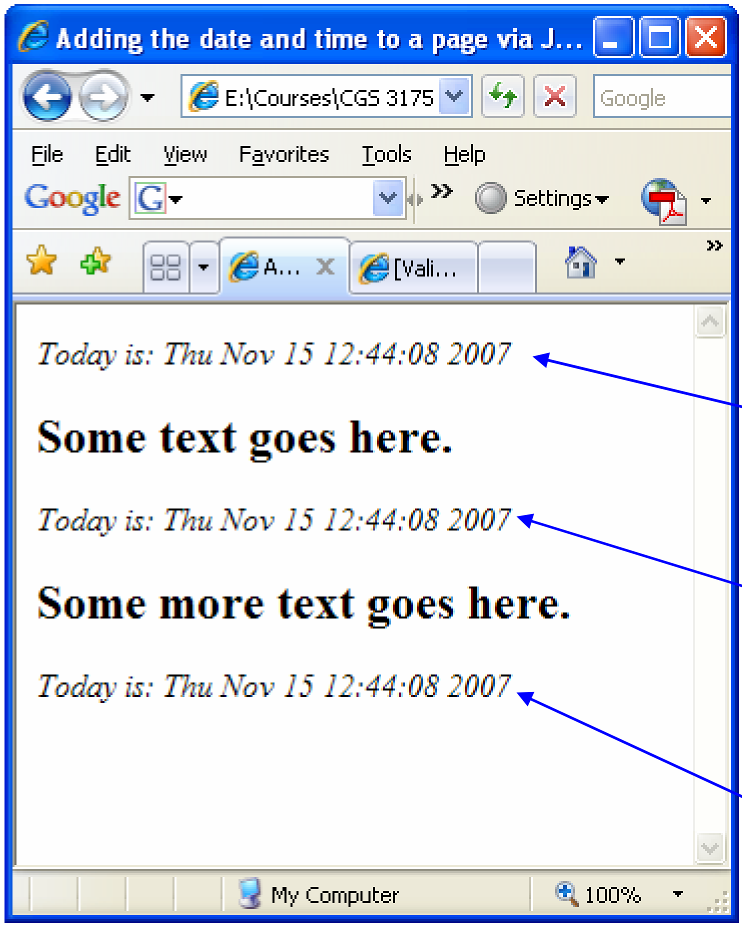
Welcome to our Web site...We hope you enjoy your stay Valentina!

After visitor clicks “OK” in prompt window

Functions In JavaScript

- A function is a set of JavaScript statements that perform some task.
- Every function must have a name and is invoked (or called) by other parts of a script. A function can be called as many times as needed during the running of a script (just like you can use the value of a variable as many times as you need).
- Look at the rendering of the XHTML document shown on the next page. Notice that the date and time appear three times. The XHTML document that produced this is also shown. Notice that the script to produce the date and time, appears three times.

Functions In JavaScript


Adding the date and time to a page via J...
E:\Courses\CGS 3175
Google
File Edit View Favorites Tools Help
Settings
My Computer
100%

Today is: Thu Nov 15 12:44:08 2007

Some text goes here.

Today is: Thu Nov 15 12:44:08 2007

Some more text goes here.

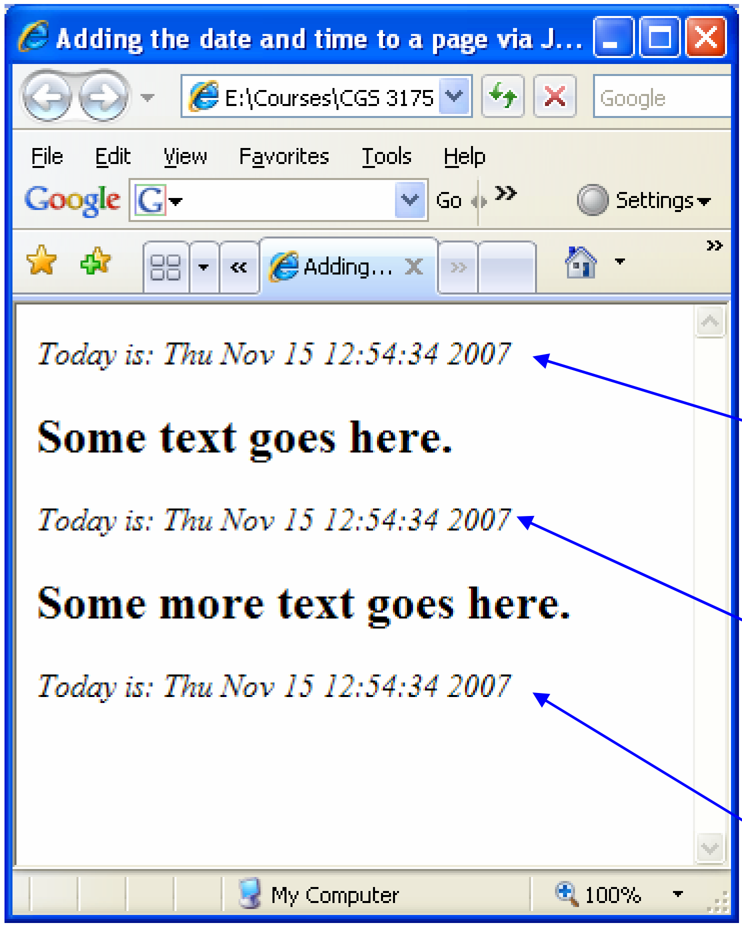
Today is: Thu Nov 15 12:44:08 2007

```
<?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>Adding the date and time to a page via
JavaScript  </title>
</head>
<body>
<div style="align:left">
<i> <script type="text/javascript">
    document.write("Today is: " + Date());
</script> </i> <br />
<h2> Some text goes here. </h2>
</div>
<div style="align:right">
<i> <script type="text/javascript">
    document.write("Today is: " + Date());
</script> </i> <br />
<h2> Some more text goes here.</h2>
</div>
<div style="align:left">
<i> <script type="text/javascript">
    document.write("Today is: " + Date());
</script> </i> <br />  </div>
</body>
</html>
```

Functions In JavaScript

- What a function allows us to do is simplify our XHTML document, by not requiring us to duplicate the script each time we would like to have its effect placed into the document.
- Look at the next page, which produces an identical rendering in a browser. Notice that the code contains only a single appearance of the script code, this time as a function.
  - In this example, since the script itself is small, there is not a lot of space saved using a function, but at least we only had to write the actual script once. We'll see more advantages with functions as we progress to larger examples.

Functions In JavaScript


Adding the date and time to a page via J...

Today is: Thu Nov 15 12:54:34 2007

Some text goes here.

Today is: Thu Nov 15 12:54:34 2007

Some more text goes here.

Today is: Thu Nov 15 12:54:34 2007

```
<?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>Adding the date and time to a page via
JavaScript  </title>
<script type="text/javascript">
    function writeDateAndTime() {
       document.write("Today is: " + Date());
    }
</script>
</head>
<body>
<div style="align:left">
<i> <script type="text/javascript">
   writeDateAndTime();
</script> </i> <br />
<h2> Some text goes here. </h2>
</div>
<div style="align:right">
<i>  <script type="text/javascript">
   writeDateAndTime();
</script> </i> <br />
<h2> Some more text goes here.</h2>
</div>
<div style="align:left">
<i> <script type="text/javascript">
   writeDateAndTime();
</script> </i> <br /> </div>
</body>  </html>
```

Things to Try Yourself

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