CIS 4004: Web Based Information Technology Spring 2011

Introduction To JavaScript – Part 2

Instructor : Dr. Mark Llewellyn
markl@cs.ucf.edu
HEC 236, 407-823-2790
http://www.cs.ucf.edu/courses/cis4004/spr2011

Department of Electrical Engineering and Computer Science
University of Central Florida

Triggering A Script

- In the examples from part 1 of the JavaScript notes, the scripts were triggered automatically. In other words, the visitor didn't need to do anything for the script to execute.
- These were "automatically triggered" scripts. Sometimes you do not want the script to run until the visitor does something to trigger it. For example, you might want to run a script when the visitor mouses over a particular image or link, or when a page is loaded.
- These actions – mousing over or loading a page – are called intrinsic events.
- There are currently 18 predefined intrinsic events you can use as triggers to determine when a script will run. The table on the next couple of pages list these intrinsic events and which elements they work with.

Table of Intrinsic Events

Event	Works With	When
onblur	`<a>, <area>,<button>,<input>, <label>, <select>, <textarea>`	The visitor leaves an element that was previously in focus (see onfocus below).
onchange	`<input>, <select>, <textarea>`	The visitor modifies the value or contents of the element.
onclick	All elements *except* `<applet>, <base>, <basefont>,  , <font>, <frame>, <frameset>, <head>, <html>, <iframe>, <meta>, <param>, <script>, <style>, <title>`	The visitor clicks on the specified area.
ondblclick	Same as for `onclick`	The visitor double clicks the specified area.
onfocus	`<a>, <area>,<button>,<input>, <label>, <select>, <textarea>`	The visitor selects, clicks, or tabs to the specified element.
onkeydown	`<input>` (of type name or password), `<textarea>`	The visitor types something in the specified element.
onkeypress	`<input>` (of type name or password), `<textarea>`	The visitor types something in the specified element.
onkeyup	`<input>` (of type name or password), `<textarea>`	The visitor lets go of the key after typing in the specified element.

Table of Intrinsic Events (continued)

Event	Works With	When
onload	`<body>, <frameset>`	The page is loaded in the browser.
onmousedown	Same as for `onclick`	The visitor presses the mouse button down over the element.
onmousemove	Same as for `onclick`	The visitor moves the mouse over the specified element after having pointed at it.
onmouseout	Same as for `onclick`	The visitor moves the mouse away from the specified element after having been over it.
onmouseover	Same as for `onclick`	The visitor points the mouse at the element.
onmouseup	Same as for `onclick`	The visitor lets the mouse button go after having clicked on the element.
onreset	`form` (not input of type `reset`)	The visitor clicks the form's reset button.
onselect	`<input>` (of type name or password), `<textarea>`	The visitor selects one or more characters or words in the element.
onsubmit	`form` (not input of type `submit`)	The visitor clicks the form's submit button.
onunload	`<body>, <frameset>`	The browser loads a different page after the specified page had been loaded.

Using An Intrinsic Event – `onclick`

C:\Courses\CIS 4004 - Web Based IT\Spring 2011\code\JavaScript\JavaScript - Part 2\triggering a script - onclick.html - Notepad++

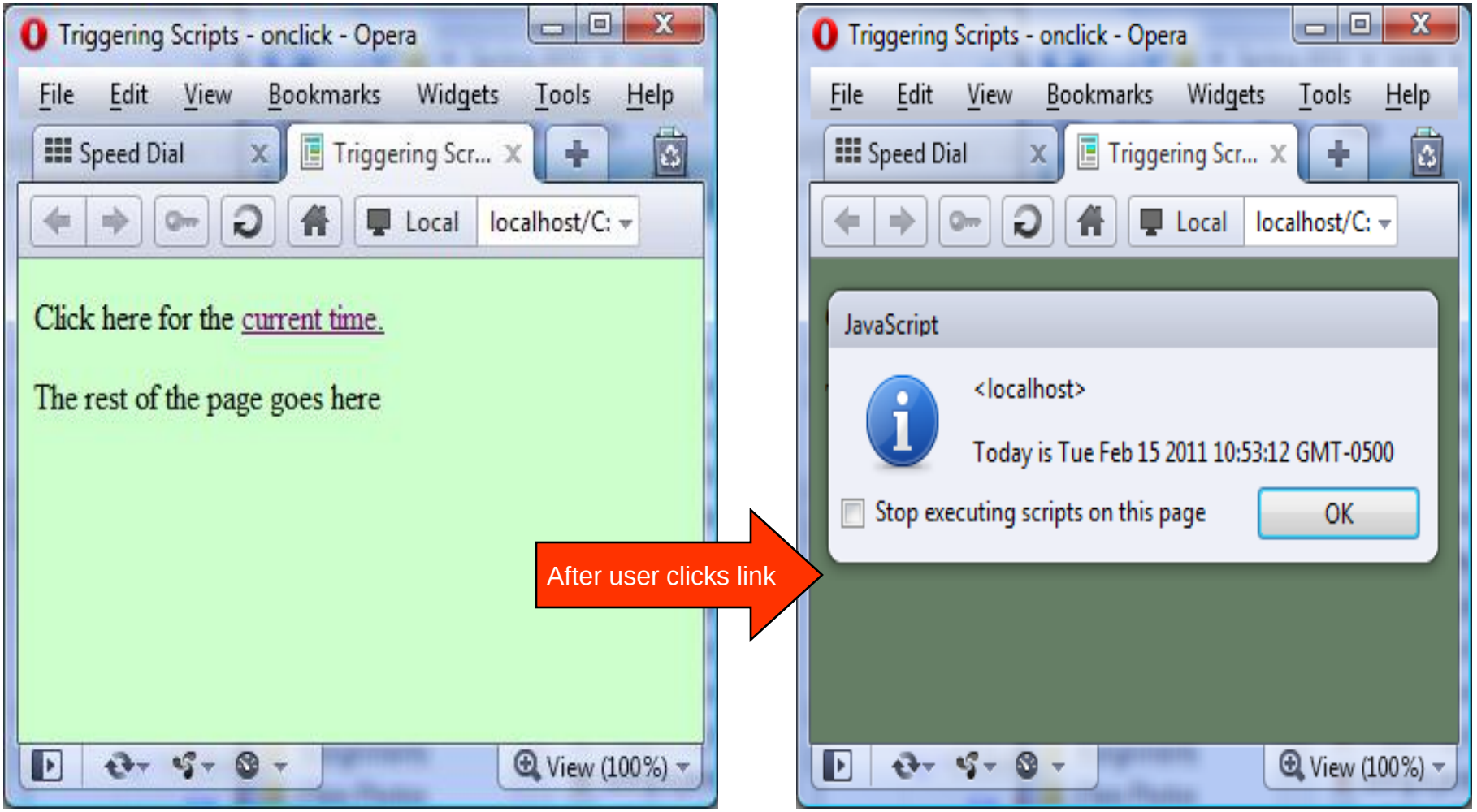
File Edit Search View Format Language Settings Macro Run TextFX Plugins Window ? X

first XHTML with JavaScript - version 2.html | first XHTML with JavaScript - version 3.html | first XHTML with JavaScript - version 4.html | triggering a script - onclick.html

```
1  <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3      "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4  <html xmlns="http://www.w3.org/1999/xhtml">
5  <head>
6      <title>Triggering Scripts - onclick  </title>
7      <style type="text/css">
8          <!-- body {background-color:#CCFFCC; }
9          -->
10     </style>
11 </head>
12 <body>
13     <p>Click here for the
14         <a href="triggering a script - onclick.html"
15         onclick="alert('Today is '+ Date() ) "> current time.</a>
16     </p>
17     <p>The rest of the page goes here</p>
18 </body>
19 </html>
20
```

Hyper Text Markup Language file | nb char : 576 nb line : 20 | Ln : 1 Col : 1 Sel : 0 | Dos\Windows ANSI | INS

Using An Intrinsic Event – `onclick`



Using An Intrinsic Event – onmouseover

C:\Courses\CIS 4004 - Web Based IT\Spring 2011\code\JavaScript\JavaScript - Part 2\triggering a script - onmouseover.html - Notepad++

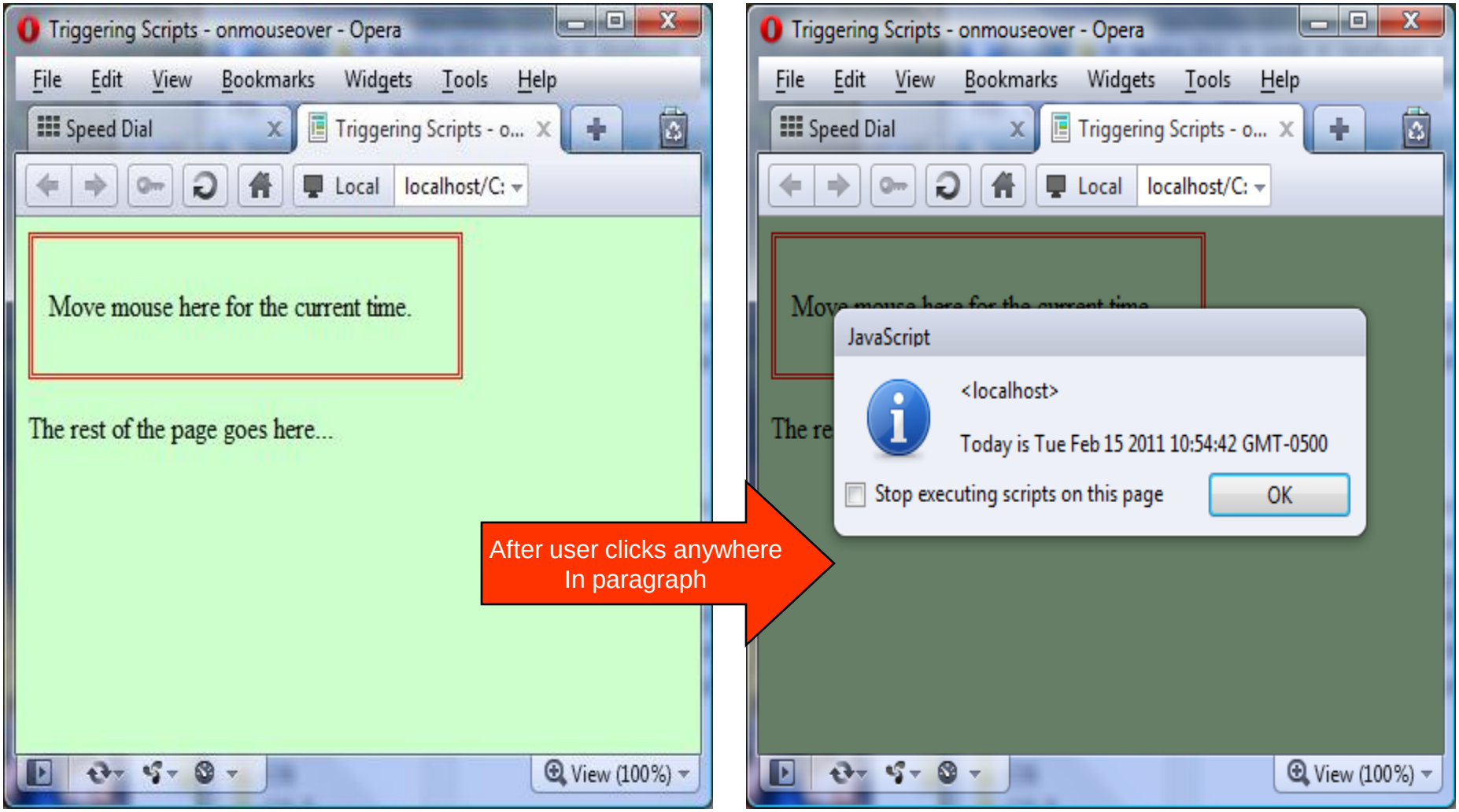
File Edit Search View Format Language Settings Macro Run TextFX Plugins Window ?

first XHTML with JavaScript - version 3.html | first XHTML with JavaScript - version 4.html | triggering a script - onclick.html | triggering a script - onmouseover.html

```
1  <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3      "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4  <html xmlns="http://www.w3.org/1999/xhtml">
5  <head>
6      <title>Triggering Scripts - onmouseover  </title>
7      <style type="text/css">
8          <!-- body {background-color:#CCFFCC; }
9              div {border:3px double red; padding: 10px; width:250px; }
10         -->
11     </style>
12 </head>
13 <body>
14     <div>
15         <p onmouseover="alert('Today is '+ Date() ) ">
16            Move mouse here for the current time.</p>
17     </div>
18     <p>The rest of the page goes here...</p>
19 </body>
20 </html>
```

Hyper Text Markup Language file | nb char : 626 nb line : 21 | Ln : 1 Col : 1 Sel : 0 | Dos\Windows ANSI | INS

Using An Intrinsic Event – `onmouseover`



Using An Intrinsic Event – onmousedown

C:\Courses\CIS 4004 - Web Based IT\Spring 2011\code\JavaScript\JavaScript - Part 2\triggering a script - onmousedown.html - Notepad++

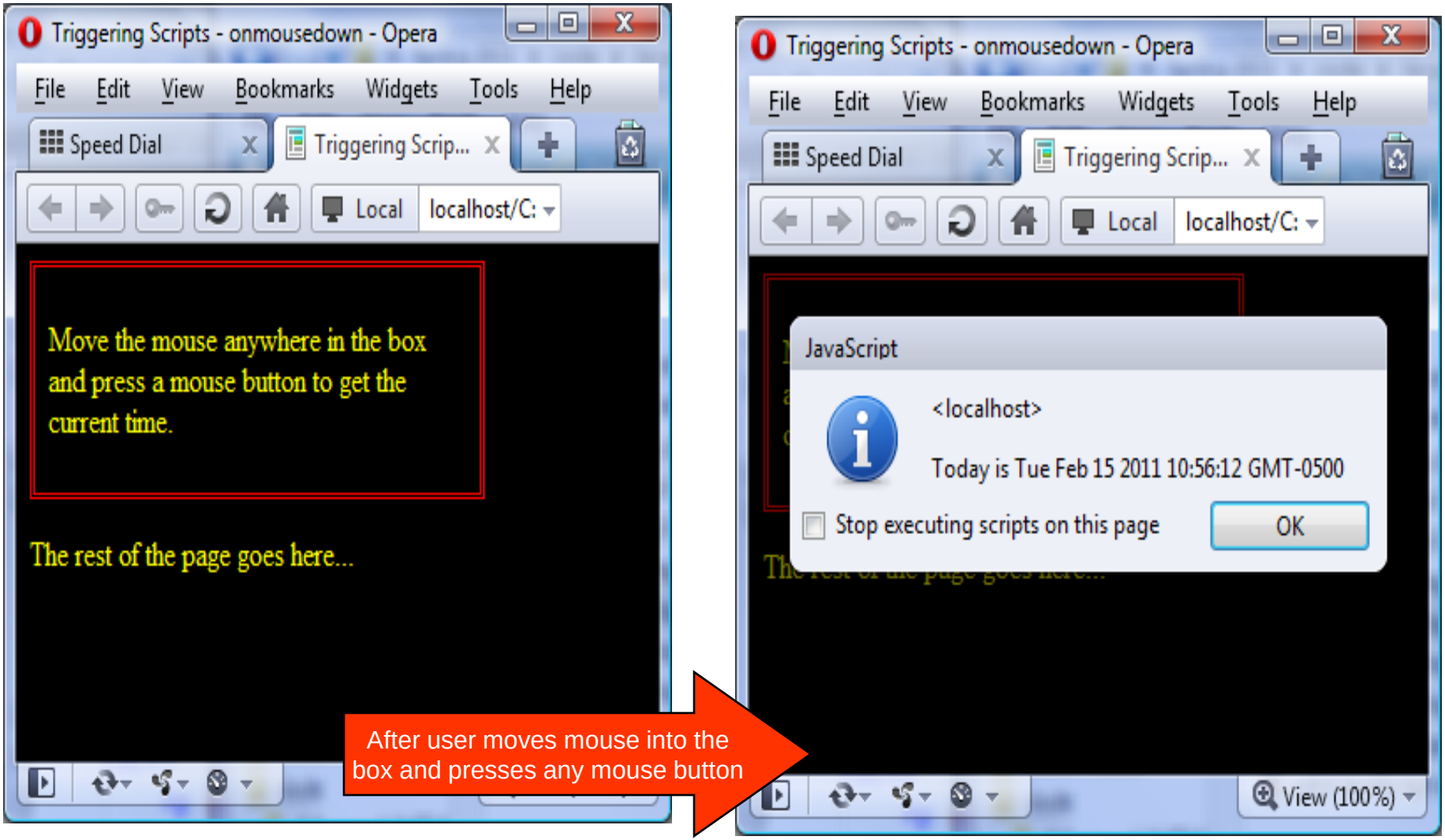
File Edit Search View Format Language Settings Macro Run TextFX Plugins Window ?

first XHTML with JavaScript - version 4.html | triggering a script - onclick.html | triggering a script - onmouseover.html | triggering a script - onmousedown.html

```
1  <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4  <html xmlns="http://www.w3.org/1999/xhtml">
5  <head>
6      <title>Triggering Scripts - onmousedown  </title>
7      <style type="text/css">
8          <!-- body {background-color:black; color: yellow;  }
9              div {border:3px double red; width:250px; padding: 8px; }
10         -->
11     </style>
12 </head>
13 <body>
14     <div>
15         <p onmousedown="alert('Today is '+ Date() ) ">
16            Move the mouse anywhere in the box and press a mouse
17            button to get the current time.</p>
18     </div>
19     <p>The rest of the page goes here...</p>
20 </body>
21 </html>
```

Hyper Text Markup Language file | nb char : 691 nb line : 22 | Ln : 1 Col : 1 Sel : 0 | Dos\Windows ANSI | INS

Using An Intrinsic Event – `onmousedown`



Creating A Button To Trigger A Script

- You can associate a button with a script to give your visitor full control over when the script should be executed.
- As we did earlier, you simply create a button, then associate a script with the `onclick` intrinsic event. You can use any intrinsic event with a button, but `onclick` makes the most sense.
- You can also add images to buttons. Simply insert the image between the opening and closing button tags.
- The example on the next page illustrates using a button to trigger a script.

Creating A Button To Trigger A Script

triggering a script - onclick.html | triggering a script - onmouseover.html | triggering a script - onmousedown.html | button activated script.html

```
5  <head>
6       <title>A Button Activated Script</title>
7       <style type="text/css">
8       <!--
9            body {background-color: #33CCFF; }
10      -->
11      </style>
12 </head>
13 <body>
14 <div>
15 <!-- <button type="button" name="time" onclick="alert('Today is '+ Date() )"
16      style="font: 0.7em Helvetica, Arial, sans-serif; background:blue: color:black; padding:0.6em" >
17      <img src="clock.bmp" alt="a clock" /> Click For Time</button>
18 -->
19 <!--   <button type="button" name="time" onclick="alert('Today is '+ Date() )"
20      style="font: 0.7em Helvetica, Arial, sans-serif; background:blue: color:black; padding:.3em">Click For Time
21      <img src="clock.bmp" alt="a clock" /></button>
22 -->
23      <button type="button" name="time" onclick="alert('Today is '+ Date() )"
24           style="font: 0.7em Helvetica, Arial, sans-serif;
25                background:blue: color:black; padding:0.6em">
26           <img src="clock.bmp" alt="a clock" />
27      </button>
28 </div>
29 </body>
```

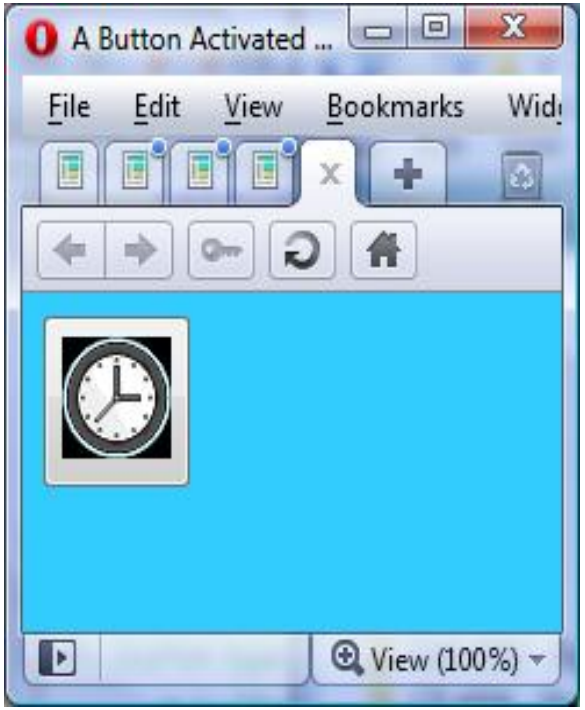
The three different lines show different versions of the button – try all three of them.

Hyper Text Markup Language file | nb char : 1125 nb line : 31 | Ln : 28 Col : 1 Sel : 0 | Dos\Windows ANSI INS

Creating A Button To Trigger A Script

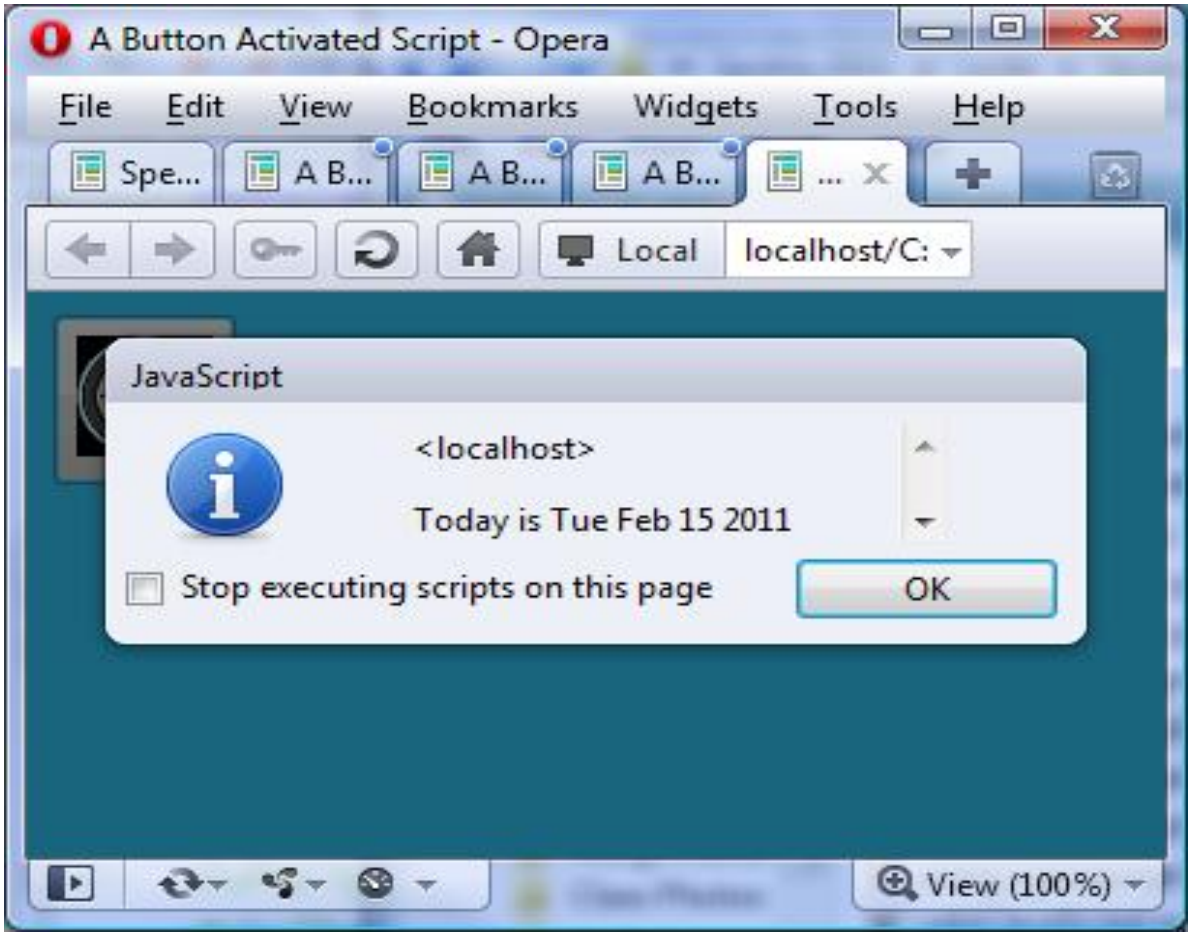
Using first line

Using second line



Using third line

Creating A Button To Trigger A Script



Window when script executes

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.

Writing Valid JavaScript Code

- 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 (i.e., create a JavaScript library file) 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 next page illustrates this technique.

Writing Valid JavaScript Code

- 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

C:\Courses\CIS 4004 - Web Based IT\Spring 2011\code\JavaScript\JavaScript - Part 2\valid JavaScript example.html - Notepad++

File Edit Search View Format Language Settings Macro Run TextFX Plugins Window ?

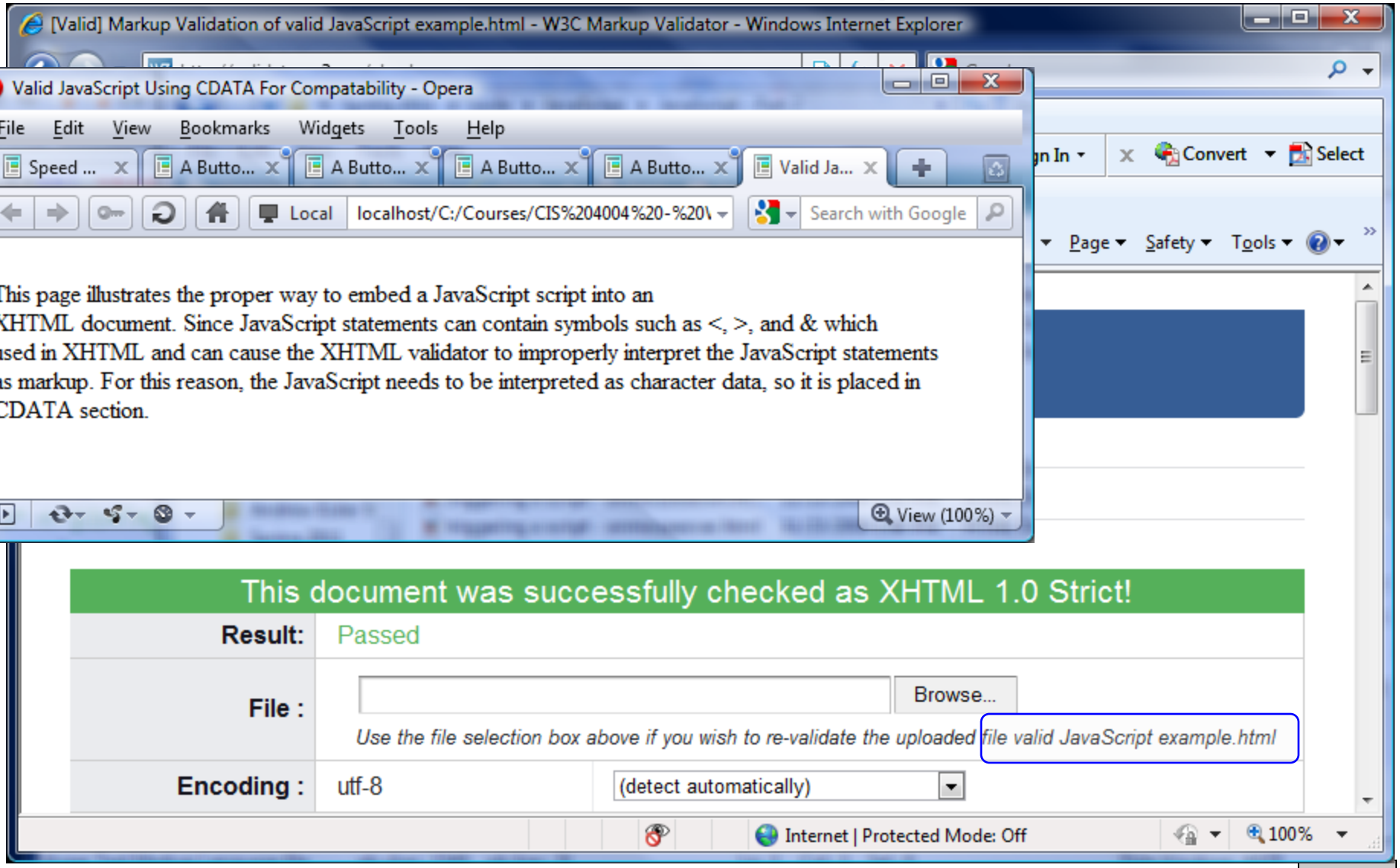
triggering a script - onmouseover.html | triggering a script - onmousedown.html | button activated script.html | valid JavaScript example.html

```
1  <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3      "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4  <html xmlns="http://www.w3.org/1999/xhtml">
5  <head>
6  <title>Valid JavaScript Using CDATA For Compatabi
7  </head>
8  <body>
9
10 <script type="text/javascript">
11 /* <!--[CDATA[ */
12     document.write("<br /> This page illustrates the proper way to embed a JavaScrip
13     document.write("XHTML document.  Since JavaScript statements can contain symbols
14     document.write("used in XHTML and can cause the XHTML validator to improperly in
15     document.write("as markup.  For this reason, the JavaScript needs to be interpre
16     document.write("CDATA section.<br />");
17 /* ]]--> */
18 </script>
```

If you remove the CDATA section this document will not validate against the Strict DTD.

Hyper Text Markup Language file | nb char : 1549 nb line : 31 | Ln : 1 Col : 1 Sel : 0 | Dos\Windows ANSI | INS

Writing Valid JavaScript Code



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.

Creating A JavaScript Library

- 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 in the next section of notes, 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.

A JavaScript Library
(page 1)

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

*C:\Courses\CIS 4004 - Web Based IT\Spring 2011\code\JavaScript\JavaScript - Part 2\myscriptlibrary2.js - Notepad++

File Edit Search View Format Language Settings Macro Run TextFX Plugins Window ?

triggering a script - onmouseover.html | triggering a script - onmousedown.html | button activated script.html | valid JavaSc

```
1  /* file name:  myscriptlibrary2.js */
2  /* This is a JavaScript library of scripts */
3  /* Created for JavaScript - Part 2 Lecture Notes */
4  /* MJL - Last Modified February 15, 2011 */
5
6  //SCRIPT #1
7  //this script writes the date and time onto a page
8
9      document.write("</p align='right'> Today is: <i>" + Date() + "</i></p>");
10     document.write("<br /> <br />");
11
12 //SCRIPT #2
13 //this script returns today's date
14
15    var currentTime = new Date()
16    var month = currentTime.getMonth() + 1
17    var day = currentTime.getDate()
18    var year = currentTime.getFullYear()
19    document.write("Today is: " + month + "/" + day + "/" + year)
20    document.write("<br /> <br />");
21
```

JavaScript file | nb char : 1543 nb line : 45 | Ln : 4 Col : 41 Sel : 0 | Dos\Windows ANSI INS

*C:\Courses\CIS 4004 - Web Based IT\Spring 2011\code\JavaScript\JavaScript - Part 2\myscriptlibrary2.js - Notepad++

File Edit Search View Format Language Settings Macro Run TextFX Plugins Window ?

triggering a script - onmouseover.html | triggering a script - onmousedown.html | button activated script.html | valid JavaScript example

A JavaScript Library (page 2)

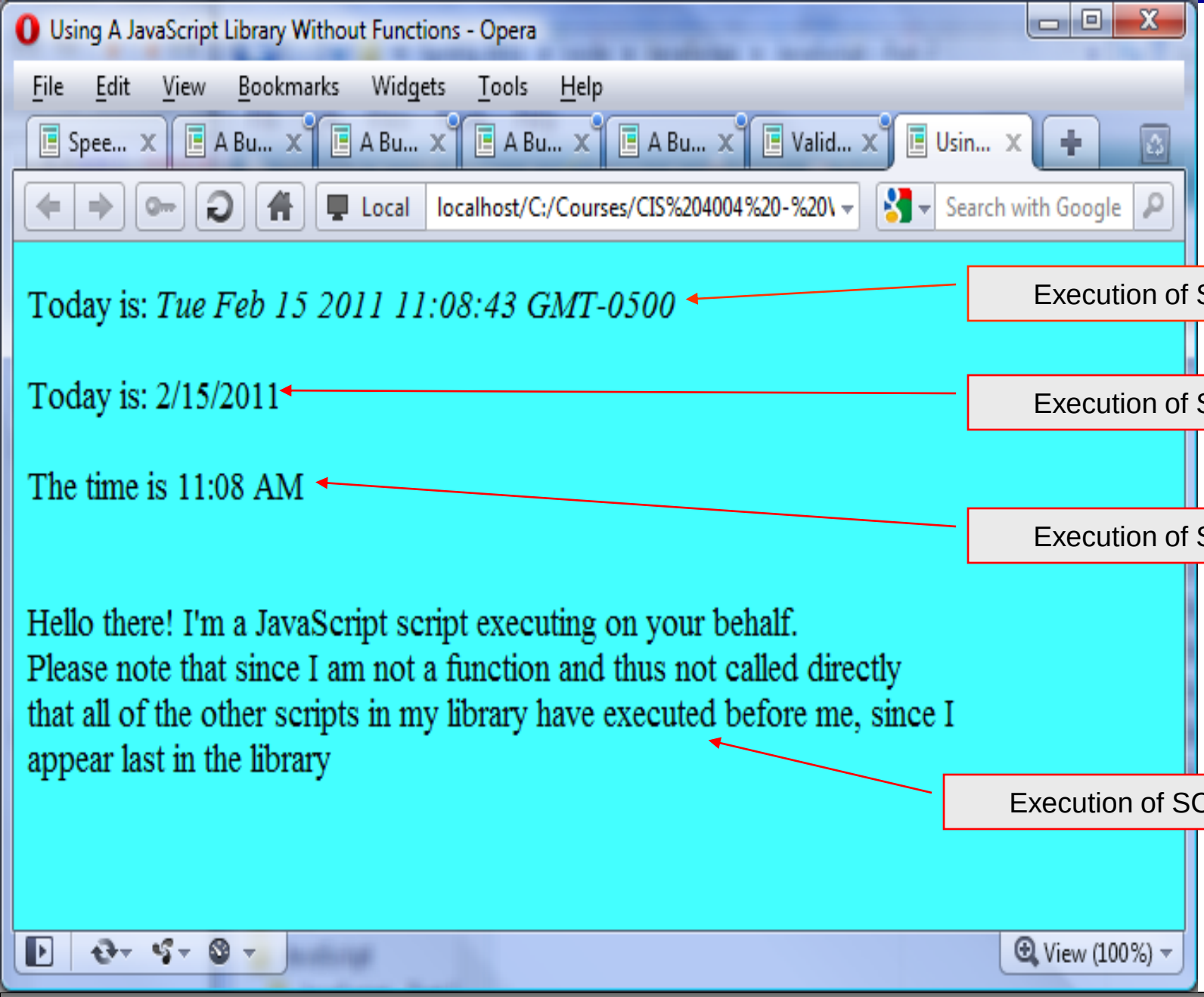
```
21  //SCRIPT #3
22  //this script returns the current time
23      var currentTime = new Date()
24      var hours = currentTime.getHours()
25      var minutes = currentTime.getMinutes()
26      if (minutes < 10)
27          minutes = "0" + minutes
28          document.write("The time is " + hours + ":" + minutes + " ")
29      if(hours > 11){
30          document.write("PM")
31      } else { document.write("AM")
32      }
33      document.write("<br /><br />");
34
35  //SCRIPT #4
36  //this script simply writes a message
37       document.write("<br /> Hello there!  I'm a JavaScript script executing on your
38      document.write("Please note that since I am not a function and thus not called
39      document.write("that all of the other scripts in my library have executed befor
40      document.write("appear last in the library <br />");
41
```

JavaScript file | nb char : 1534 nb line : 41 | Ln : 8 Col : 2 Sel : 1 | Dos\Windows ANSI INS

Execution Using A JavaScript Library

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

```
<
<
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <style type="text/css">
    <!--
        body{background-color:#4FF; font-size:14pt;}
    -->
    </style>
    <title>Using A JavaScript Library Without Functions </title>
</head>
<body>
    <script type="text/javascript"  src="myscriptlibrary2.js"></script>
</body>
</html>
```



Execution of SCRIPT #1

Execution of SCRIPT #2

Execution of SCRIPT #3

Execution of SCRIPT #4