

CGS 3175: Internet Applications Fall 2009

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/fall2009>

School of Electrical Engineering and Computer Science
University of Central Florida



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 page 5.)



Naming Variables

- Some examples:

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

Invalid identifiers: `Didi Thureau`, `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 | |

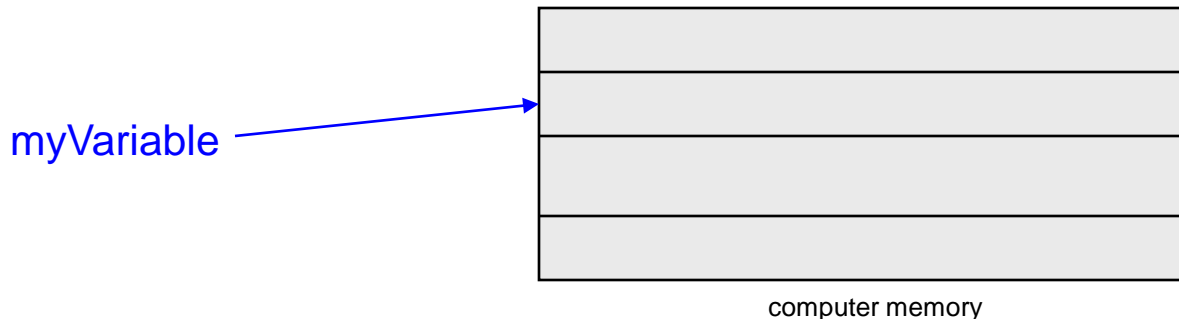


Using Variables

- 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

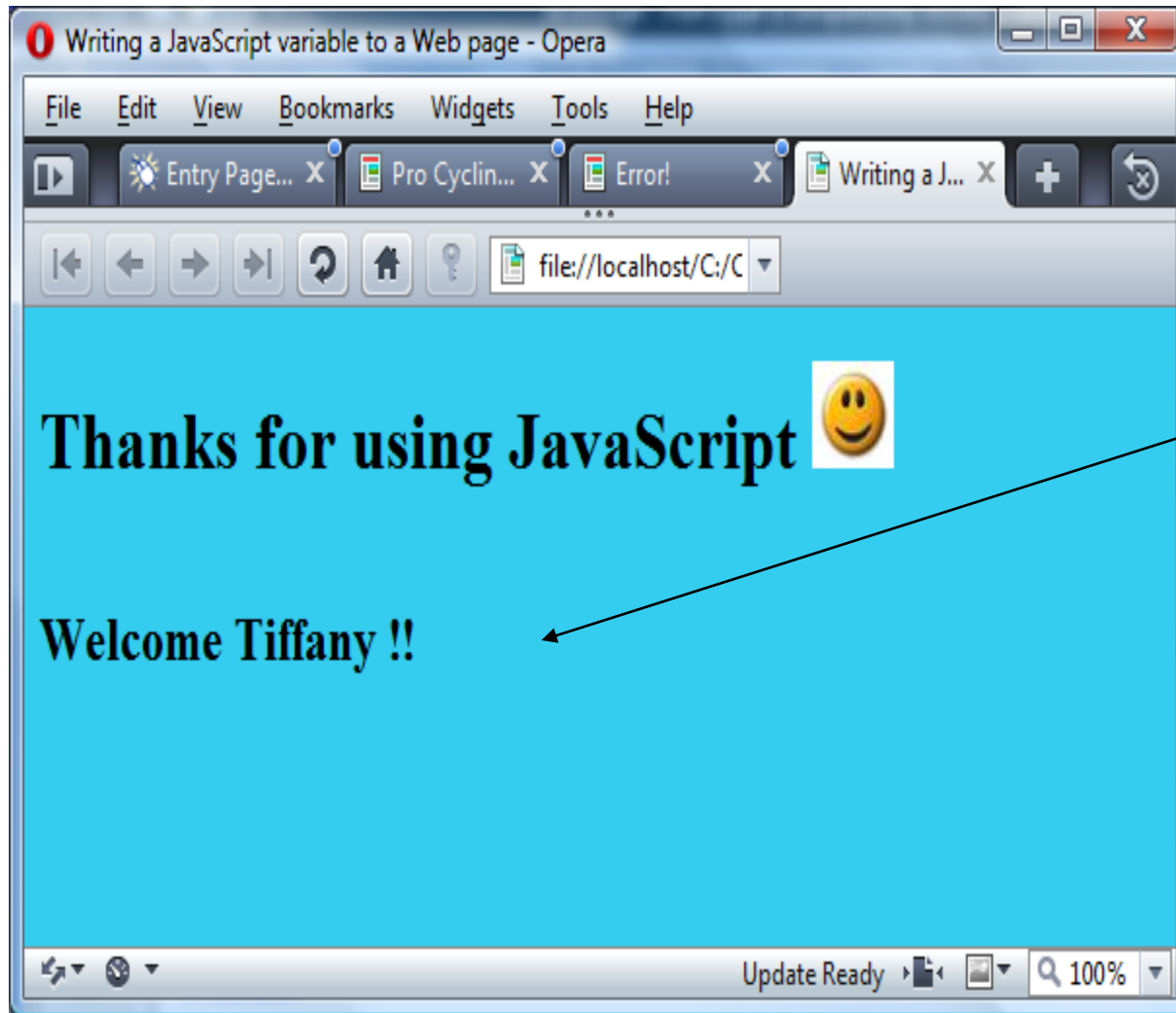
- While some variables will be used only internally to JavaScript, others will need to be sent to the Web page for display.
- This is quite simple in JavaScript and is basically no different than what we have already been doing with our simple scripts that have been printing text (strings) to the Web 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.




```
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> Writing a JavaScript variable to a Web page </title>
7   <style type="text/css">
8     <!--
9       body {background-color: #34CDEF;}
10     -->
11   </style>
12 </head>
13 <body>
14   <h1> Thanks for using JavaScript  </h1>
15   <h2>
16   <script type="text/javascript">
17     /* <!--[CDATA[ */
18       var userName = "Tiffany";
19       document.write("<br /> Welcome ");
20       document.write(userName);
21       document.write(" !!<br />");
22     /* ]]--> */
23   </script>
24   </h2>
25 </body>
26 </html>
```



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



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.

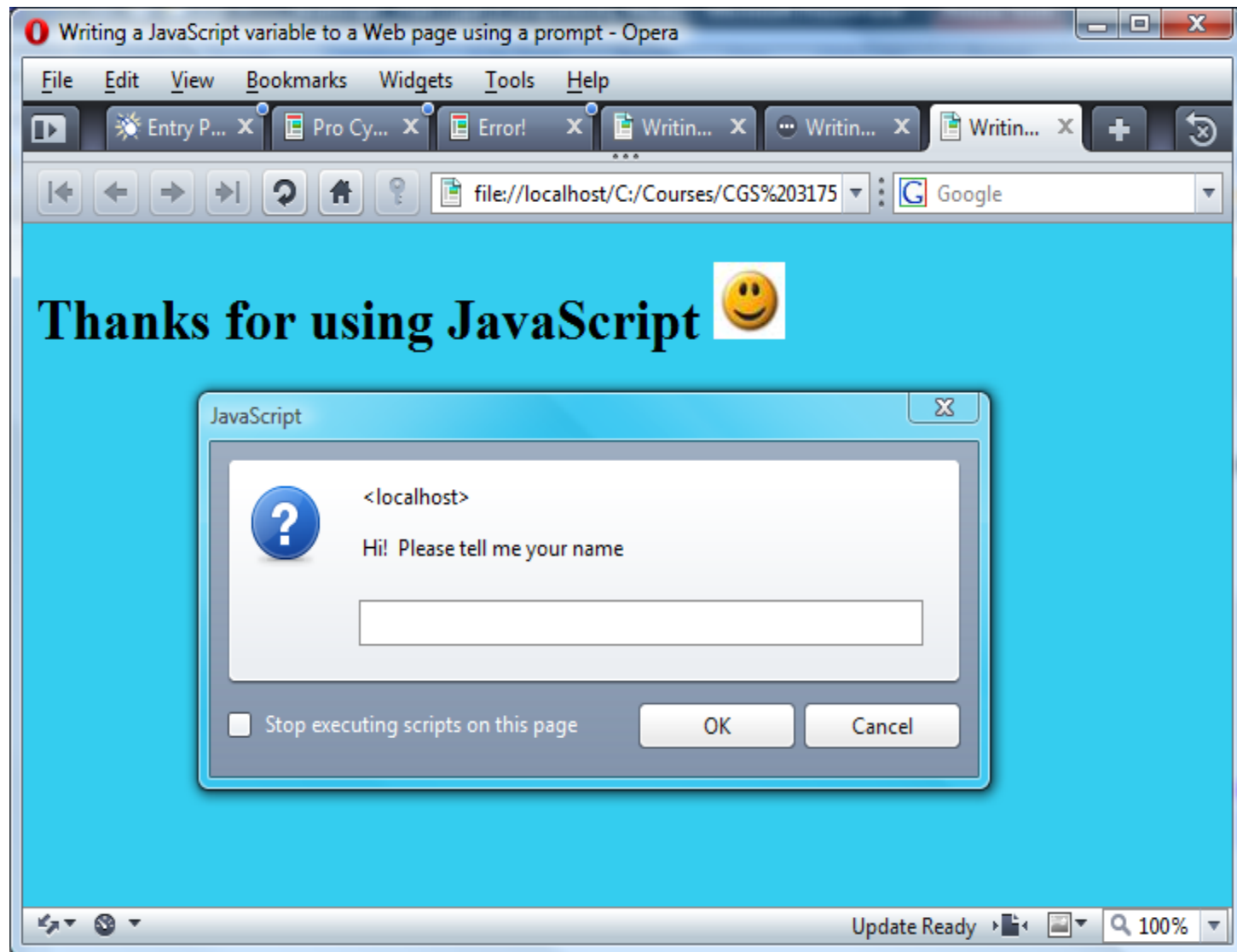
```
someVariable = prompt("prompt message");
```

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



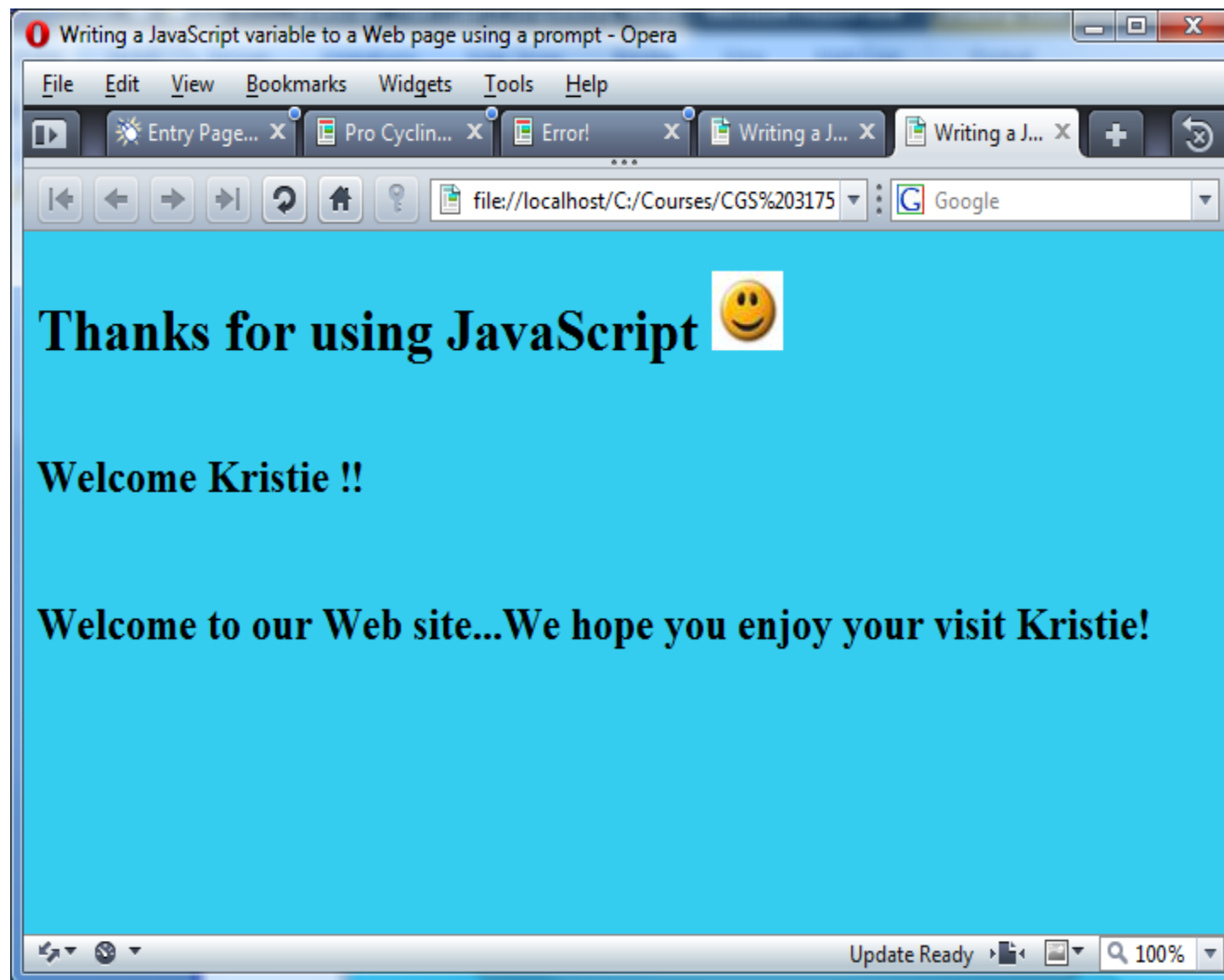
```
database.php formDatabase2.php dynamicForm2.php using a prompt with a variable.html
1 <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4 <html xmlns="http://www.w3.org/1999/xhtml">
5 <head>
6 <title> Writing a JavaScript variable to a Web page using a prompt </title>
7 <style type="text/css">
8 <!--
9 body {background-color: #34CDEF; }
10 -->
11 </style>
12 </head>
13 <body>
14 <h1> Thanks for using JavaScript  </h1>
15 <h2>
16 <script type="text/javascript">
17 /* <!-- [CDATA[ */
18 var userName;
19 userName = prompt("Hi! Please tell me your name");
20 document.write("<br /> Welcome ");
21 document.write(userName);
22 document.write(" !!<br /><br /> <br />");
23 document.write("Welcome to our Web site...We hope you enjoy your visit " + userName + "!!<br
24 /* ]]--> */
25 </script>
26 </h2>
27 </body>
28 </html>
```





Initial page





After visitor enters their name and clicks "OK" in prompt window



Assigning Variable Values Using A Prompt

- For a second example of using variables combined with user prompts, let's build a simple page that will ask the user to enter their favorite color and then reset the background color of the active window to their choice.
- In this case we'll be asking the user to enter a color and then use the value they enter as a parameter to a built-in JavaScript function that defines the background color of the active window.



C:\Courses\CGS 3175 - Internet Applications\Fall 2009\example code - new\JavaScript - Part 3\prompting a user's preference with a variable.html - Notepad++

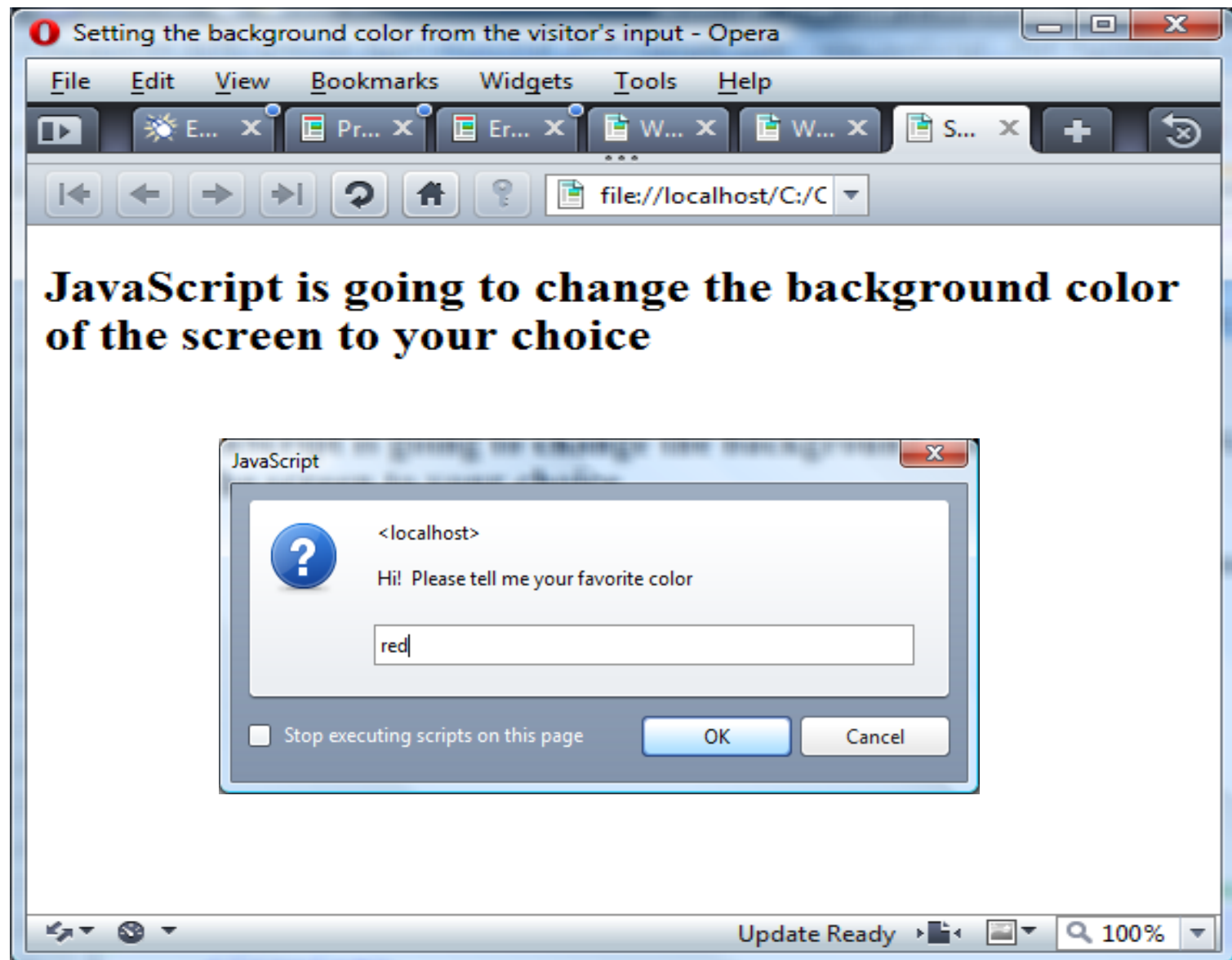
File Edit Search View Format Language Settings Macro Run TextFX Plugins Window ?

database.php formDatabase2.php dynamicForm2.php prompting a user's preference with a variable.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> Setting the background color from the visitor's input </title>
7 </head>
8 <body>
9   <h2> JavaScript is going to change the background color of the screen to your choice<
10  <h2>
11  <script type="text/javascript">
12    /* <!-- [CDATA[ */
13      var userFavoriteColor;
14      userFavoriteColor = prompt("Hi! Please tell me your favorite color");
15      window.document.bgColor = userFavoriteColor;
16      document.write("<br /><br /> <br />");
17      document.write("Thank you! I've changed the screen color to your choice.<br />")
18    /* ]]--> */
19  </script>
20  </h2>
21 </body>
22 </html>
```

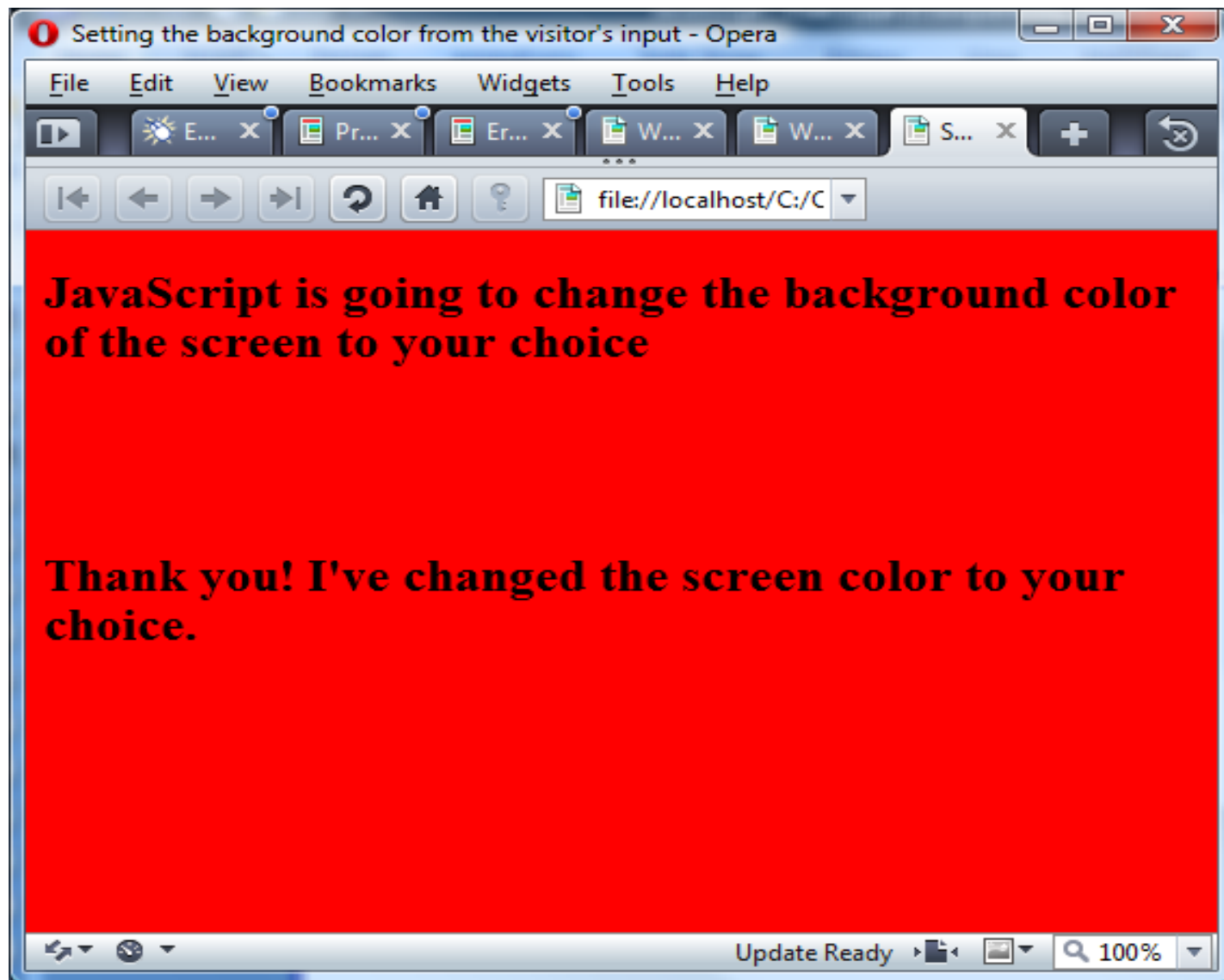
Hyper Text Markup Language file nb char: 812 nb line: 23 Ln: 15 Col: 53 Sel: 0 Dos\Windows ANSI INS





Initial page





After visitor enters their favorite color and clicks "OK" in prompt window

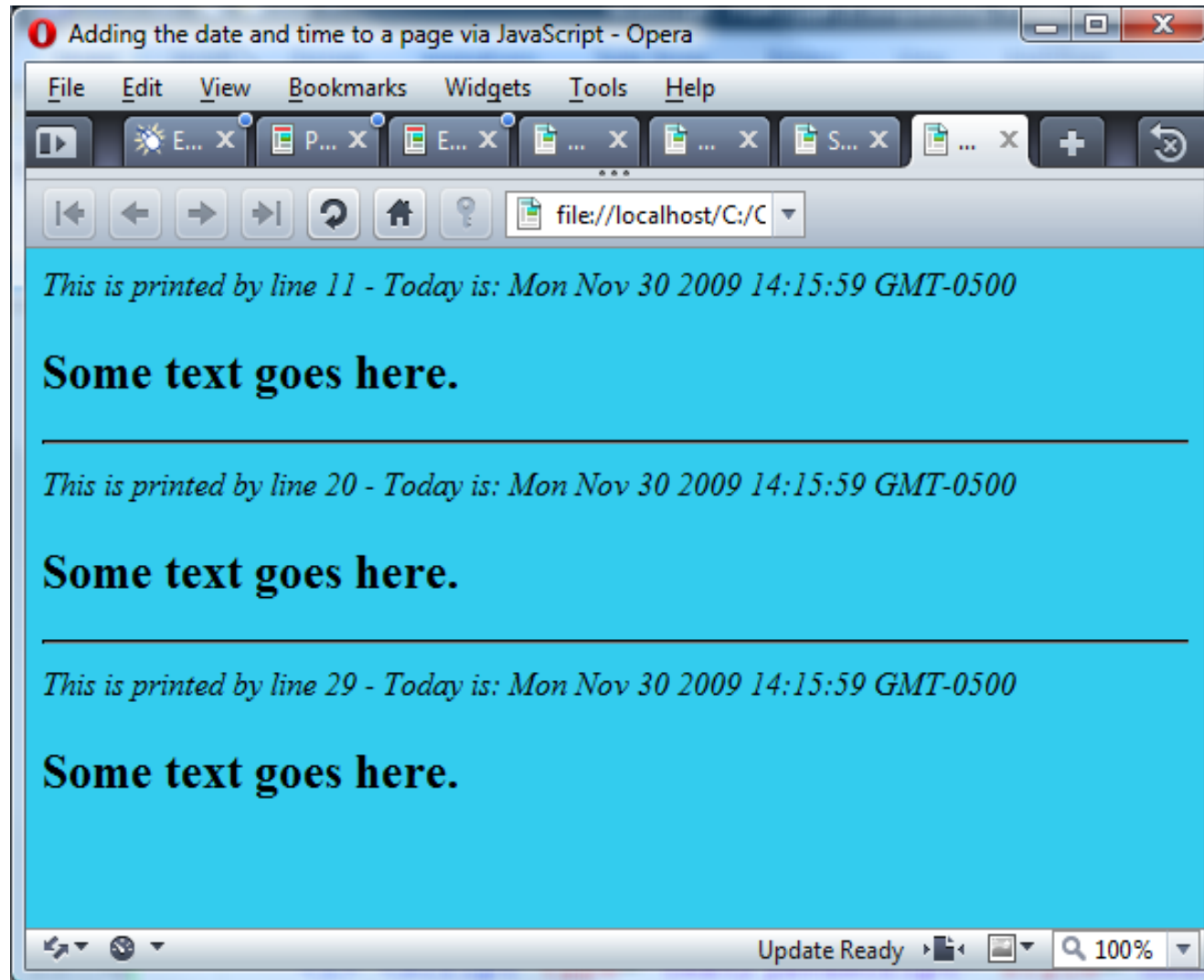


Functions In JavaScript

- 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 shown on the subsequent page. Notice that the script to produce the date and time, appears three times.
- This markup does not use a function, it simply repeats the script code every place we want the date/time to appear in the rendering, in this case in three places.



Functions In JavaScript



C:\Courses\CGS 3175 - Internet Applications\Fall 2009\example code - new\JavaScript - Part 3\date and time without functions.html - Notepad++

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

formDatabase2.php dynamicForm2.php prompting a user's preference with a variable.html date and time with a function.html date and time without functions.html

```
13 <body>
14   <div>
15     <i> <script type="text/javascript">
16       document.write("This is printed by line 11 - Today is: " + Date()); </script> </i>
17     <br />
18     <h2> Some text goes here. </h2>
19   </div>
20   <hr />
21   <div>
22     <i> <script type="text/javascript">
23       document.write("This is printed by line 20 - Today is: " + Date()); </script> </i>
24     <br />
25     <h2> Some text goes here.</h2>
26   </div>
27   <hr />
28   <div>
29     <i> <script type="text/javascript">
30       document.write("This is printed by line 29 - Today is: " + Date()); </script> </i>
31     <br />
32     <h2> Some text goes here.</h2>
33   </div>
34 </body>
35 </html>
36
```

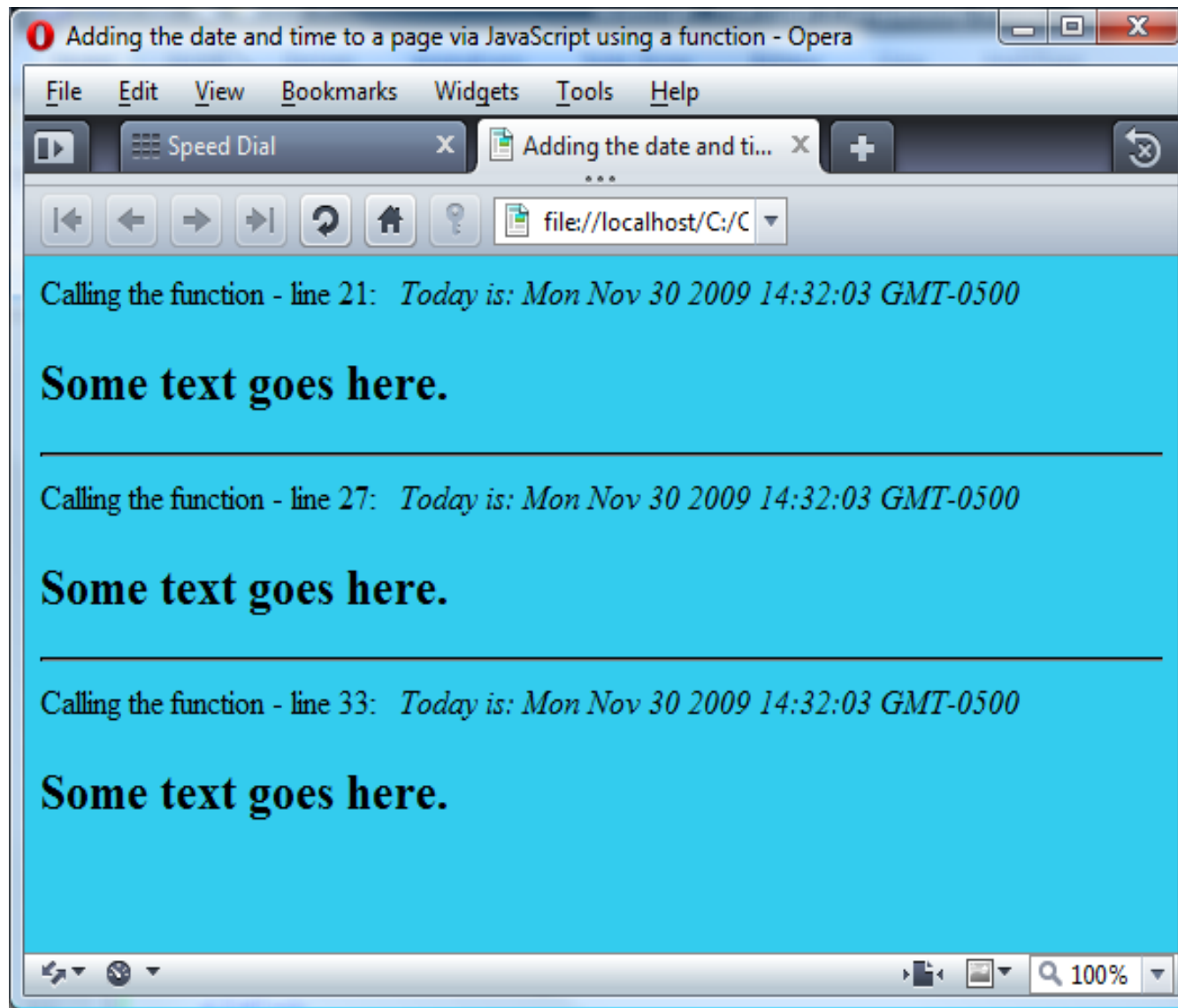
Hyper Text Markup Language file nb char: 998 nb line: 36 Ln: 15 Col: 43 Sel: 0 Dos\Windows ANSI INS



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.





*C:\Courses\CGS 3175 - Internet Applications\Fall 2009\example code - new\JavaScript - Part 3\date and time with a function.html - Notepad++

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

formDatabase2.php dynamicForm2.php prompting a user's preference with a variable.html date and time with a function.html date and time without fu

```
5 <head>
6   <title>Adding the date and time to a page via JavaScript using a function </title>
7   <script type="text/javascript">
8       function writeDateAndTime() {
9           document.write("Today is: " + Date());
10      }
11  </script>
12  <style type="text/css">
13      <!-- body {background-color: #33CCEE;} -->
14  </style>
15 </head>
16 <body>
17     <div>
18         Calling the function - line 21: &nbsp;
19         <i><script type="text/javascript"> writeDateAndTime(); </script> </i> <br />
20         <h2> Some text goes here. </h2>
21     </div>
22     <hr />
23     <div>
24         Calling the function - line 27: &nbsp;
25         <i><script type="text/javascript"> writeDateAndTime(); </script> </i> <br />
26         <h2> Some text goes here.</h2>
27     </div>
28     <hr />
29     <div>
30         Calling the function - line 33: &nbsp;
31         <i><script type="text/javascript"> writeDateAndTime(); </script> </i> <br />
32         <h2> Some text goes here. </h2>
33     </div>
```

Hyper Text Markup Language file nb char: 1108 nb line: 36 Ln: 31 Col: 80 Sel: 0 Dos\Window



Functions In JavaScript

- The next function example is similar to the first except that the function simply prints a person's name every time it is called. Note that it always prints the same name!
- This function does not invoke a built-in function (like `Date()` in the previous example), it simply prints a line of text (a name) every time it is invoked.



C:\Courses\CGS 3175 - Internet Applications\Fall 2009\example code - new\JavaScript - Part 3\simple function example.html - Notep...

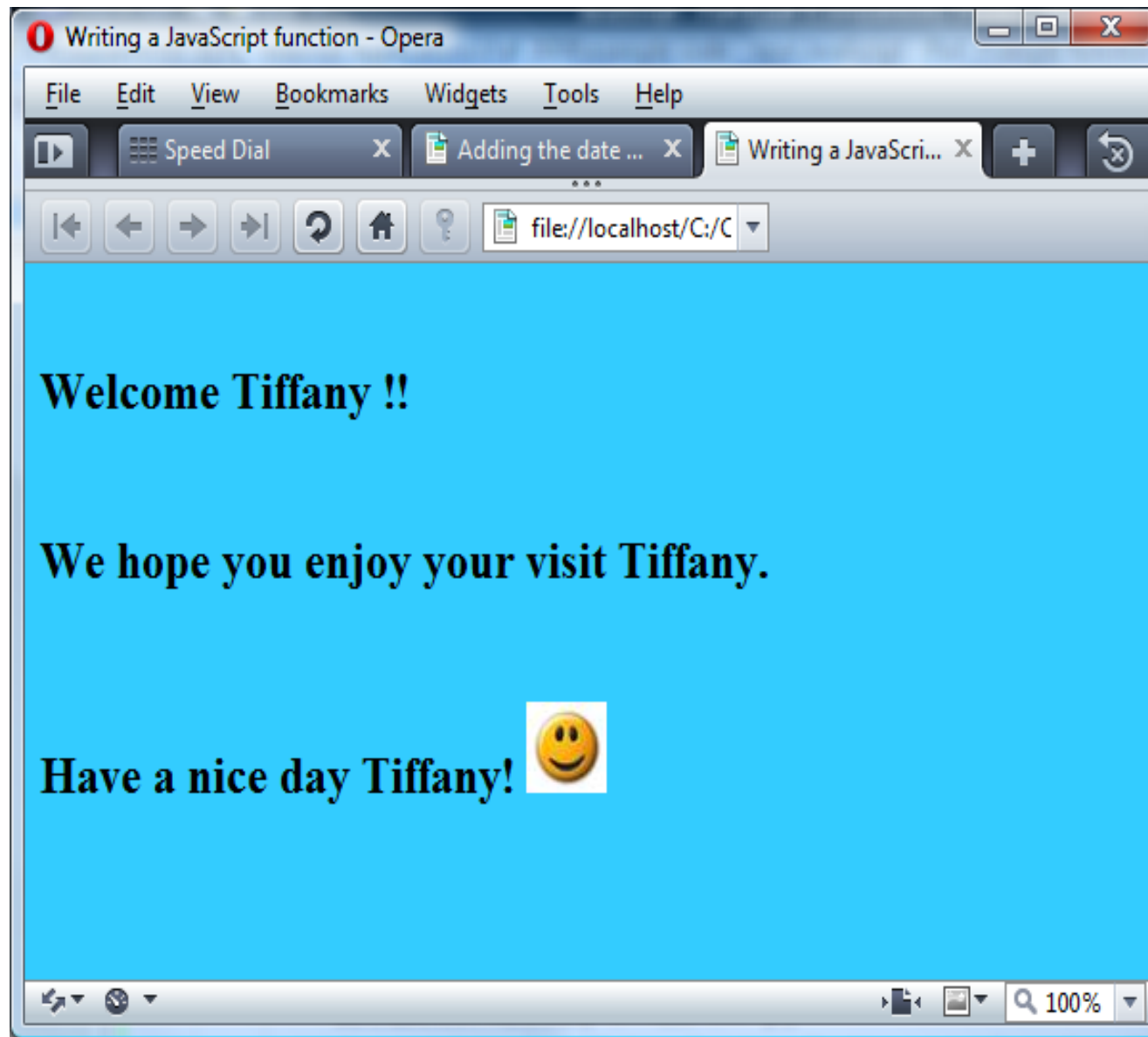
File Edit Search View Format Language Settings Macro Run TextFX Plugins Window ?

dynamicForm2.php date and time with a function.html simple function example.html using a prompt with a variable.html

```
7 <script type="text/javascript">
8     function writeTiffany() {
9         document.write("Tiffany");
10    }
11 </script>
12 <style type="text/css">
13     <!-- body {background-color: #3CF; } -->
14 </style>
15 </head>
16 <body>
17     <h2>
18     <script type="text/javascript">
19     /* <!-- [CDATA[ */
20         document.write("<br /> Welcome ");
21         writeTiffany();
22         document.write(" !!<br />");
23         document.write("<br /> <br /> We hope you enjoy your visit ");
24         writeTiffany();
25         document.write(". <br />");
26         document.write("<br /> <br /> Have a nice day ");
27         writeTiffany();
28         document.write("!");
29     /* ]] --> */
30 </script>
31 
32 </h2>
```

Hyper Text Markup nb char : 973 nb line : 34 Ln : 31 Col : 50 Sel : 0 Dos\Windows ANSI INS





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 the previous example 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 the previous example using a function with a parameter in which we will send the function the visitor's name.



C:\Courses\CGS 3175 - Internet Applications\Fall 2009\example code - new\JavaScript - Part 3\javascript function with parameters.html - Notepad++

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

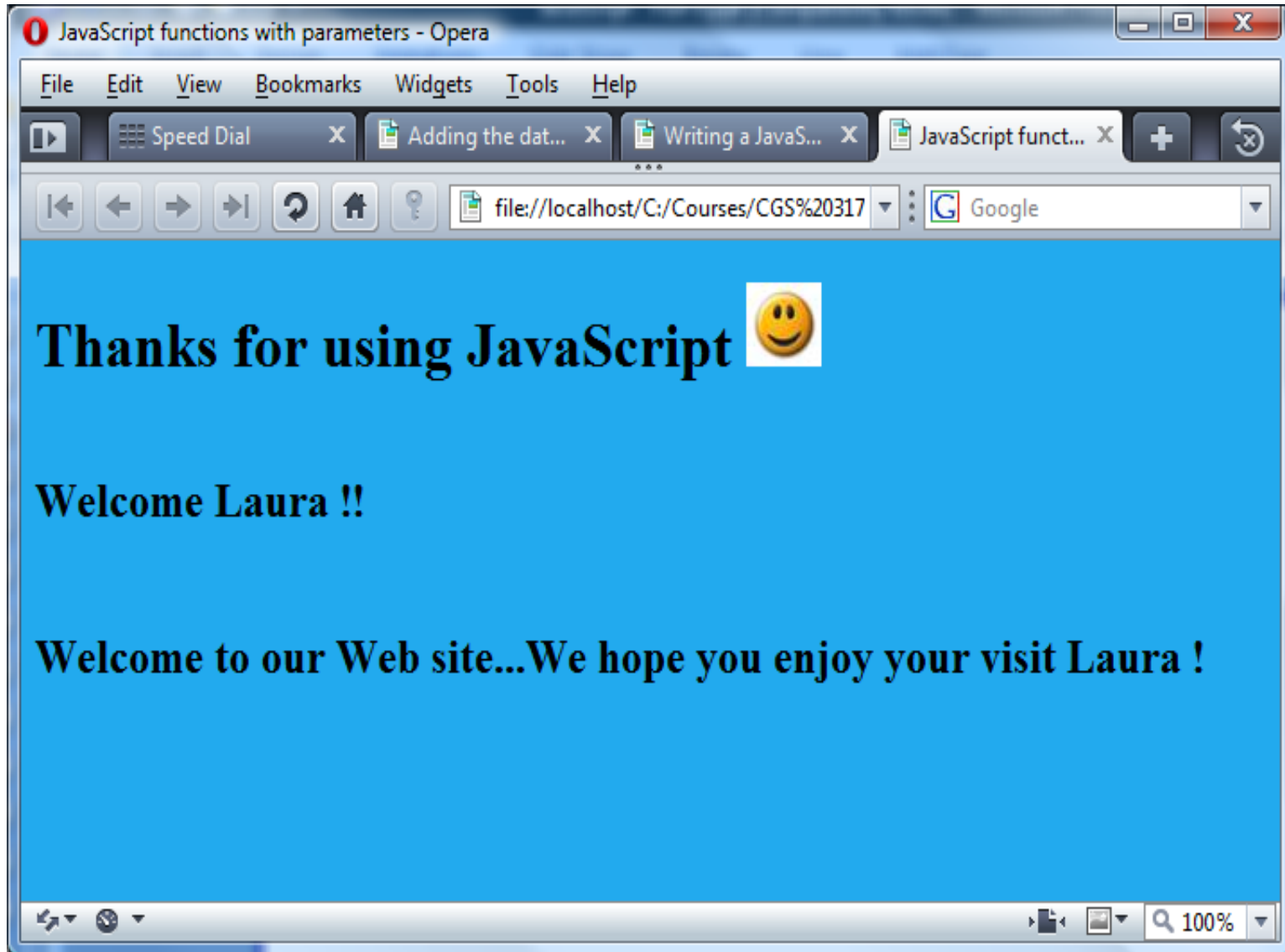
date and time with a function.html simple function example.html using a prompt with a variable.html javascript function with parameters.html

```
6 <title> JavaScript functions with parameters </title>
7 <script type="text/javascript">
8     function writeVisitorName(name) {
9         document.write(" " + name + " ");
10    }
11 </script>
12 <style type="text/css">
13     <!-- body {background-color: #2AE; } -->
14 </style>
15 </head>
16 <body>
17     <h1> Thanks for using JavaScript  </h1>
18     <h2>
19     <script type="text/javascript">
20     /* <!-- [CDATA[ */
21         var userName;
22         userName = prompt("Hi! Please tell me your name");
23         document.write("<br /> Welcome ");
24         writeVisitorName(userName);
25         document.write(" !!<br /><br /> <br />");
26         document.write("Welcome to our Web site...We hope you enjoy your visit ");
27         writeVisitorName(userName);
28         document.write("!<br />");
29     /* ]] --> */
30 </script>
31 </h2>
32 </body>
```

Hyper Text Markup Language file nb char: 1054 nb line: 34 Ln: 13 Col: 43 Sel: 0 Dos\Windows ANSI INS



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 introduce some more JavaScript features in this example.



Creating A Slide Show Using JavaScript

```
slideshow.html  slideshow.js

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>Image Slideshow</title>
7    <script type="text/javascript" src="slideshow.js"></script>
8    <style type="text/css">
9    <!--
10      body { margin: 0px 10% 0px 10%;
11        background-image: url(image1.jpg);
12      }
13    -->
14  </style>
15 </head>
16 <body style="background-color:#FFFFFF">
17   <div style="text-align:center">
18     <h1>USA Softball 2008-2009</h1>
19     
21     <h2>
22       <a href="previous.html" id="prevLink">&lt;&lt; Previous</a>
23       &nbsp;&nbsp;&nbsp;
24       <a href="next.html" id="nextLink">Next &gt;&gt;</a>
25     </h2>
26   </div>
27 </body>
```

The XHTML
document



Creating A Slide Show Using JavaScript

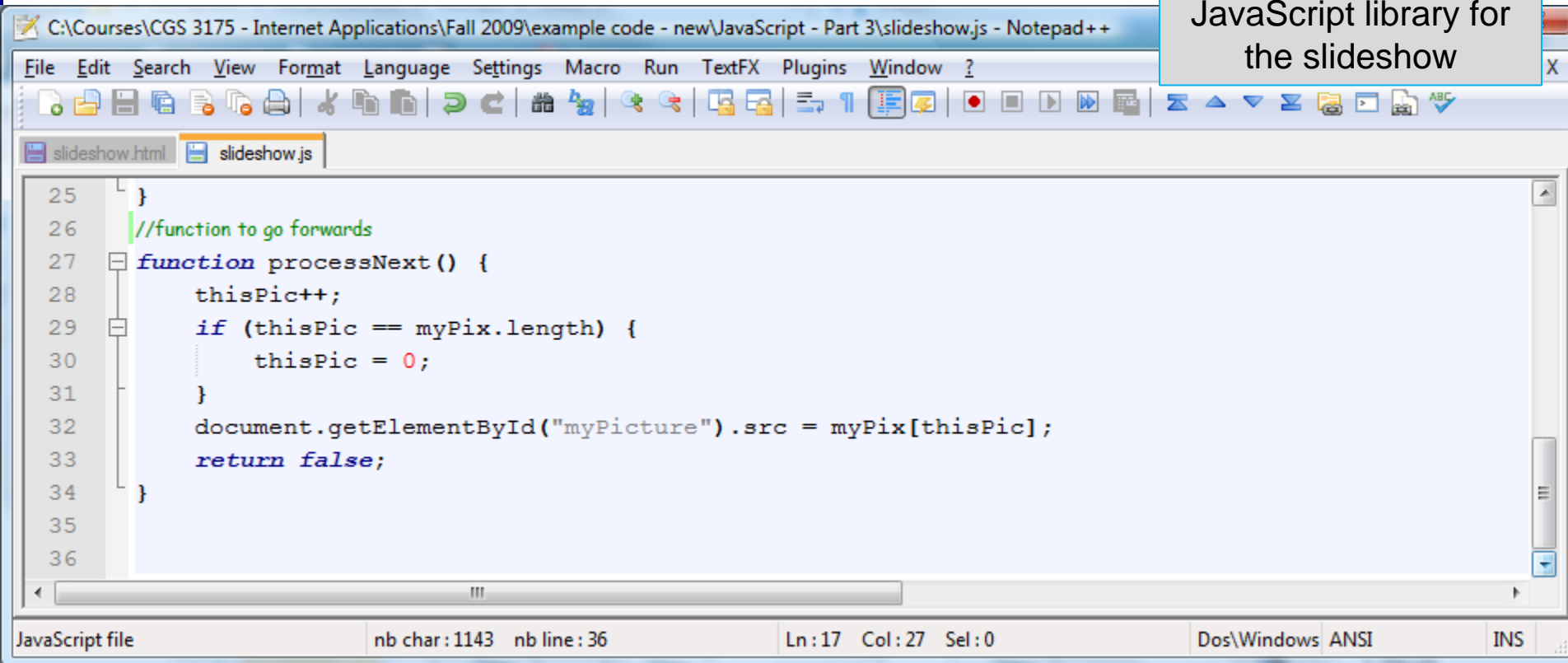
The JavaScript library
for the slideshow

```
1 //The JavaScript code for the slideshow - JavaScript - Part 3 Notes
2 //MJL November 2009
3
4 window.onload = initLinks;
5 //the loaded picture array
6 var myPix = new Array("images/team usa before game.jpg","images/Jennie Finch 2.jpg",
7                       "images/Mendosa 1.jpg", "images/Cat 1.jpg","images/Natasha Watley.jpg",
8                       "images/caitlin lowe heads to bat.jpg", "images/Lovie Jung 2.jpg",
9                       "images/Tairia Flowers.jpg", "images/mowatt 11.jpg", "images/a.jpg",
10                      "images/cat 2.jpg", "images/Mowatt 1.jpg");
11 var thisPic = 0;
12 //function to initialize the direction buttons
13 function initLinks() {
14     document.getElementById("prevLink").onclick = processPrevious;
15     document.getElementById("nextLink").onclick = processNext;
16 }
17 //function to go backwards
18 function processPrevious() {
19     if (thisPic == 0) {
20         thisPic = myPix.length;
21     }
22     thisPic--;
23     document.getElementById("myPicture").src = myPix[thisPic];
24     return false;
25 }
```



Creating A Slide Show Using JavaScript

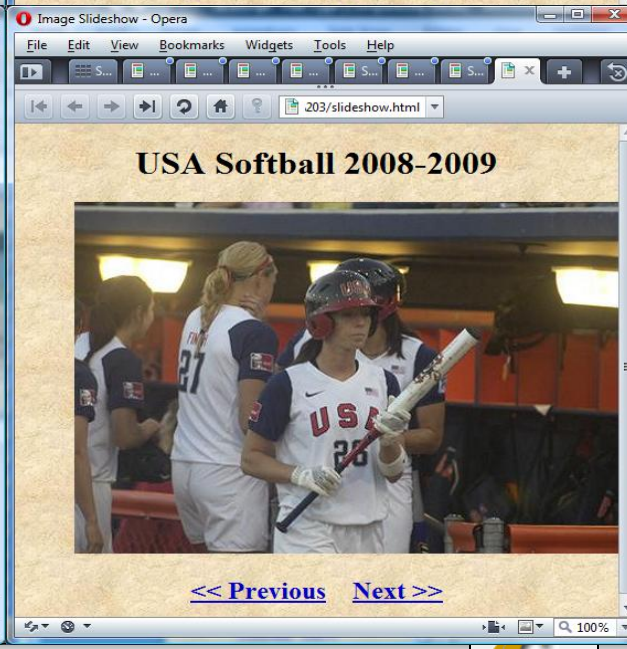
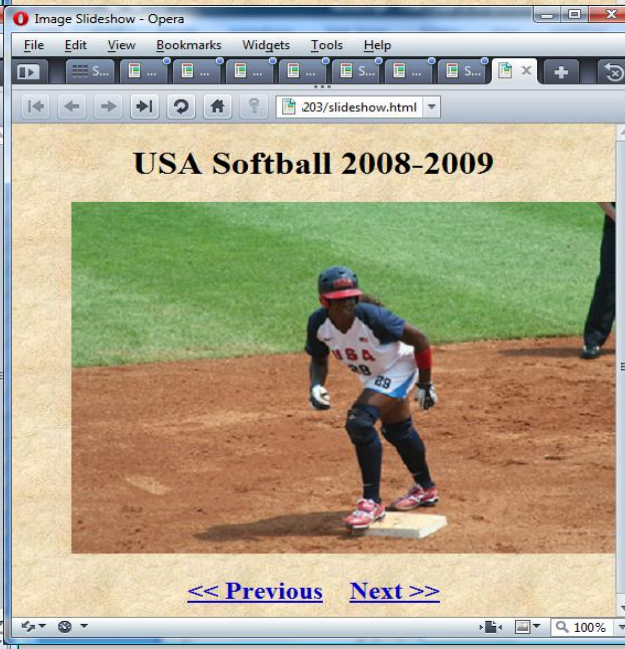
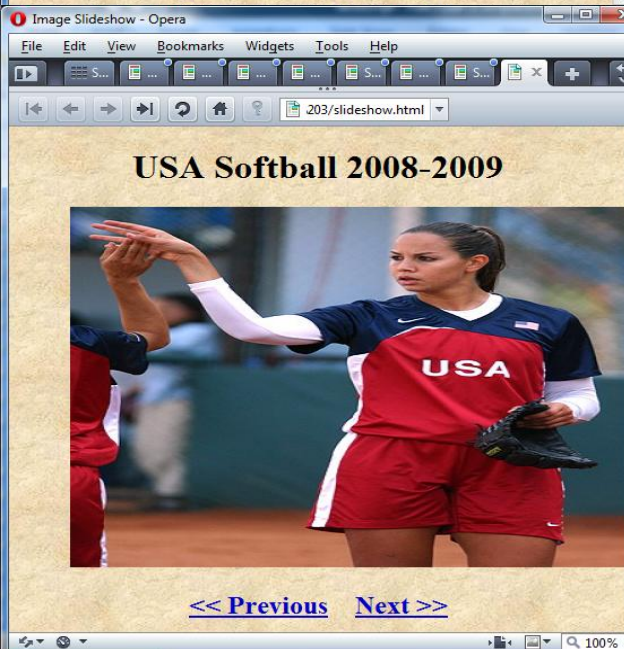
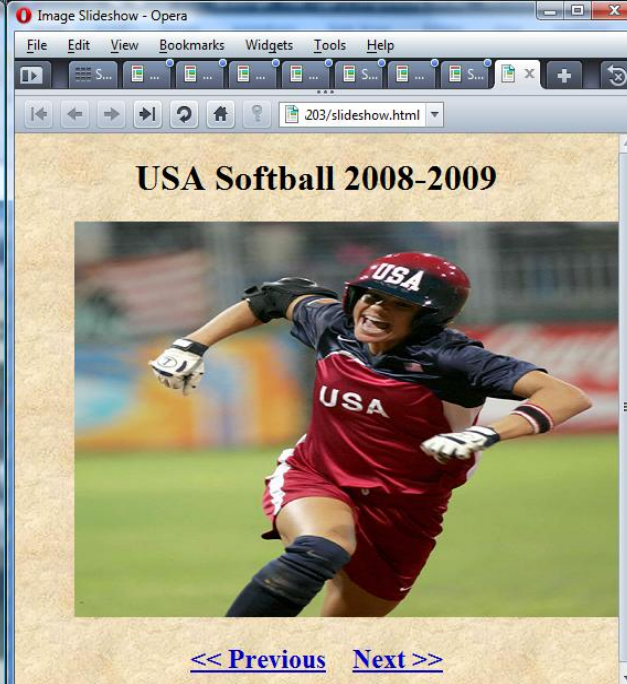
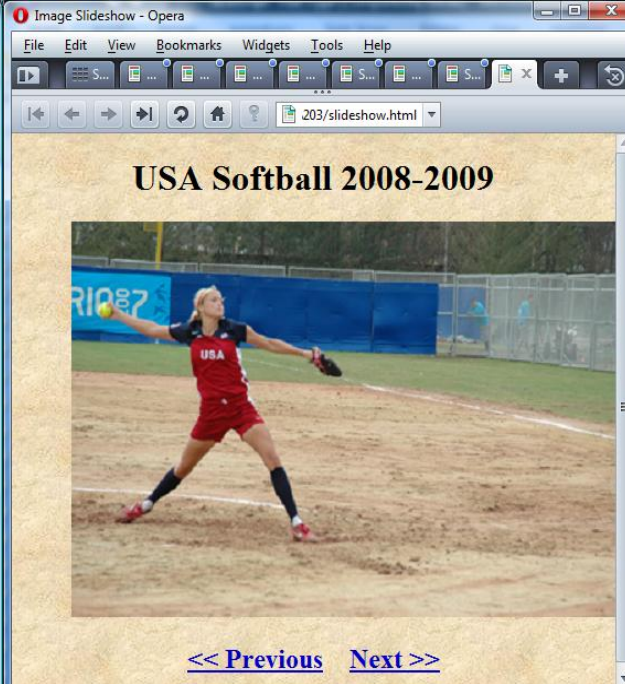
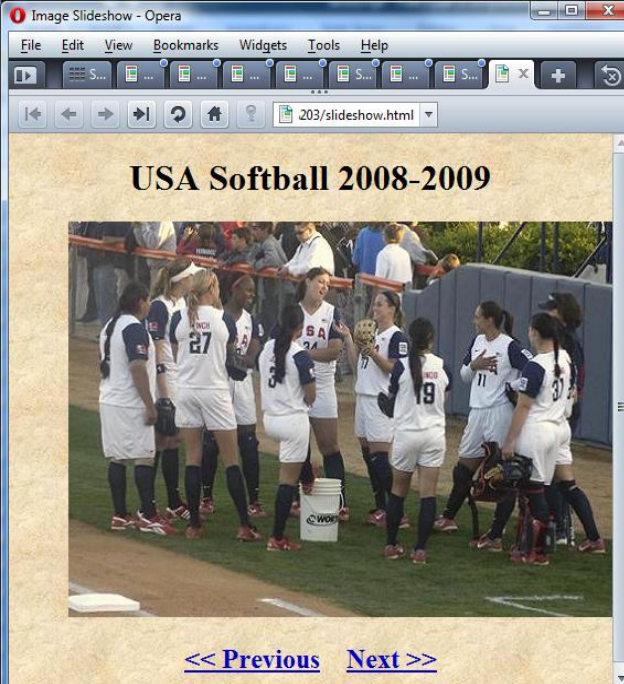
The rest of the JavaScript library for the slideshow

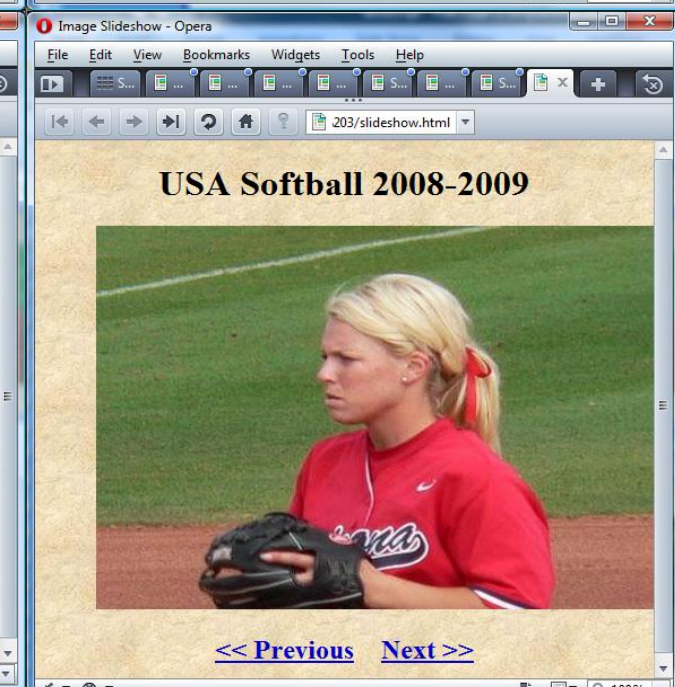
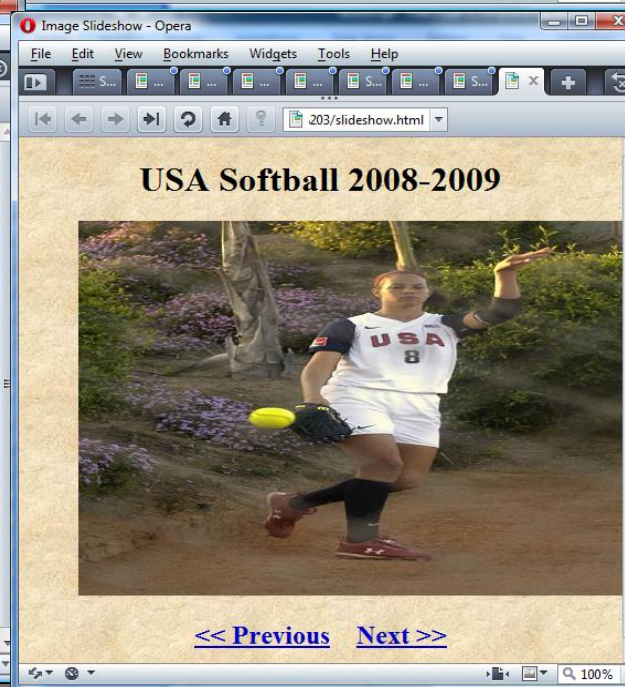
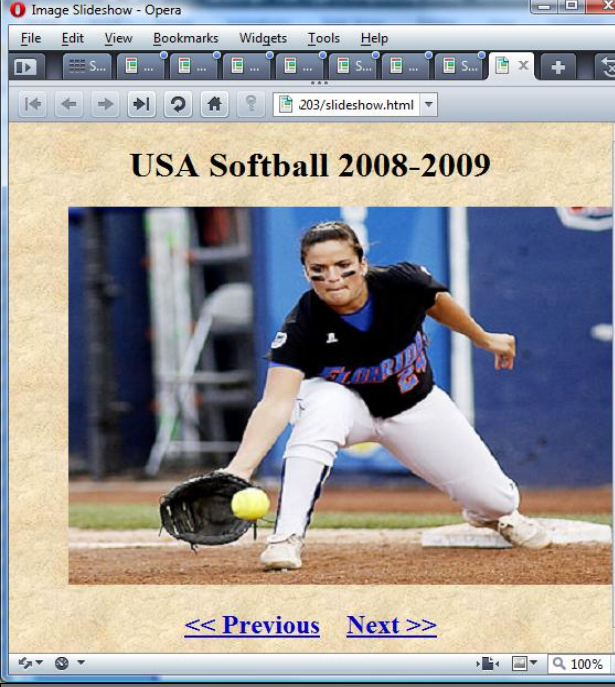
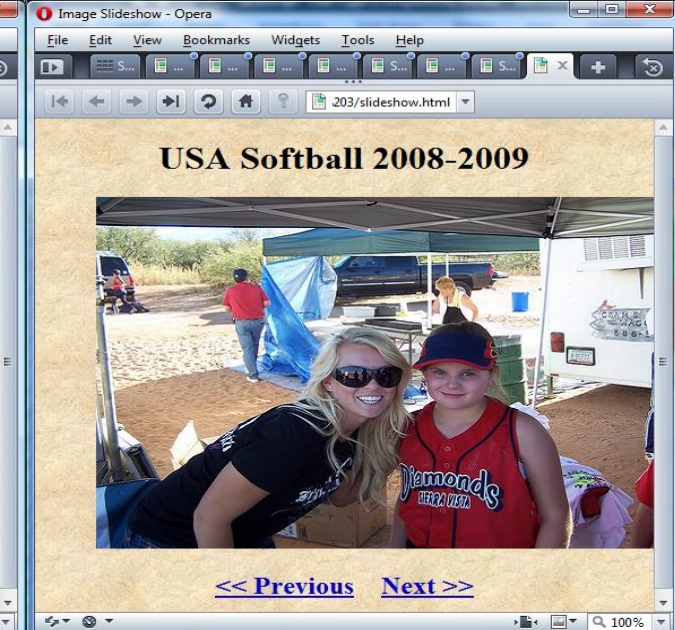
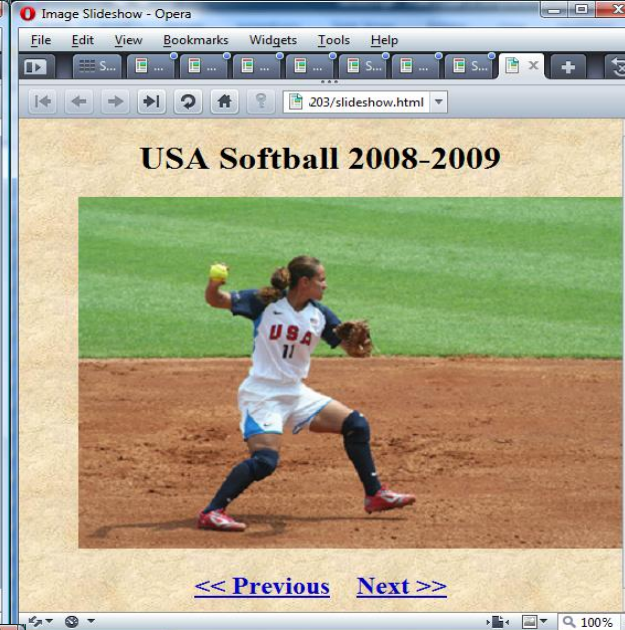
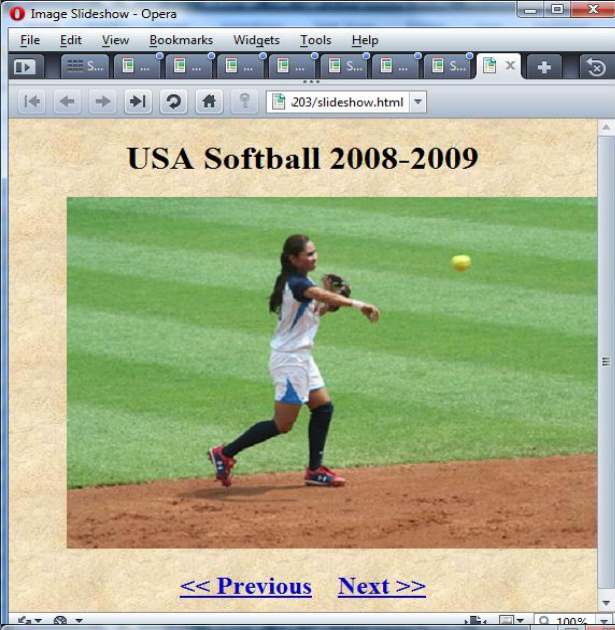


The screenshot shows a Notepad++ window with the title bar "C:\Courses\CGS 3175 - Internet Applications\Fall 2009\example code - new\JavaScript - Part 3\slideshow.js - Notepad++". The menu bar includes File, Edit, Search, View, Format, Language, Settings, Macro, Run, TextFX, Plugins, Window, and ?. The toolbar contains various icons for file operations and editing. The status bar at the bottom displays "JavaScript file", "nb char : 1143", "nb line : 36", "Ln : 17", "Col : 27", "Sel : 0", "Dos\Windows", "ANSI", and "INS".

```
25 }  
26 //function to go forwards  
27 function processNext() {  
28     thisPic++;  
29     if (thisPic == myPix.length) {  
30         thisPic = 0;  
31     }  
32     document.getElementById("myPicture").src = myPix[thisPic];  
33     return false;  
34 }  
35  
36
```







How The Slide Show Script Works

- The browser loads the `<head>` and stores twelve new image objects in an array called `myPix` 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 the page is displayed in the user's browser with the first image loaded and navigation links to the previous and next slide displayed.
- The JavaScript function `initLinks` determines which link has been “clicked” by the user, using the `onclick` intrinsic event. If the user clicked “Next” then the function `processNext` is invoked, otherwise if “Previous” is clicked, then the function `processPrevious` is invoked.
- 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.

