

CS 312

Internet Concepts

Web Programming/Scripting: Javascript

Dr. Michele Weigle

Department of Computer Science

Old Dominion University

mweigle@cs.odu.edu

<http://www.cs.odu.edu/~mweigle/CS312-F08/>

1

First Things...

- ◆ Fixing the CSS Layout example for IE
 - » Add proper DOCTYPE to css-example.html
 - ❖ forces IE to behave more according to standards
 - ❖ allows the use of a[href]:hover
 - » Being more precise with the location of the navigation section

```
#navigation {  
    position: absolute;  
    top: 125 px;  
    left: 0 px;  
    width: 200 px;  
}
```

2

First Things...

- ◆ To be standards-compliant, each HTML document should have a *document type declaration*.
 - » begins the HTML document
 - » tells a validator which version of HTML/XHTML to use in checking the syntax
- ◆ Here's the one that seems to work

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

3

First Things...

- ◆ To check how your page will render on lots of different systems
 - » <http://browsershots.org/>
- ◆ Pick the browser/OS combinations that you want
- ◆ Provide the URL
- ◆ Wait...
- ◆ Eventually (10-15 minutes), you'll have screenshots of how your page rendered on the systems

4

Javascript

- ◆ A *scripting language* designed to add interactivity to HTML page
 - » lightweight programming language
- ◆ Usually embedded directly into HTML pages
- ◆ Interpreted language
 - » needs no preliminary compilation
- ◆ Similar in syntax to Java, but **not** Java

5

Javascript Capabilities

- ◆ Gives HTML designers a programming tool
- ◆ Can be used to
 - » Put dynamic text into an HTML page
 - » React to events
 - ❖ ex: open new windows, changing images as mouse moves over
 - » Read and write HTML elements
 - » Validate entered data
 - » Detect the visitor's browser
 - » Create cookies

6

Javascript and Browsers

- ◆ Since Javascript is mainly client-side, it can be disabled at the user's browser
 - » or the browser may not support Javascript
- ◆ We need to make sure that the web page is still understandable even if Javascript is turned off
 - » or at least give the user a message if they are not using Javascript

7

Client-Side Javascript

- ◆ Interpreted by the browser upon running
 - » Displaying the page as read and executing JavaScript statements
- ◆ Can respond to user events
- ◆ Generally embedded in HTML file
- ◆ Can also be specified in separate file (with extension .js)

8

Javascript Comments

◆ Comments in HTML

`<!-- This is a comment -->`

Older browsers do not understand JavaScript.

◆ Comments in JavaScript

» Single line

`// This is a comment`

Without commenting out the JavaScript code, an older browser would display the code as regular text.

» Multiple lines

`/* Starting comment`

`Ending comment */`

9

Including Javascript

◆ For newer browsers, inside HTML, use:

`<script type="text/javascript">`

JavaScript code

`</script>`

◆ For older browsers, inside HTML, need:

`<script language="JavaScript" type="text/javascript">`

`<!-- // needed to start the entire JavaScript code`

JavaScript code

`/* comments for the code */`

JavaScript code

`// needed to end the entire JavaScript code -->`

`</script>`

◆ In separate file, use:

`<script src="scriptName.js" type="text/javascript"> </script>`

10

Programming Elements

- ◆ Declare a variable
 - » `var variableName = value;`
- ◆ Conditional
 - » `if (condition) {statements;} else {statements;}`
- ◆ Loops
 - » `while (condition) {statements;}`
 - » `do {statements;} while (condition)`
 - » `for (initial value; condition; update) {statements;}`
- ◆ Functions
 - » `function functionName (var1,var2,...,varX) {statements;}`

11

Useful Javascript Objects

- | | |
|------------|-------------|
| ◆ Document | ◆ Location |
| ◆ Date | ◆ Navigator |
| ◆ Event | ◆ Screen |
| ◆ Form | ◆ Window |

<http://www.javascriptkit.com/jsref/>

12

Useful Document Methods

- ◆ write (“*string*”)

- » write to the document as it’s loading

- ◆ writeln (“*string*”)

- » write to the document as it’s loading and insert a newline character at the end

<http://www.javascriptkit.com/jsref/document.shtml>

13

Javascript Example

Display a Message

```
<html>
<head>
<title>document.write Example 1</title>
<script type="text/javascript">
  document.write ("This is a message written when the page starts to load.");
</script>
</head>
<body>

<h1>document.write Example 1</h1>

<p>This is an example of including Javascript in the HTML header.</p>

</body>
</html>
```

<http://www.cs.odu.edu/~mweigle/cs312/js/write-head.html>

14

Javascript Example

Display a Message

```
<html>
<head>
<title>document.write Example 2</title>
</head>
<body>

<script type="text/javascript">
  document.write ("This is a message written when the page starts to load.");
</script>

<h1>document.write Example 2</h1>

<p>This is an example of including Javascript in the HTML body.</p>

</body>
</html>
```

<http://www.cs.odu.edu/~mweigle/cs312/js/write-body.html>

15

Alerts

◆ Alerts are pop-up boxes with buttons

- » `alert("sometext");`
 - ❖ Pops up a box that says *sometext*

◆ Prompts

- » `prompt("sometext", "defaultvalue");`
 - ❖ Pops up a box that asks for user input
 - ❖ If the user clicks "OK" the box returns the input value
 - ❖ If the user clicks "Cancel" the box returns null.

16

Example Function with Alert

```
<body>

<h1>Alert Example (with a Function)</h1>

<script type="text/javascript">
  function displaymessage()
  {
    alert ("Hello World!");
  }
</script>

<p>Press the button to display the message.</p>
<input type="button" value="Click Here" onclick="displaymessage()">

</body>
```

*<html>, <head>
tags omitted to
save space*

function definition

function call

<http://www.cs.odu.edu/~mweigle/cs312/js/alert.html>

17

Example Function with Prompt

```
<head><title>Prompt Example</title>
<script type="text/javascript">
  function display_prompt()
  {
    var name = prompt ("Please enter your name","Harry Potter");
    return name;
  }
</script>
</head>

<body>
<h1>Prompt Example (with a Value-Returning Function)</h1>

<script type="text/javascript">
  document.write ("<h3>Welcome " + display_prompt() + "</h3>");
</script>

</body>
```

*<html> tags
omitted to
save space*

<http://www.cs.odu.edu/~mweigle/cs312/js/prompt.html>

18

Useful Document Properties

- ◆ URL
 - » returns URL of the current document
- ◆ lastModified
 - » returns date and time document was last modified
- ◆ bgColor
 - » specifies the background color of the document
- ◆ links[]
 - » an array containing all of the links on the page
- ◆ referrer
 - » string that specifies the URL of the page that contained the link to the current page

<http://www.javascriptkit.com/jsref/document.shtml>

19

Example

Greeting According to Time of Day

```
<script type="text/javascript">
  var d = new Date();
  var hrs = d.getHours();

  if (hrs < 12) {
    document.bgColor = "yellow";
    document.write ("<h3>Good morning!</h3>");
  } else {
    document.bgColor = "green";
    document.write ("<h3>Good day!</h3>");
  }
</script>

<p>The current time is
<script type="text/javascript">document.write (new Date().toLocaleTimeString());
</script> </p>
```

<http://www.cs.odu.edu/~mweigle/cs312/js/greeting.html>

20

Javascript Event Handlers

- ◆ Allows a Javascript function to be executed when the user does something
- ◆ onclick
 - » when the mouse clicks an object
- ◆ onload
 - » when the page or image is finished loading
- ◆ onunload
 - » when the user leaves a page
- ◆ onmouseover
 - » when the mouse is moved over an element

http://www.w3schools.com/jsref/jsref_events.asp

21

Example

Alert on Leaving a Page

```
<html>
<head>
<script type="text/javascript">
  function bye()
  {
    alert ("Bye! Come back soon!");
  }
</head>
```

```
<body onunload="bye()">
This page will tell you "Bye!" when you
<a href="../index.html">leave</a>.
</body>
```

<http://www.cs.odu.edu/~mweigle/cs312/js/bye.html>

22

Example

Alert on Click, Using External File

```
<head>
<script type="text/javascript" src="externalFunction.js">
</script>
</head>

<body>
<p onclick="start()">
If you click on this paragraph, then it will call an external script
function named "start" in the file named "externalFunction.js".
Click anywhere in this paragraph and you'll get an alert.</p>

<p>Another paragraph, click and no alert.</p>
</body>
```

23

Example

Alert on Click, Using External File

<http://www.cs.odu.edu/~mweigle/cs312/js/click.html>

externalFunction.js:

```
function start()
{
    alert ("Hello, glad you clicked!");
}
```

*externalFunction.js must be in the same directory as the HTML file that uses it, or the directory must be specified in the HTML file (src attribute) **and** must be chmod 644.*

24

Passing Arguments to a Function

Javascript and Forms

```
<script type="text/javascript">
function userAlert (label) {
    alert (label.value);
}
</script>
```

```
<form>
Enter alert text: <input type="text" name="alertText">
<br>
<input type="button" value="Issue Alert"
onclick="userAlert (alertText)">
</form>
```

<http://www.cs.odu.edu/~mweigle/cs312/js/passing-args.html>

25

Using Javascript in Page Design

- ◆ Since you can't put HTML code in a CSS file, how can you get a standard header on a set of webpages?
 - » Create an external Javascript file that writes out the header HTML.
 - » Call the Javascript in the header part of each page.
 - » If you want to change the header, just change the Javascript file.

26

Using Javascript in Page Design

◆ Example:

- » <http://www.cs.odu.edu/~mweigle/may08-website/>
- » <http://www.cs.odu.edu/~mweigle/may08-website/header.js>
- » <http://www.cs.odu.edu/~mweigle/may08-website/footer.js>

```
<script type="text/javascript" src="header.js"></script>
...
<script type="text/javascript" src="footer.js"></script>
```

27

Using Javascript in Page Design

header.js and footer.js

```
<!--
document.writeln ('<table cellpadding=10>');
document.writeln ('<tr>');
document.writeln ('<td>');
// -->
```

28

Exercise: Create Countdown Clock

Useful Objects/Methods

- ◆ Target date
 - » `target = new Date (string);`
 - » format: “mm/dd/yyyy hh:mm AM/PM”
 - » example: `new Date (“12/11/2008 12:30 PM”)`
- ◆ Current date/time
 - » `now = new Date();`
- ◆ Date differences
 - » `new Date (target – now);`
 - » difference in milliseconds

29

Exercise: Create Countdown Clock

Useful Objects/Methods

- ◆ Write text to a certain section
 - » If we have in our HTML
`<div id=“name”> </div>`
 - » We can use in our Javascript
`document.getElementById (“name”).innerHTML = “mytext”;`
- ◆ Math functions
 - » `Math.floor (3/2)`
 - ❖ will return 1
- ◆ Calling functions repeatedly
 - » `setInterval (“function() ”, milliseconds);`

30

Exercise: Create Countdown Clock

Setup HTML

```
<p>We're counting down until the CS 312 final exam  
(Dec 11, 2008 @ 12:30pm).</p>
```

```
<div id="examCountdown" style="background-color:  
blue; color: yellow; font-weight: bold; text-align:  
center"></div>
```

31

Exercise: Create Countdown Clock

Setup Javascript

```
<script type="text/javascript">  
// set the target date and time  
var examDate = new Date ("12/11/2008 12:30 PM");  
  
/* call the function to display the time left  
until examDate in the area labeled 'examCountdown' */  
displayTimeLeft (examDate, "examCountdown");  
  
// call the function every 1000 ms (every 1 second)  
setInterval ("displayTimeLeft(examDate,  
    'examCountdown')", 1000);
```

<http://www.cs.odu.edu/~mweigle/cs312/js/countdown.html>

32

Exercise: Create Countdown Clock

Setup Javascript displayTimeLeft function

```
function displayTimeLeft (date, label) {  
    var now = new Date();  
    var diff = new Date (date – now); // in ms  
    var secs = Math.floor (diff.valueOf()/1000);  
  
    if (secs < 0) {  
        document.getElementById(label).innerHTML =  
        “It’s already past!”;  
        return;  
    }  
}
```

<http://www.cs.odu.edu/~mweigle/cs312/js/countdown.html>

33

Exercise: Create Countdown Clock

Setup Javascript displayTimeLeft function

```
// calculate days, hours, minutes, seconds  
var days = Math.floor (secs / 86400);  
  
var leftover = secs % 86400;  
  
var hours = Math.floor (leftover / 3600);  
  
leftover = leftover % 3600;  
  
var mins = Math.floor (leftover / 60);  
  
leftover = leftover % 60;
```

<http://www.cs.odu.edu/~mweigle/cs312/js/countdown.html>

34

Exercise: Create Countdown Clock

Setup Javascript displayTimeLeft function

```
// setup string to display
var display = days + “ days, ” + hours + “ hours, ” +
    mins + “ minutes, ” + leftover + “ seconds”;

// set the text of the countdown element
document.getElementById(label).innerHTML =
    display;
}

</script>
```

<http://www.cs.odu.edu/~mweigle/cs312/js/countdown.html>