

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-F10/>

1

Outline

- ◆ What is JavaScript?
- ◆ Document Properties
- ◆ JavaScript Syntax
- ◆ Event Handlers
- ◆ Built-in JavaScript Objects
- ◆ Exercise: Creating a Countdown Clock
- ◆ Pop-Ups and Functions

2

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

3

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

4

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

5

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)

6

Outline

- ◆ What is JavaScript?
- ◆ Document Properties
- ◆ JavaScript Syntax
- ◆ Event Handlers
- ◆ Built-in JavaScript Objects
- ◆ Exercise: Creating a Countdown Clock
- ◆ Pop-Ups and Functions

7

JavaScript Comments

- ◆ Comments in HTML
 - <!-- This is a comment -->
- ◆ Comments in JavaScript
 - » Single line
 - // This is a comment
 - » Multiple lines
 - /* Starting comment
 - Ending comment */

Older browsers do not understand JavaScript.

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

8

Including JavaScript

- ◆ JavaScript inside HTML (for XHTML validation), use:

```
<script type="text/javascript">  
  <!-- // needed to start the entire JavaScript code  
    JavaScript code  
  /* comments for the code */  
    JavaScript code  
  // next line needed to end the entire JavaScript code  
  -->  
</script>
```

- ◆ JavaScript in a separate file, use:

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

9

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; }`

10

Outline

- ◆ What is JavaScript?
- ◆ JavaScript Syntax
- ◆ Built-in JavaScript Objects
- ◆ Pop-Ups and Functions
- ◆ Document Properties
- ◆ Event Handlers
- ◆ Exercise: Creating a Countdown Clock

11

Useful JavaScript Objects

- ◆ Document
- ◆ Date
- ◆ Event
- ◆ Form
- ◆ Location
- ◆ Navigator
- ◆ Screen
- ◆ 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
- ◆ getElementById (“*string*”)
 - » access any element on the page via its *id* attribute
 - » can alter the element using .innerHTML

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

13

JavaScript Example

Display a Message – Script in <head>

```
<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>
```

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

<?xml>, <!DOCTYPE>,
<html> tags omitted to
save space

14

JavaScript Example

Display a Message - Script in <body>

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

15

JavaScript Example

Display a Message – using getElementById

```
<head>
<title>getElementById Example</title>
</head>
<body>

<h1>getElementById Example</h1>

<p>This is an example of using getElementById to add text.</p>

<div id="target"></div>

<script type="text/javascript">
  <!-- document.getElementById("target").innerHTML = "<em>This
text is written by the script.</em>";
  // -->
</script>

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

Named element has to
appear before
getElementById

16

Outline

- ◆ What is JavaScript?
- ◆ Document Properties
- ◆ JavaScript Syntax
- ◆ Event Handlers
- ◆ Built-in JavaScript Objects
- ◆ Exercise: Creating a Countdown Clock
- ◆ Pop-Ups and Functions

17

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.

18

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>
```

function definition

function call

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

19

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>
```

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

20

Passing Arguments to a Function

JavaScript and Forms

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

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

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

21

JavaScript and Forms

Accessing Form Data

- ◆ `document.forms[i].inputname.value` – is a String
 - » what the user typed in the box named *inputname* in the form with index *i* (starting at 0)

```
<form action="">
  <label>Enter name: <input type="text" name="myName" /> </label>
</form>
```

`document.forms[0].myName.value`
is the String that the user types in the box

22

JavaScript and Forms

Validating Form Data

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

```
<!--
```

```
function validateForm() {
```

```
    if (document.forms[0].email.value == "") {
```

```
        alert ("You must enter an email address.");
```

```
    } else if (document.forms[0].email.value.search("@") == -1) {
```

```
        alert ("You entered an invalid email address.");
```

```
    }
```

```
} // -->
```

```
</script>
```

If the search text does not appear in the string, -1 will be returned.

What do we need to add to make the alert show the entered text if it's OK?

```
<form action="">
```

```
<label>Email: <input type="text" name="email" /> </label>
```

```
<br />
```

```
<input type="button" value="Submit" onclick="validateForm()" />
```

```
</form>
```

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

23

Outline

◆ What is JavaScript?

◆ Document Properties

◆ JavaScript Syntax

◆ Event Handlers

◆ Built-in JavaScript Objects

◆ Exercise: Creating a Countdown Clock

◆ Pop-Ups and Functions

24

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>

25

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>
```

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

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

26

Outline

- ◆ What is JavaScript?
- ◆ Document Properties
- ◆ JavaScript Syntax
- ◆ Event Handlers
- ◆ Built-in JavaScript Objects
- ◆ Exercise: Creating a Countdown Clock
- ◆ Pop-Ups and Functions

27

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

28

Example

Alert on Leaving a Page

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

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

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

29

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>
```

30

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.*

31

Exercise

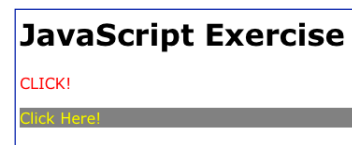
- ◆ Write JavaScript and XHTML code that meets the following requirements:

- » page title is “JavaScript Exercise”
- » level 1 heading “JavaScript Exercise”
- » create a div with the id “target” and red text
- » create a paragraph with a gray background and yellow text that reads “Click Here!”
 - ❖ when the users clicks in the paragraph, the JavaScript function click() should be called
- » the click() function should write the word “CLICK!” in the target div

Before:



After:



32

Outline

- ◆ What is JavaScript?
- ◆ Document Properties
- ◆ JavaScript Syntax
- ◆ Event Handlers
- ◆ Built-in JavaScript Objects
- ◆ Exercise: Creating a Countdown Clock
- ◆ Pop-Ups and Functions

33

Exercise: Create Countdown Clock

- ◆ We're going to create a page that has a countdown timer, updating the time until our final exam.

Countdown Clock

We're counting down until the CS 312 final exam (Dec 16, 2010 @ 8:30am).

34 days, 18 hours, 11 minutes, 35 seconds

34

Exercise: Create Countdown Clock

Useful Objects/Methods

- ◆ Target date
 - » `target = new Date (string);`
 - » format: “mm/dd/yyyy hh:mm AM/PM”
 - » example: `new Date (“12/16/2010 8:30 AM”)`
- ◆ Current date/time
 - » `now = new Date();`
- ◆ Date differences
 - » `new Date (target – now);`
 - » difference in milliseconds

35

Exercise: Create Countdown Clock

Useful Objects/Methods

- ◆ Write text to a certain section
 - » `getElementById("name").innerHTML`
- ◆ Math functions
 - » `Math.floor (3/2)`
 - ❖ will return 1
- ◆ Calling functions repeatedly
 - » `setInterval (“function()”, milliseconds);`

36

Exercise: Create Countdown Clock

Setup HTML

```
<p>We're counting down until the CS 312 final exam  
(Dec 16, 2010 @ 8:30am).</p>
```

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

37

Exercise: Create Countdown Clock

Setup JavaScript

```
<script type="text/javascript">  
<!--  
// set the target date and time  
var examDate = new Date ("12/17/2009 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>

38

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>

39

Exercise: Create Countdown Clock

Setup JavaScript displayTimeLeft function

```
// calculate days, hours, minutes, seconds  
var days = Math.floor (secs / 86400); // 86,400 secs / day  
var leftover = secs % 86400;  
  
var hours = Math.floor (leftover / 3600); // 3,600 secs / hour  
leftover = leftover % 3600;  
  
var mins = Math.floor (leftover / 60); // 60 secs / min  
leftover = leftover % 60;
```

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

40

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;

} // end of function
// -->
</script>
```

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

41

Outline

- ◆ What is JavaScript?
- ◆ Document Properties
- ◆ JavaScript Syntax
- ◆ Event Handlers
- ◆ Built-in JavaScript Objects
- ◆ Exercise: Creating a Countdown Clock
- ◆ Pop-Ups and Functions

42