

# **Dynamic Web Pages Client Side Events JavaScript**

**Jeff Offutt**

**<http://www.cs.gmu.edu/~offutt/>**

**SWE 432  
Design and Implementation of  
Software for the Web**

## **Overview**

1. Overview of HTML
2. Styling and Cascading
3. Dynamic Event Handling with JavaScript

Each Browser Handles These Differently

## Why Learn HTML?

- Many HTML editors are available
- We can create reasonably good HTML without knowing the language
- But ...
  - We need to know the language to write excellent HTML
  - Editors get in the way of small changes
  - We need to write programs that create HTML
- Thus, a web software engineer needs to know HTML

11 October 2011

© Offutt, 2011

3

## HTML Basic Definitions

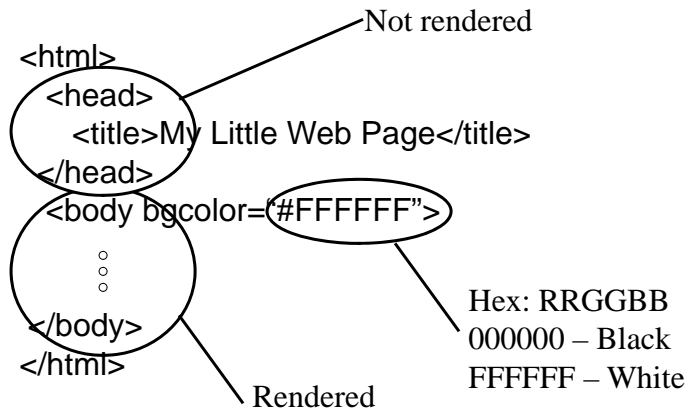
- *Element* : A piece of a document sentence, paragraph, list, table, head, ...
- *Tag* : A command to control the format  
`<name>`    { Not case sensitive  
`</name>`    { Some people use upper case for tags as a matter of style
- *Attribute* : An option or parameter to a tag
- *Rendering* : A browser formats the text into a window according to the formatting rules

11 October 2011

© Offutt, 2011

4

# HTML Structure



11 October 2011

© Offutt, 2011

5

## Some HTML Tags

- Headers : `<h1>`, `<h2>`, ...
- Breaks : `<p>`, `<br>`, `<hr>`
- Fonts : `<b>`, `<em>`, `<i>`, `<tt>`, `<u>`
- Lists : `<ul><li>`, `<ol><li>`,  
`<dl><dt>A<dd>stuff`
- Color : `<font color="#RRGGBB" size="-1">`
- Special chars : `&lt;`; `&gt;`; `&amp;`; `&nbsp;`;  
`&ouml;` (ö) `&ntilde;` (ñ) `&grave;`; (È)

Note that `<font>` is deprecated in HTML 4 and not supported in XHTML, but still widely used

11 October 2011

© Offutt, 2011

6

## Links in HTML

- `<a href="classes/432/">432</a>`
- Relative links are:
  - Faster
  - Easier to move
  - Require less typing, thus less mistakes
- `target="_blank" // new window`
- `href="#NAME" // within the current file`
- `<a href="mailto:offutt@gmu.edu">offutt</a>`
- ``

11 October 2011

© Offutt, 2011

7

## HTML Tables

```
<table>
  <tr>
    <td></td>
    <td></td>
  </tr>
  <tr>
  </table>
```

```
<table border=2 cellspacing=2
  bgcolor="#FFFFFF" width=500 align="center">
```

Note that bgcolor and align are deprecated in HTML 4,  
but still widely used

11 October 2011

© Offutt, 2011

8

## HTML Forms

`<form action="URL of program" method="post">`

`<input type = {text, radio, checkbox, file, hidden}>`

`<label>`

`<select>, <option>`

`<textarea>`

`<button>`

`<input type = {Submit, Reset}>`

`<fieldset>`

`<legend>`

11 October 2011

© Ofutt, 2011

9

## A Small Form

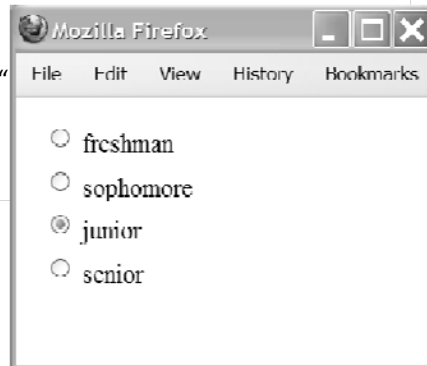
```
<form method="post" action="http://cs.gmu.edu:8080/offutt/servlet/calculate">
<table border=1><!-- outer table -->
<tr> <td>
<table align="center">
<tr>
<td>First val:
<td><input type="text" name="LHS" value="" size=5>
</tr>
<tr>
<td>Second val:
<td><input type="text" name="RHS" value="" size=5>
</tr>
<tr>
<td>Result:
<td><input type="text" name="RSLT" value="" size=6>
</tr>
</table>
</td>
<tr>
<td><input type="submit" name="Operation" value="Add">
<td><input type="submit" name="Operation" value="Subtract">
<td><input type="submit" name="Operation" value="Multiply">
<td><input type="submit" name="Operation" value="Divide">
</td>
<tr>
<td align=center><input type="reset" name="reset" value="Reset">
</td>
</tr>
</table><!-- outer table -->
</form>
```



10

## Radio Buttons

```
<form>
  <input type="radio" name="Year" id="freshman" value="freshman" />
  <label for="freshman">freshman</label>
  <br/>
  <input type="radio" name="Year" id="sophomore" value="sophomore" />
  <label for="sophomore">sophomore</label>
  <br/>
  <input type="radio" name="Year" id="junior" value="junior" />
  <label for="junior">junior</label>
  <br/>
  <input type="radio" name="Year" id="senior" value="senior" />
  <label for="senior">senior</label>
  <br/>
</form>
```



11 October 2011

© Offutt, 2011

11

## Checkboxes

```
<form>
  I have a bike:
  <input type="checkbox" name="vehicle" value="Bike" />
  <br>
  I have a car:
  <input type="checkbox" name="vehicle" value="Car" />
  <br>
  I have an airplane:
  <input type="checkbox" name="vehicle" value="Airplane" />
</form>
```



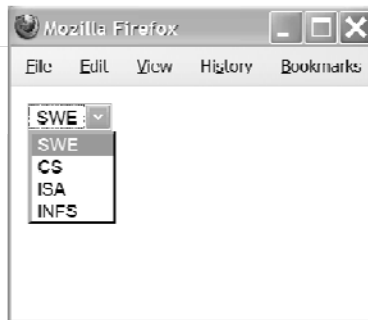
11 October 2011

© Offutt, 2011

12

## Dropdown

```
<form>
  <select name="major">
    <option value="SWE" selected="selected">SWE</option>
    <option value="CS">CS</option>
    <option value="ISA">ISA</option>
    <option value="INFS">INFS</option>
  </select>
</form>
```



11 October 2011

© Offutt, 2011

13

## Select Box

```
<form>
  <select name="classes" size="5" multiple="multiple">
    <option value="619">619</option>
    <option value="620">620</option>
    <option value="621">621</option>
    <option value="622">622</option>
    <option value="623">623</option>
    <option value="625">625</option>
    <option value="626">626</option>
    <option value="630">630</option>
    <option value="632">632</option>
    <option value="637">637</option>
    <option value="641">641</option>
    <option value="642">642</option>
    <option value="645">645</option>
    <option value="699">699</option>
  </select>
</form>
```



11 October 2011

© Offutt, 2011

14

## Text Area

```
<p align="justify">
```

Enter your message in the box below:

```
</p>
```

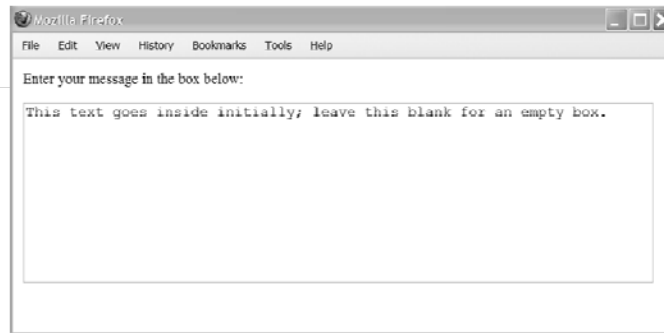
```
<form>
```

```
<textarea rows="10" cols="70">
```

This text goes inside initially; leave this blank for an empty box.

```
</textarea>
```

```
</form>
```



11 October 2011

© Offutt, 2011

15

## Buttons

```
<form>
```

```
<input type="button" value="Don't click me!" />
```

```
</form>
```



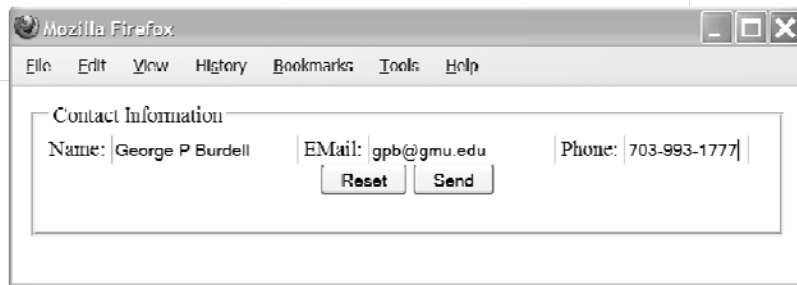
11 October 2011

© Offutt, 2011

16

## Fieldset Borders and Mailto

```
<fieldset>
<legend>
  Contact Information
</legend>
<form action="mailto:offutt@gmu.edu" method="post" enctype="text/plain">
  Name: <input type="text" name="theName">
  EMail: <input type="text" name="EMail">
  Phone: <input type="text" name="Phone" maxlength="12" size="12">
<center>
  <input type="reset" value="Reset">
  <input type="submit" value="Send">
</center>
</form>
</fieldset>
```



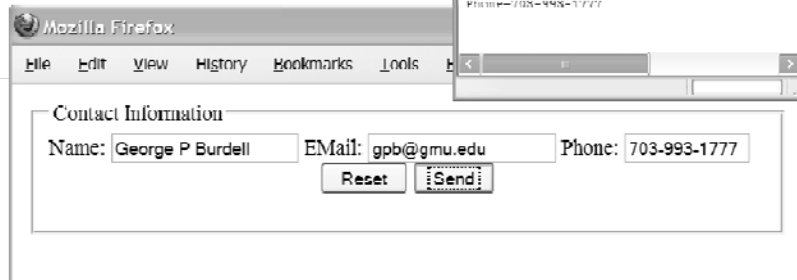
11 October 2011

© Offutt, 2011

17

## Fieldset Borders and Mailto

```
<fieldset>
<legend>
  Contact Information
</legend>
<form action="mailto:offutt@gmu.edu" method="post" e...
  Name: <input type="text" name="theName">
  EMail: <input type="text" name="EMail">
  Phone: <input type="text" name="Phone" maxlength="1...
<center>
  <input type="reset" value="Reset">
  <input type="submit" value="Send">
</center>
</form>
</fieldset>
```



11 October 2011

© Offutt, 2011

18

## HTML Suggestions

- Don't build: Borrow, modify, experiment
- Maintenance is crucial: Make your HTML readable
  - Choose a style and stick with it
  - Use white space and carriage returns
  - Use comments liberally
  - Indent lists and nested elements
- Let the browsers make as many decisions as possible when they render your HTML
  - Don't over-specify the fonts
- Look at your HTML with different browsers

11 October 2011

© Offutt, 2011

19

## HTML Suggestions (2)

- Always use descriptive titles
- Specifying the height and width of images will help browsers display faster
  - But may not work as well on different size browser windows
- Use ALT on every image
- Forms:
  - Be sure to use unique descriptive names
  - Radio buttons must have the same name or users can select multiple buttons
- Close all tags (SGML standard)

11 October 2011

© Offutt, 2011

20

# Overview

1. Overview of HTML
2. Styling and Cascading
3. Dynamic Event Handling with JavaScript

## Styling and Cascading

- Original HTML used attributes and tags to control the style of text and objects
  - `<font color="#RRGGBB" size="-1">`
- Style Sheets and Cascading Style Sheets (CSS) were added to allow for more abstraction
  - The same style commands can be used in different places in a document
  - The same style commands can be applied to different documents

## Some Style Commands

- Size : `font-size:110%`
- Color: `color:#900000; background-color:#FFFF00`
- Font: `font-weight:bold; font-style:italic; text-decoration:underline; font-family:helvetica`
- Many many more ...
  - <http://www.yourhtmlsource.com/stylesheets/introduction.html>
  - <http://www.w3.org/Style/Examples/011/firstcss>

## Embedding Style in HTML

- The HTML style attribute can be inserted into various HTML tags
  - `<span style="color:red">Do they vote?`
  - `<a href="/index.html" style="text-decoration:none;color:blue">`
  - `<span style="color:#800000; font-size:60%; font-style:italic">`
  - `<p style="text-align:justify; font-style:italic; font-family:comic sans ms; font-size:85%">`

## Defining Style Classes in HTML

- In the head:

```
<head>
  <title>...
  <style>
    P{text-align:justify}
    P.menu{font-size:75%;color:#555555;background-color:E4F4E4}
    A:hover {background:white}
  </style>
```

- In the body:

```
<p class="menu">
  <a href="#CLASSES">Classes</a> *
  <a href="#RSRCH">Research</a>
</p>
```

11 October 2011

© Offutt, 2011

25

## Defining Style in a .CSS File

```
/* For use in 432 web site pages. */
P {text-align:justify}
LI {text-align:justify}
A:hover {background:white}
.ans {color:red}
DIV.sol {color:red; font-size:80%}
SPAN.sol {color:red; font-size:80%}
.noteI {color:red}
.noteII {color:red; font-style:italic}
.noteIII {color:red; font-weight:bold; font-style:italic}
.head {font-size:150%; font-weight:bold}
TD.but {text-align:center; color:black; background:pink; font-size:70%}
TD.topic {font-weight:bold}
.commands {font-family:helvetica; color:black; font-weight:bold; font-size:90%}
.header {font-size:115%; font-weight:bold; text-decoration:underline}
```

File: 432-style.css

11 October 2011

© Offutt, 2011

26

## Using Style from a .CSS File

- In the head:

```
<head>  
  <link rel="stylesheet" href="/432-style.css"  
  type="text/css" />
```

- In the body:

```
<div class="head">  
<span class="note1">  
<tr class="snow">  
<span class="quote">
```

## Overview

1. Overview of HTML
2. Styling and Cascading
3. Dynamic Event Handling with JavaScript

## Java Script

- I was told recently that more programs are written in JavaScript than any other language
  - I find that scary !
- Advantages :
  - Can modify HTML on the client
  - Fairly easy to write simple functions
- Disadvantages :
  - Weak typing
  - Poor development tools
  - Many scalability problems (maintenance, reliability, readability, security, efficiency, portability, ...)

11 October 2011

© Offutt, 2011

29

## Java Script Origins

- JavaScript was introduced as part of the Netscape 2.0 browser
- Microsoft soon released its own version called Jscript
- ECMA developed a standard language known as ECMAScript
- ECMAScript Edition 3 is widely supported and is what we will call “JavaScript”

11 October 2011

© Offutt, 2011

30

## Developing JavaScript Software

- Writing JavaScript code
  - Any text editor (e.g., Vim, Notepad, Emacs)
  - Specialized software (e.g., MS Visual InterDev)
- Executing JavaScript
  - Load into browser (need HTML document)
  - Browser detects syntax and run-time errors
    - Firefox: JavaScript console lists errors (Ctrl-Shift-J)
    - IE6: Exclamation icon and pop-up window (Bottom left)

11 October 2011

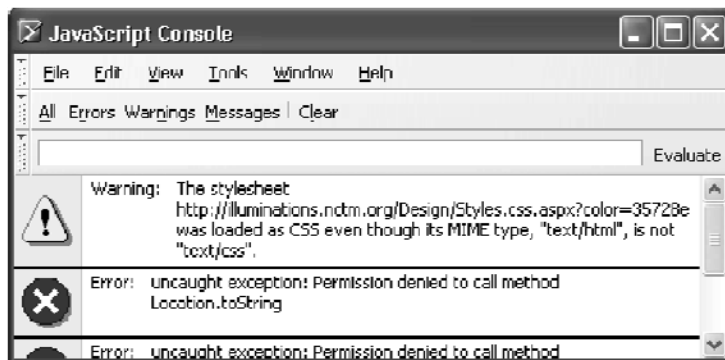
© Offutt, 2011

31

## Java Script Development

Mozilla JavaScript console

Tools → JavaScript Console (Ctrl-Shift-J)



11 October 2011

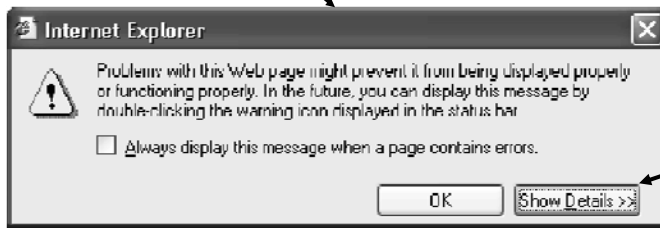
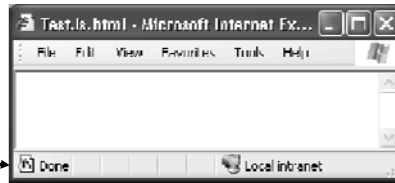
© Offutt, 2011

32

# Java Script Development

IE 6 / 7 Error Window :

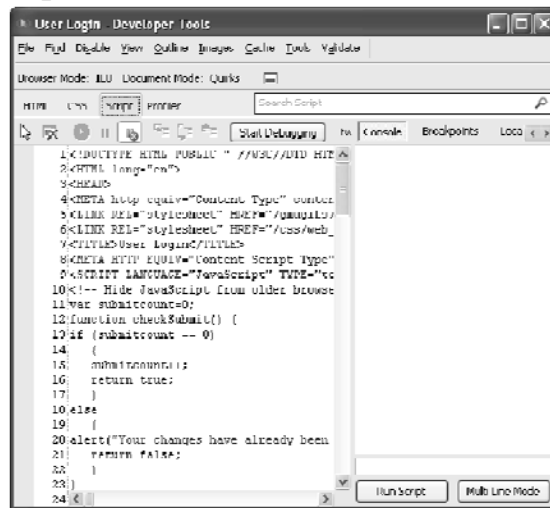
Error indicator;  
double-clicking icon  
opens error window



Click to see  
error messages

# Developing JavaScript Software

IE 8 Developer Tools :



Not sure how to see the JS errors ...

## Java Script

- JavaScript is **not** Java
- JavaScript is a scripting language :
  - Embedded in HTML
  - Interpreted as the page is loaded
  - Can manipulate the HTML page
- Is executed on the client (client side)
  - Serverside JavaScript: Netscape's Livewire allows JavaScript to be used for form processing on the server
  - We will not discuss this
- No type checking

11 October 2011

© Offutt, 2011

35

## Java Script Characteristics

- JavaScript does not need to be compiled
  - JavaScript is an interpreted language
  - A JS interpreter is software that runs inside a browser that reads and executes JavaScript
- Interpreted vs. compiled languages :
  - Advantage : simplicity
  - Disadvantage : efficiency, maintainability, scalability, reliability

11 October 2011

© Offutt, 2011

36

## Why and Why Not JavaScript?

- What can be done with JavaScript on the client and cannot be done with other techniques on the server?
  - Monitor user events and take action
  - Some dynamic effects
- What can be done on both client and server, but are better with JavaScript?
  - Build HTML dynamically when page is loaded
  - Interactive web pages
  - Communicate with the server asynchronously (Ajax)
- What are the drawbacks of JavaScript?
  - Platform dependent
  - Can be turned off
  - Performance
  - Security
  - Very hard to write reliable and maintainable Javascript

11 October 2011

© Offutt, 2011

37

## Where Does JavaScript Go?

- Between `<head>` and `</head>` or `<body>` and `</body>`
  - `<SCRIPT Language="JavaScript">...</SCRIPT>`
  - `<SCRIPT Language="JavaScript" Src="com_function.js">`
  - ...
  - `</SCRIPT>`
- Some "script" calls may be embedded in the HTML tags
  - `<SELECT name=country onchange="jmp(url)">`
  - `<A href="javascript:new_window();">New Win</A>`
- Comments: `//` single-line and `/* */` multi-line

11 October 2011

© Offutt, 2011

38

# JavaScript Objects

- JS objects have collections of properties
- A property is something that can be modified :
  - Data properties : primitive values or references to objects
  - Method properties : can be executed

# JavaScript Syntax (1)

- Data Types
  - Numeric, String, Array, Boolean, NULL (NaN)
  - Mixing numbers and strings (parseInt, parseFloat)
- Variables
  - Begin with letter or \_, no white space or punctuation
  - Case sensitive
  - No fixed types
- Operators (similar to Java)
- Statements (similar to Java: if, for, while, switch...)
  - for in loops :  
var myArray = new Array ("A", "B", "C");  
for (i in myArray) ⇔ for (i=0; i<3; i++)

## JavaScript Syntax (2)

### 28 Reserved Words

[http://en.wikibooks.org/wiki/JavaScript/Reserved\\_Words](http://en.wikibooks.org/wiki/JavaScript/Reserved_Words)

break	do	if	switch	typeof
case	else	in	this	var
catch	false	instanceof	throw	void
continue	finally	new	true	while
default	for	null	try	with
delete	function	return		

11 October 2011

© Oflutt, 2011

41

## JavaScript Syntax (3)

- Much of the syntax is similar to C and Java
- No typing (dynamic typing)  

```
var X = 5;  
X = 5.5;
```
- Fields can be added at anytime  

```
var tmp = new Object();  
tmp.f1 = "yes";  
tmp.f2 = 42;  
– Literal notation: tmp2 = {f1:"yes", f2:42};
```
- Iterating over fields :  

```
for (f in tmp2)  
  document.writeln (f + "is" + tmp2[f]);
```

11 October 2011

© Oflutt, 2011

42

## JavaScript Syntax (4)

### Constructors

```
function Course (name, num, time, enroll)
{
    this.name = name;
    this.num = num;
    this.time = time;
    this.enroll = enroll;
}
```

11 October 2011

© Offutt, 2011

43

## JavaScript Syntax (5)

- Object Methods :

```
function Print ()
{
    document.writeln("<TABLE>\n" + "<TR><TH>Field<TH>Value\n");
    for (fname in this)
        document.writeln("<TR><TD>" + fname + "<TD>" + this[fname]);
    document.writeln("</TABLE>");
}
```

- Use :

```
var SWE432 = new Course ("Software for the Web", "SWE 432",
                        "Th 4:10", 44);

SWE432.Print ();
```

11 October 2011

© Offutt, 2011

44

## JavaScript Syntax (6)

- Arrays :  

```
var Students = new Array (36);  
Students (0) = "AA";  
for (var s=0; s<= Students.length; s++)  
    document.writeln (Students (s));
```
- Put functions in <HEAD>
  - Easier to read
  - Function can be used in multiple places
  - Functions parsed and available first if page loads slowly
- Use a backslash to insert quotes in strings:  
"You can\'t be too careful of \'s."

11 October 2011

© Offutt, 2011

45

## JavaScript Syntax (7)

- Older browsers do not understand JavaScript
- Hide JavaScript inside HTML comments :  

```
<!--  
    -- Put the JavaScript here  
-->
```
- Browsers that support JavaScript are smart enough to look inside that comment
  - Few of these browsers are still in use ... pre-2001

11 October 2011

© Offutt, 2011

46

## First Example

<http://www.cs.gmu.edu/~offutt/classes/432/examples/JS/updatefield-js.html>

- How do we deploy JavaScript components?
- How do we obtain data and how do we export data?
- How do we monitor user events, and how do we take actions?
- Pay attention to those statements :
  - `document.FieldName.Entry.value`
  - `OnMouseOver = "ChangeFieldName ()"`
  - `OnClick = "ChangeFieldName ()"`

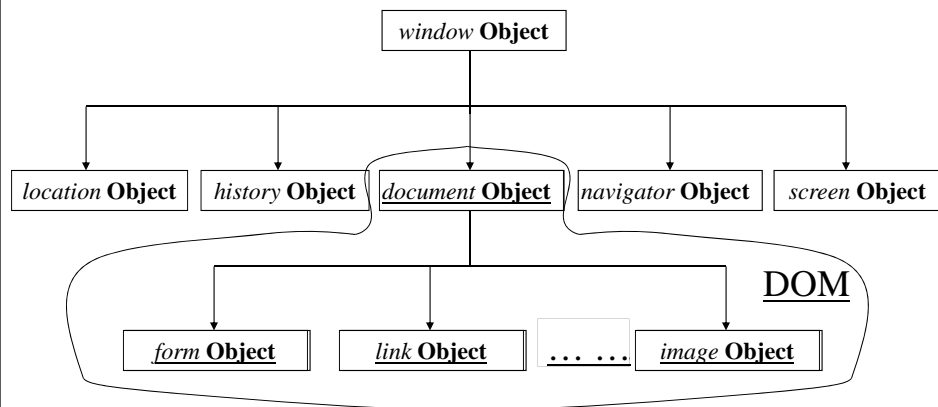
11 October 2011

© Offutt, 2011

47

## Browser Object Model (BOM)

BOM – collection of objects that the browser makes available to us for use with JavaScript



11 October 2011

© Offutt, 2011

48

## Using BOM Objects

- *window* is a global object
- Addressing is done by separating objects, methods, and properties with dots
  - `window.alert("Hello");`
  - `alert("Hello");`
- Alternative ways to access the objects :
  - `document.forms[0].elements[0].value`
  - `document.myform.nameinput.value`
  - `document.myform["nameinput"].value`

11 October 2011

© Offutt, 2011

49

## More BOM / DOM Examples

- `window.open`, `window.close`, `confirm`, `alert`
- `history.go(-1)` `history.go(0)`
- `window.location.replace("")`, `location.href=""`;
- `window.Navigator.appName` `navigator.appVersion`  
`navigator.cookieEnabled`, `navigator.userAgent`
- `window.screen.height`, `screen.width`
- `window.document`
- `this.value`

11 October 2011

© Offutt, 2011

50

## JavaScript Objects (1)

- Math object:
  - Methods: `sin()`, `cos()`, `floor()`, `max()`, `max()`, ...
- Number object:
  - Data : `MIN_VALUE`, `MAX_VALUE`, `NaN`, `POSITIVE_INFINITY`, `NEGATIVE_INFINITY`, `PI`
  - Methods : `isNaN()`, `toString()`
  - `if (isNaN(age)) ...` — false if age is a number, true otherwise

11 October 2011

© Offutt, 2011

51

## JavaScript Objects (2)

### Strings :

- Catenate : `var str = "Andrew" + " " + "Offutt";`
- Length : `str.length`; — 13
- Finding a character : `str.charAt (7)`; — 'O', starts at zero!
- Finding a position : `str.indexOf ('f')`; — 8, first occurrence
- Getting a substring : `str.substring (0, 5)`; — "Andrew"
- Lower case : `str.toLowerCase()`; — "andrew offutt"
- Upper case : `str.toUpperCase()`; — "ANDREW OFFUTT"
- `var pos = str.search (/ff/)`; — 8, where "ff" starts
- `var nameArr = str.split (" ")`; — ["Andrew", "Offutt"]

11 October 2011

© Offutt, 2011

52

## JavaScript Uses

- Two common purposes :
  - I. Build HTML dynamically when page is loaded
  - II. Monitor user events and take action
- Classes of applications
  1. Customizing web pages
  2. Making web pages more dynamic
  3. Validating forms
  4. Manipulating cookies
  5. Interacting with frames
  6. Calling Java programs
  7. Accessing JavaScript from Java

## I. Generating HTML Dynamically

- JavaScript will execute when page is loaded, and output appears in the document

```
<SCRIPT Language="JavaScript">
  <!-- Comment so browsers without JS will ignore it.
    document.write ("JavaScript output\n");
  // -- >
</SCRIPT>
```
- This is just a demonstration – the line could have been inserted directly into the HTML

## I. Generating HTML Dynamically (2)

```
function ReferringPage ()
{
  if (document.referrer.length == 0)
    return ("<EM>none</EM>");
  else
    return (document.referrer);
}

document.writeln ("Document Info: \n" +
"<UL>\n" +
" <LI><B>URL:</B> " + document.location + "\n" +
  document.lastModified + "\n"
" <LI><B>Title:</B> " + document.title + "\n" +
" <LI><B>Referring page:</B> " +
  ReferringPage () + "\n" +
"</UL>");
```

11 October 2011

© Offutt, 2011

55

## I. Generating HTML Dynamically (3)

```
document.writeln ("Browser Info: \n" +
"<UL>\n" +
" <LI><B>Name:</B> " + navigator.appName + "\n" +
" <LI><B>Version:</B> " + navigator.appVersion + "\n" +
"</UL>");
```

### Language :

- Specifies different versions of JavaScript
- Different browsers know different JavaScripts

11 October 2011

© Offutt, 2011

56

## I. Generating HTML Dynamically (4)

```
<script language="Javascript">
if (top.frames.length != 0) // Keep page from loading in a frame
{
    top.location = document.URL;
}

function setFocus() // Set the mouse focus in a form
{
    document.gcdform.LHS.focus();
}

var lastModificationDate = new Date(document.lastModified)
document.write ('<span class="quote">Last update: ' +
(lastModificationDate.getMonth()+1) + '/' +
lastModificationDate.getDate() + '/2010' + '</span>');
</script>
```

11 October 2011

© Offutt, 2011

57

## II. Monitoring User Events

- Clicking on a button — onClick
- Clicking on a link — onSelect
- Exiting a page — onUnload
- Moving onto or off of a link — OnMouseOver, OnMouseOut
- Input focus on a FORM element — onFocus
- Submitting form — onSubmit
- Image loading error — onAbort

10/11/2011

© Offutt & Wu

58

## II. Monitoring User Events (2)

```
<HEAD>

  <SCRIPT>
    <!--
      function DontClick ()
      {
        alert ("I told you not to click!");
      }
    // -->
  </SCRIPT>
</HEAD>

<FORM>
  <INPUT Type="button" Value="Don't click me"
    onClick="DontClick()">
</FORM>
```

10/11/2011

© Offutt & Wu

59

## Events and Event Handling Examples

<http://www.cs.gmu.edu/~offutt/classes/432/examples/JS/redrawform-js.html>

- **Event handlers as Attributes**  
<A HRef="next.html" Name="next" OnClick="verify();">next</A>
- **Event handlers as Properties**  
document.links[0].onclick = verify();  
Note: 1. document.links ["next"] won't work.  
2. Verify is the name of the event handling function.
- Obtain values within the page
- Change values within the page
- Add or remove objects

10/11/2011

© Offutt & Wu

60

## Validating Forms

- Make sure that each data value is “*valid*”
  - The value conforms to specific constraints
  - Not necessarily correct
- Validate each input field (text, radio, and etc)
- Submitting form data
  - `<FORM onSubmit="return submitIt(this)" Method="POST" ACTION="...">`
    - if `submitIt(this)` returns false, the data will not be passed to the server
    - “this” refers to the current form

10/11/2011

© Offutt & Wu

61

## JavaScript Uses

### Classes of applications

1. Customizing web pages
2. Making web pages more dynamic
3. Validating forms
4. Manipulating cookies
5. Interacting with frames
6. Calling Java programs
7. Accessing JavaScript from Java

10/11/2011

© Offutt & Wu

62

## 1. Customizing Web Pages

- Compatibility is a major problem with web browsers
- Professional web sites go to a lot of trouble to check and adjust the web pages
- Old browsers do not support BGCOLOR on tables:

```
if navigator.appVersion.substring (0,1) >= 3 {  
  var HeadCellColor = "black";  
  var HeadFontColor = "white";  
} else {  
  var HeadCellColor = "white";  
  var HeadFontColor = "black";  
}  
dw("<TABLE>\n" +  
  "<TR BGCOLOR=" + HeadCellColor + ">\n"+  
  "<TH><FONT Color=" + HeadFontColor + ">\n" +
```

*Note: "dw" is a shorthand for "document.writeln" – just for use in the slides.*

10/11/2011

© Offutt & Wu

63

## 1. Customizing Web Pages (2)

Adjusting images to fit window size

```
function Image (url, w, h)  
{  
  return("<IMG Src=\"" + url + "\" +  
    \"Width=" + w +  
    \"Height=" + h + ">");  
}  
function Apple (w)  
{  
  return (Image ("apple.gif", w, math.round (w*1.323)));  
}  
function Apple2 (w)  
{  
  return (Image ("apple.gif", w, math.round (w*1.155)));  
}
```

10/11/2011

© Offutt & Wu

64

## 1. Customizing Web Pages (3)

```
Adjusting images to fit window size ... cont ...
if navigator.appVersion.substring (0, 1) >= 4)
{
  var imageWidth = Window.innerWidth / 4;
  var fontSize = Math.min (7, math.round
                          (Window.innerWidth / 100));
} else {
  var imageWidth = 200;
  var fontSize = 5;
}
dw ("<TABLE>\n" +
    " <TR><TD>" + Apple (imageWidth) + "\n" +
    "      <TH><FONT Size=" + fontSize + ">\n" +
    "      \"Apples are great during the break!\"\n" +
    "      </FONT>\n" +
    " <TABLE>\n");
```

10/11/2011

© Offutt & Wu

65

## 1. Customizing Web Pages (4)

- Checking for plug-ins
- Firefox and Netscape versions 3 and above have a navigator.plugins array (IE does not)

```
if (navigator.plugins ["cosmo player 1.0"])
  dw ("<EMBED src="CoolWorld.vrml" ... >");
else
  dw ("This example requires VRML.");
```

10/11/2011

© Offutt & Wu

66

## 2. Making Web Pages Dynamic

- We can change images on mouse movements or time
- document.images – an array of IMG objects

```
<IMG src="Cool.jpg" Name="cool">
```

```
function MakeCooler ()  
{  
    document.images ["cool"].src = "cooler.jpg";  
}
```

10/11/2011

© Offutt & Wu

67

## 2. Making Web Pages Dynamic (2)

Changing on click :

```
function SetImg (name, image)  
{  
    document.images [name].src = image;  
}  
  
function ClickButton (name, GrayImg)  
{  
    var origImg = document.images[name].src; // save it.  
    SetImg (name, GrayImg);  
    var resetstring = "SetImg  
        (" + name + "\",\"" + origImg + "\")";  
    // Runs after 100 milliseconds  
    set Timeout (resetString, 100);  
}
```

10/11/2011

© Offutt & Wu

68

## 2. Making Web Pages Dynamic (3)

1. Attach ClickButton() to a button
2. Cache the images

```
<A Href= "location1.html" onClick="ClickButton (\\"button1\\",  
  \\"images/Button1-down.gif\\")">  
<IMG Src="images/Button2-Up.gif" Name="Button1"> </A>
```

3. Force a pre-load to save time:

```
ImageFiles = new Array ("images/Button1-Up.gif",  
                        "images/Button1-Down.gif",  
                        "images/Button2-Up.gif",  
                        "images/Button2-Down.gif");  
ImageObjects = new Array (ImageFiles.length);  
for (var i=0; i<ImageFiles.length; i++)  
{  
  imageObjects [i] = new Images (150, 25);  
  imageObjects [i].src = ImageFiles [i];  
}
```

10/11/2011

© Offutt & Wu

69

## JavaScript Tips

- Build your script one piece at a time
- Store to files and load with HTML files
- Do not change variable types
- Do not use same names for global and local variables
- Remember, JavaScript is case sensitive
- Viewing JavaScript errors
  - Firefox : Tools → Error Console (Ctrl-Shift-J)
  - Microsoft IE : ?
- Debugger
  - Microsoft IE
    - <http://msdn.microsoft.com/scripting>
  - Firefox Javascript debugger – Venkman
    - <https://addons.mozilla.org/en-US/firefox/addon/216>
  - Another Firefox debugger – Firebug
    - <http://getfirebug.com/>

10/11/2011

© Offutt & Wu

70

## JavaScript Examples

Most of the previous examples are fully worked out here:

<http://www.cs.gmu.edu/~offutt/classes/432/examples/JS/>

## Conclusions

**Learning HTML syntax is not hard**

**Learning JS syntax is not hard**

**Designing good dynamic web pages is  
VERY hard**