CITS 1231 Web Technologies
JavaScript: Document, Event, Date objects
Document Object

• JavaScript can interact with HTML page elements via the Document Object.

• The Document Object is accessed via the keyword document.

• Recall from previous lectures: document.writeln() to print out to page.

• Document Object has:
  – Collections
  – Properties
  – Methods
Document Methods

- **write()**
  - Writes HTML or JavaScript to a document

- **writeln()**
  - Same as write(), but adds a newline character

Note that newline characters are normally ignored by browser. You can use the `<pre>` tag to preserve line breaks. Example:

```html
<pre>
  document.writeln("Hello");
  document.writeln("World");
  document.write("Hello");
  document.write("World");
</pre>
```
Document Methods

- `getElementById()`
  - Accesses the first element with the specified id
  - One can then access the various properties of this element
  - `innerHTML` property refers to the text between the element’s opening and closing tags

```html
<html>
<head>
<script type="text/javascript">
function getValue()
{
    var x=document.getElementById("myHeader");
    alert(x.innerHTML);
}
</script>
</head>
<body>
<h1 id="myHeader" onclick="getValue()">Click <em>me!</em></h1>
</body>
</html>
```
Document Methods

• `getElementsByTagName()`
  - Accesses all elements with a specified name

```html
<html>
<head>
<script type="text/javascript">
function calc()
{
    var x=document.getElementsByName("para");
    alert("There are "+x.length+" paragraphs. First one says:"+x[0].innerHTML);
}
</script>
</head>
<body>
<p name="para">Para 1</p>
<p name="para">Para 2</p>
<input type="button" onclick="calc()" value="How paragraphs named 'para'?" />
</body>
</html>
```
Document Methods

• **getElementsByName()**
  - Accesses all elements with a specified tagname

```html
<html>
<head>
<script type="text/javascript">
function getTags() {
    var x=document.getElementsByTagName("input");
    alert("There are "+x.length+" inputs. First second has value: "+x[1].value);
}
</script>
</head>
<body>
<input type="text"/>
<input type="text"/>
<input type="text"/>
<input type="button" onclick="getTags()" value="How many input elements?"/>
</body>
</html>
```
Understanding Event Handlers

• Event
  – Action that occurs within a Web browser

• Event handler
  – Statement that tells browsers what code to run in response to specified event

• Syntax to insert event handler as an attribute
  \(<\text{element onevent=",script" } ...> ...\)
# JavaScript Event Handlers

<table>
<thead>
<tr>
<th>Category</th>
<th>Event Handler</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Window and document event handlers</td>
<td>onload</td>
<td>The browser has completed loading the document</td>
</tr>
<tr>
<td></td>
<td>onunload</td>
<td>The browser has completed unloading the document</td>
</tr>
<tr>
<td></td>
<td>onerror</td>
<td>An error has occurred in the JavaScript program</td>
</tr>
<tr>
<td></td>
<td>onmove</td>
<td>The user has moved the browser window</td>
</tr>
<tr>
<td></td>
<td>onresize</td>
<td>The user has resized the browser window</td>
</tr>
<tr>
<td></td>
<td>onscroll</td>
<td>The user has moved the scrollbar within the browser window</td>
</tr>
<tr>
<td>Form event handlers</td>
<td>onfocus</td>
<td>The user has entered an input field</td>
</tr>
<tr>
<td></td>
<td>onblur</td>
<td>The user has exited an input field</td>
</tr>
<tr>
<td></td>
<td>onchange</td>
<td>The content of an input field has changed</td>
</tr>
<tr>
<td></td>
<td>onselect</td>
<td>The user has selected text within an input or text area field</td>
</tr>
<tr>
<td></td>
<td>onsubmit</td>
<td>The user has submitted the Web form</td>
</tr>
<tr>
<td></td>
<td>onreset</td>
<td>The user has reset the Web form</td>
</tr>
<tr>
<td>Mouse and keyboard event handlers</td>
<td>onkeydown</td>
<td>The user has pressed a key</td>
</tr>
<tr>
<td></td>
<td>onkeypress</td>
<td>The user has pressed and released a key</td>
</tr>
<tr>
<td></td>
<td>onclick</td>
<td>The user has clicked the mouse button</td>
</tr>
<tr>
<td></td>
<td>ondblclick</td>
<td>The user has double-clicked the mouse button</td>
</tr>
<tr>
<td></td>
<td>onmousedown</td>
<td>The user has pressed down the mouse button</td>
</tr>
<tr>
<td></td>
<td>onmouseup</td>
<td>The user has released the mouse button</td>
</tr>
<tr>
<td></td>
<td>onmousemove</td>
<td>The user has moved the mouse pointer over the element</td>
</tr>
<tr>
<td></td>
<td>onmouseout</td>
<td>The user has moved the mouse pointer off of the element</td>
</tr>
</tbody>
</table>

Many more. See text or [http://www.w3schools.com/jsref/dom_obj_event.asp](http://www.w3schools.com/jsref/dom_obj_event.asp)
Event: onkeydown

• The user has pressed a key. Example:

```html
<html>
<head>
<script type="text/javascript">
  function update(id) {
    var txt=document.getElementById(id).value;
    document.getElementById(id).value=txt.toUpperCase();
  }
</script>
</head>
<body>
  Type something: <input type="text" id="fname" onkeydown="update(this.id)" />
</body>
</html>
```

• Why is the last character not uppercase?
• Character is not added to input text until key goes up.
• When key down, function is called to convert input text to upper case.
• Only when function returns and key is released, the last character is appended to input text (still lower case!).
Event: onkeyup

• The user pressed a key and released it. Example:

```html
<html>
<head>

  <script type="text/javascript">
    function update(id) {
      var txt=document.getElementById(id).value;
      document.getElementById(id).value=txt.toUpperCase();
    }
  </script>

<body>
  Type something: <input type="text" id="fname" onkeyup="update(this.id)" />
</body>
</html>
```

• Each character is made upper case on key going up.
Event: onchange

• The content of the input field has changed. **Example:**

```html
<html>
<head>
  <script type="text/javascript">
    function update(id) {
      var txt=document.getElementById(id).value;
      document.getElementById(id).value=txt.toUpperCase();
    }
  </script>
</head>
<body>
  Type something: <input type="text" id="fname" onchange="update(this.id)" />
</body>
</html>
```

• No change to characters as you type.

• Event onchange is triggered only when you hit Enter key.
Event: onclick

- User has clicked mouse button. **Example:**

```html
<html>
<head>
<script type="text/javascript">
    function calc() {
        var num1 = parseInt(document.getElementById('num1').value);
        document.getElementById('num2').value = num1*num1;
    }
</script>
</head>
<body>
    Enter a number: <input type="text" id="num1" />
    <button onclick="calc()">Click to square it</button>
    <input type="text" id="num2" readOnly="true" />
</body>
</html>
```
Date Object

• Allows your JavaScript to work with dates and times.
• There are 4 ways to create a Date object:
  • Date Object with current date and time:

```html
<html>
<body>
<script type="text/javascript">
var dt=new Date();
document.write(dt);
</script>
</body>
</html>
```
Date Initialise Using String

- variable = new Date("month day, year hours:minutes:seconds");

```html
<html>
<body>
<script type="text/javascript">
var dt=new Date("June 1, 1977 10:11:00");
document.write(dt);
</script>
</body>
</html>
```
Date Initialised Using Number

- var dt = new Date(num)
  - num is number of mS past 01 Jan 01, 1970 00:00:00 UTC
  - Perth is UTC+08:00:00
  - There are 86,400,000 mS in one day

```html
<html>
<body>
  <script type="text/javascript">
    var dt=new Date(86400000);
    document.write(dt);
  </script>
</body>
</html>
```
Date Initialised Using Several Parameters

- new Date(yr, mth, day, hrs, mins, secs, ms)
  - Parameters are numbers and optional.
  - If you leave out a parameter, zero will be assumed and all subsequent parameters must be left out also.
  - new Date(2010) is valid.
  - new Date(2010, , 1) is invalid.

```html
<html>
<body>
<script type="text/javascript">
  var dt=new Date(2010,0,1,2);
  document.write(dt);
</script>
</body>
</html>
```
## Extracting Date and Time Values

<table>
<thead>
<tr>
<th>Method</th>
<th>Retrieves</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>thisDate.getSeconds()</td>
<td>Retrieves the seconds value</td>
<td>28</td>
</tr>
<tr>
<td>thisDate.getMinutes()</td>
<td>Retrieves the minutes value</td>
<td>35</td>
</tr>
<tr>
<td>thisDate.getHours()</td>
<td>Retrieves the hours value (in military time)</td>
<td>14</td>
</tr>
<tr>
<td>thisDate.getDate()</td>
<td>Retrieves the day of the month value</td>
<td>15</td>
</tr>
<tr>
<td>thisDate.getDay()</td>
<td>Retrieves the day of the week (0 = Sunday, 1 = Monday, 2 = Tuesday, 3 = Wednesday, 4 = Thursday, 5 = Friday, 6 = Saturday)</td>
<td>5</td>
</tr>
<tr>
<td>thisDate.getMonth()</td>
<td>Retrieves the month value (0 = January, 1 = February, 2 = March, etc.)</td>
<td>5</td>
</tr>
<tr>
<td>thisDate.getFullYear()</td>
<td>Retrieves the four-digit year value</td>
<td>2007</td>
</tr>
<tr>
<td>thisDate.getTime()</td>
<td>Retrieves the time value, as expressed in milliseconds, since January 1, 1970</td>
<td>1,181,936,128,000</td>
</tr>
</tbody>
</table>
## Setting Date and Time Values

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>DateObject.setSeconds(value)</code></td>
<td>Sets the seconds value of <code>DateObject</code> to <code>value</code></td>
</tr>
<tr>
<td><code>DateObject.setMinutes(value)</code></td>
<td>Sets the minutes value of <code>DateObject</code> to <code>value</code></td>
</tr>
<tr>
<td><code>DateObject.setHours(value)</code></td>
<td>Sets the hours value of <code>DateObject</code> to <code>value</code></td>
</tr>
<tr>
<td><code>DateObject.setDate(value)</code></td>
<td>Sets the day of the month value of the <code>DateObject</code> to <code>value</code></td>
</tr>
<tr>
<td><code>DateObject.setMonth(value)</code></td>
<td>Sets the month number of <code>DateObject</code> to <code>value</code> (0=January, 1=February, etc.)</td>
</tr>
<tr>
<td><code>DateObject.setFullYear(value)</code></td>
<td>Sets the four-digit year value of <code>DateObject</code> to <code>value</code></td>
</tr>
<tr>
<td><code>DateObject.setTime(value)</code></td>
<td>Sets the time of <code>DateObject</code> in milliseconds since January 1, 1970</td>
</tr>
</tbody>
</table>
Creating a Custom Date Function

- `showDate()` function returns dd/mm/yyyy formatted date.

```html
<html>
<head>
  <script type="text/javascript">
    function showDate(dateObj) {
      var thisDate = dateObj.getDate();       // get day
      var thisMonth = dateObj.getMonth()+1;    // get month (jan=0 so add 1)
      var thisYear = dateObj.getFullYear();   // get year
      return thisMonth + "/" + thisDate + "/" + thisYear;     // format as dd/mm/yyyy
    }
  </script>
</head>
<body>
  <script type="text/javascript">
    var dt = new Date();            // create current date obj
    var msg = showDate(dt);    // call showDate fn
    document.write(msg);         // display formatted date
  </script>
</body>
</html>
```
Creating a Custom Time Function

- `showTime()` function returns hh:mm:ss formatted time.

```html
<html>
<head>
<script type="text/javascript">
function showTime(dateObj) {
    thisSecond = dateObj.getSeconds(); // get seconds
    var thisMinute = dateObj.getMinutes(); // get minutes
    var thisHour = dateObj.getHours(); // get hours

    return thisHour + ":" + thisMinute + ":" + thisSecond; // format as hh:mm:ss
}
</script>
</head>
<body>
<script type="text/javascript">
var dt = new Date();
var msg = showTime(dt);
document.write(msg);
</script>
</body>
</html>
```
Calculate Days to New Year

- `calcDays()` function returns number days to new year.

```html
<html>
<head>
<script type="text/javascript">
function calcDays(dateObj) {
    var nextYear=dateObj.getFullYear()+1; // get next year
    var newYear=new Date(nextYear,0,1); // create date of next new year
    var mS = newYear-dateObj; // calc mS from now to new year
    var days = mS /(1000*60*60*24); // convert mS to days
    return days;
}
</script>
<body>
<script type="text/javascript">
var dt = new Date();
var msg = calcDays(dt);
document.write(msg);
</script>
</body>
</html>
```

We want integer whole days. How to rid of decimals?
**Calculate Days to New Year - Cont**

- *calcDays()* function returns number days to new year.

```html
<html>
<head>
<script type="text/javascript">
function calcDays(dateObj) {
    var nextYear = dateObj.getFullYear() + 1; // get next year
    var newYear = new Date(nextYear, 0, 1); // create date of next new year
    var mS = newYear - dateObj; // calc mS from now to new year
    var days = mS / (1000*60*60*24); // convert mS to days
    return days;
}
</script>
<body>
<script type="text/javascript">
var dt = new Date();
var days = calcDays(dt);
document.write(Math.floor(days));
</script>
</body>
</html>
```
**Time to New Year**

- Time to New Year can be expressed in number of days, hours, minutes and seconds.

- `calcDays()` returns a number.
  - Integer portion is number whole days to new year.
  - Decimal portion can be converted to remainder hours, mins and secs.

- `calcHours()` – returns remainder hours

- `calcMins()` – returns remainder minutes

- `calcSecs()` – returns remainder seconds
calcHours() - remainder hours to New Year

<html>
<head>
<script type="text/javascript">

...  

function calcHours(days) {
    var fracDay = days - Math.floor(days);   // get fractional day
    var hours = fracDay*24;                   // convert to hours
    return hours;

}
</script>
</head>
<body>

<pre>

Days=2.60
Hours=23

</pre>
</body>
</html>
calcMins() - remainder minutes to New Year

<html>
<head>
<script type="text/javascript">
...

function calcMins(hours) {
    var fracHours = hours - Math.floor(hours); // get fractional hours
    var minutes = fracHours*60; // convert to minutes
    return minutes;
}
</script>
</head>
<body>
<pre>
...<br>
</pre>
</body>
calcSecs() - remainder seconds to New Year

```html
<html>
<head>
  <script type="text/javascript">
    function calcSecs(minutes) {
      var fracMins = minutes - Math.floor(minutes); // get fractional minutes
      var seconds = fracMins*60; // convert to seconds
      return seconds;
    }
  </script>
</head>
<body>
  Days=260
  Hours=23
  Minutes=18
  Seconds=24
</body>
</html>
```
Automatically Updating Countdown

• Use JavaScript timing events

• $id = setTimeout("command", delay)$
  – execute code sometime in future

• $clearTimeout(id)$ – cancels $setTimeOut()$
setTimeout Example

```html
<html>
<head>
  
  <script type="text/javascript">
    var c=1000;
    function start()
    {
      document.getElementById('num').value=c;
      c--;
      setTimeout("start()",1000);
    }
  
  </script>
</head>
<body>
<form>
  <input type="button" value="Start!" onClick="start()">
  <input type="text" id="num" value="1000">
</form>
</body>
</html>
```
Creating the *monthName* Array

```javascript
function calendar() {
    document.write("<table id='calendar_table'>");
    document.write("</table>");
}

function writeCalTitle(calendarDay) {
    var monthName = new Array("January", "February", "March", "April", "May",
                             "June", "July", "August", "September", "October", "November", "December");
}
```
Creating the `writeDayNames()` Function

```javascript
function calendar() {
    var calDate = new Date("May 18, 2007");
    document.write("<table id='calendar_table'>");
    writeCalTitle(calDate);
    writeDayNames();
    document.write("</table>");
}

function writeDayNames() {
    var dayName = new Array("Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat");
    document.write("<tr>");
    for (var i=0; i<dayName.length; i++) {
        document.write("<th class='calendar_weekdays'>"+dayName[i]+"</th>");
    }
    document.write("</tr>");
}
```
The Complete `daysInMonth()` Function

```javascript
function daysInMonth(calendarDay) {
    var thisYear = calendarDay.getFullYear();
    var thisMonth = calendarDay.getMonth();
    var dayCount = new Array(31,28,31,30,31,30,31,31,30,31,30,31);
    if (thisYear % 4 == 0) {
        if (((thisYear % 100 !=0) || (thisYear % 400 == 0)) {
            dayCount[1] = 29; // this is a leap year
        }
    }
    return dayCount[thisMonth]; // return the number of days in the month
}
```
Running JavaScript Commands as Links

- Syntax

  `<a href="javascript:script">content</a>`

- The following code runs the `calcTotal()` function

  `<a href="javascript:calcTotal()">Calculate total cost</a>`