

# Even More Javascript

CS174

Chris Pollett

Oct 9, 2006.

# Outline

- Moving Elements
- Visibility
- Dynamic Content
- Timers
- Introduction to XML

# Moving Elements

- Last day we talked about absolute and relative positioning.
- Today we'll consider this in the context of moving elements around the screen.
- Consider the following Javascript function:

```
function moveIt(id, newTop, newLeft)
{
    myStyle = document.getElementById(id).style;
    myStyle.top = newTop + "px"; /* notice how CSS properties are
    properties of the style object*/
    myStyle.left = newLeft + "px";
}
```

- This could be used to move an element to a specific location on the screen.

# Element Visibility, Colors and Fonts

- You can control the visibility of an XHTML element using the CSS property.

```
<div id="test" style="visibility: hidden">hi there</div>
```

```
<input type="button" onclick='show("test")' />
```

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

```
    function show(id)
```

```
    {
```

```
        myStyle = document.getElementById(id).style;
```

```
        myStyle.visibility = "visible";
```

```
    }
```

```
</script>
```

- In a similar fashion one can change other CSS properties in response to events. For instance, if you like you could change the color or font type or size.

# Dynamic Content

- It is sometimes useful to be able to give help information when a person hovers over a form element. This can be achieved as follows:

```
<script type="text/javascript" >
    msgs = ["<p>mesg0</p>", "<p>mesg1</p>"]
    function showMessage(evt, num)
    {
        box = document.getElementById("adviceBox");
        box.innerHTML = msgs[num];
        myStyle =box.style;
        myStyle.position = "absolute";
        myStyle.top = evt.clientY +10 +"px";
        myStyle.left = evt.clientX +10 +"px";
        myStyle.visibility = "visible";
    }
    function hideMessage()
    {
        myStyle = document.getElementById("adviceBox").style;
        myStyle.visibility = "hidden";
    }
</script>
<div id="adviceBox" style="visibility:hidden" />
<input type="text" onmouseover="showMessage(event, 1)" onmouseout="hideMessage()" />
```

# Timers

- It is sometimes useful to update the contents of a page every so many milliseconds.
- Javascript support this by using either the functions `setInterval` or `setTimeout`.
- For example,

```
setTimeout(myCallback, repeatTimeInMilliSec);  
// myCallback is the Javascript function you would like  
called
```
- To stop the timer you can use `clearTimeout/clearInterval`.

# Introduction to XML

- Recall that HTML was originally specified as an SGML (Standard Generalized Markup Language) doctype.
- HTML only provides limited semantic information about a document. You can tell if something is in `<h1>` tags that is probably important, but not much else.
- Starting in 1998 a stripped down version of SGML called XML (extensible markup language) was developed to make it easier to create new tag-based mark-up languages where the tags can be used to carry whatever semantic information is desired.

# The Syntax of XML

- A new XML language can be specified in one of two ways:
  - Give a DTD (Document Type Definition) -- this is closer to the SGML way of specifying languages
  - Give an XML schema -- unlike DTDs such schemas are also XML documents so can easily be parse with XMLParsers. Further schemas can be more detailed.
- In both a DTD and a schema, one specifies:
  - What tag elements exist in the language
  - What subelements or data a given element is allowed to contain
  - What attributes an element has and what their values can be.
- Tags are case-sensitive in XML, and every tag must have a close tag, although the abbreviation `<element />` works as an implicit close tag.