

DHTML and JavaScript

"Fun on the client side"

DHTML

- Dynamic HTML (DHTML) allows modification to the components of the document at the tag level.
- JavaScript programming is required.
- JScript is the Microsoft name for JavaScript.
- A standardized version is known as ECMAScript.
- There are very minor differences that can be generally ignored.

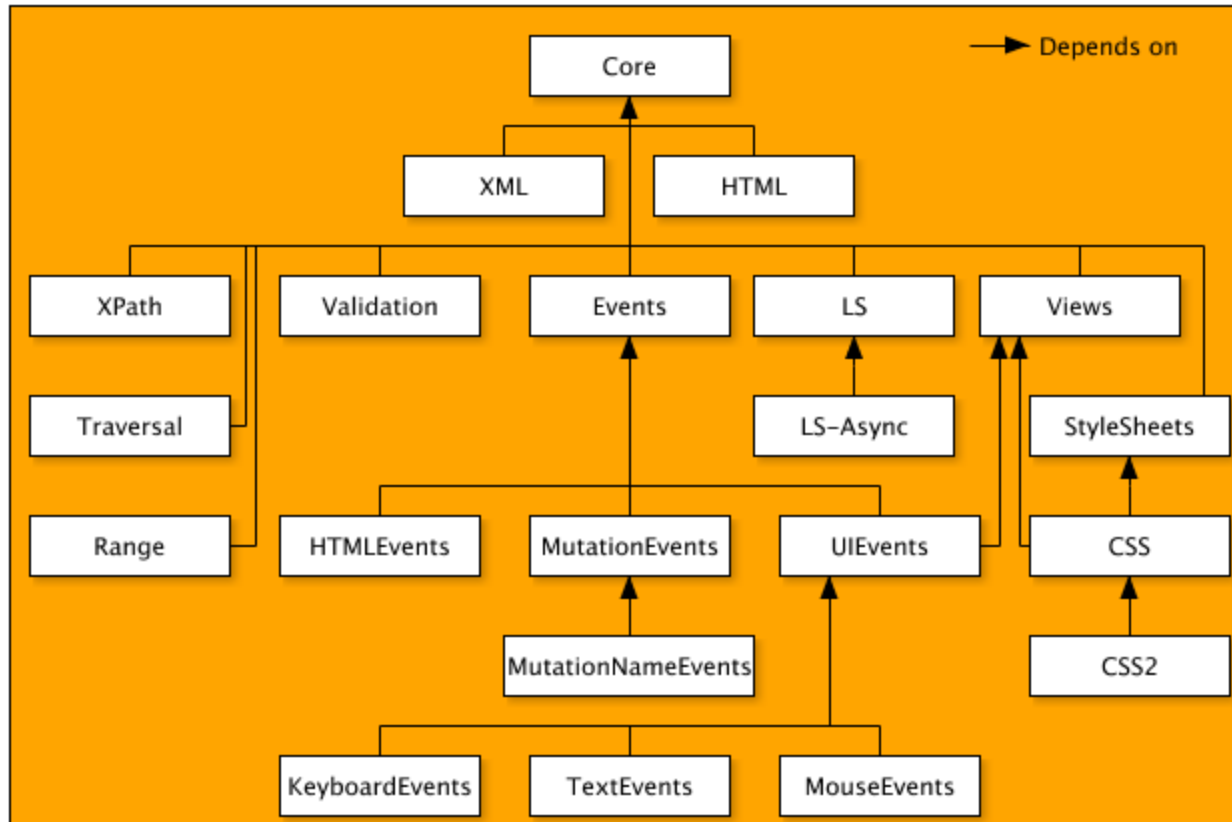
The Document Object Model

- An HTML document has a structure that is based on the Document Object Model, or DOM.
- This is specified by the W3C although all popular browsers differ in some way.
- This makes browser independent development tricky but less so with current browsers.
- The current DOM can be found at <http://www.w3.org/DOM/>

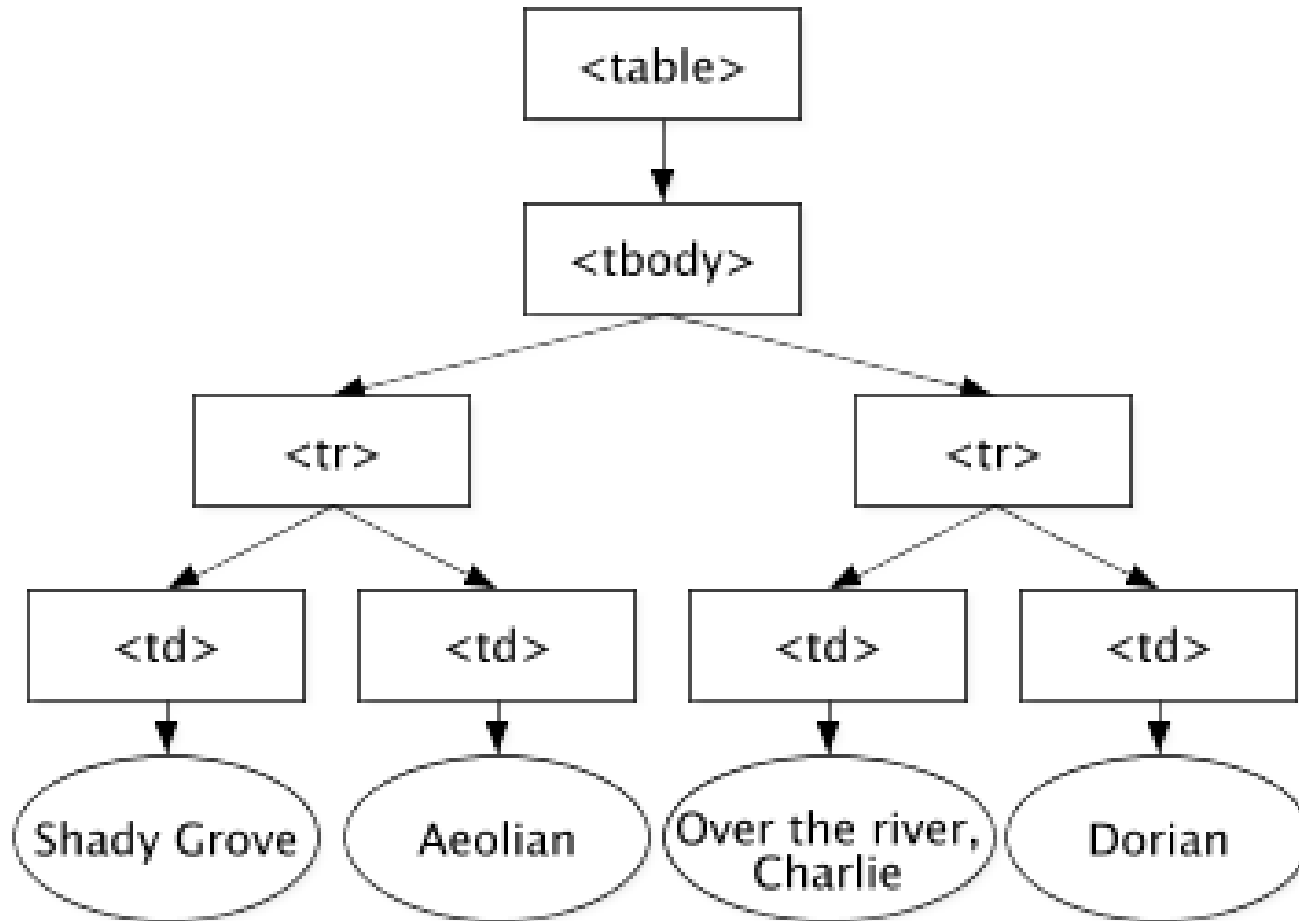
DOM

- The Document Object Model is a platform- and language-neutral interface that will allow programs and scripts to dynamically access and update the content, structure and style of documents. The document can be further processed and the results of that processing can be incorporated back into the presented page.

The full Level 3 DOM

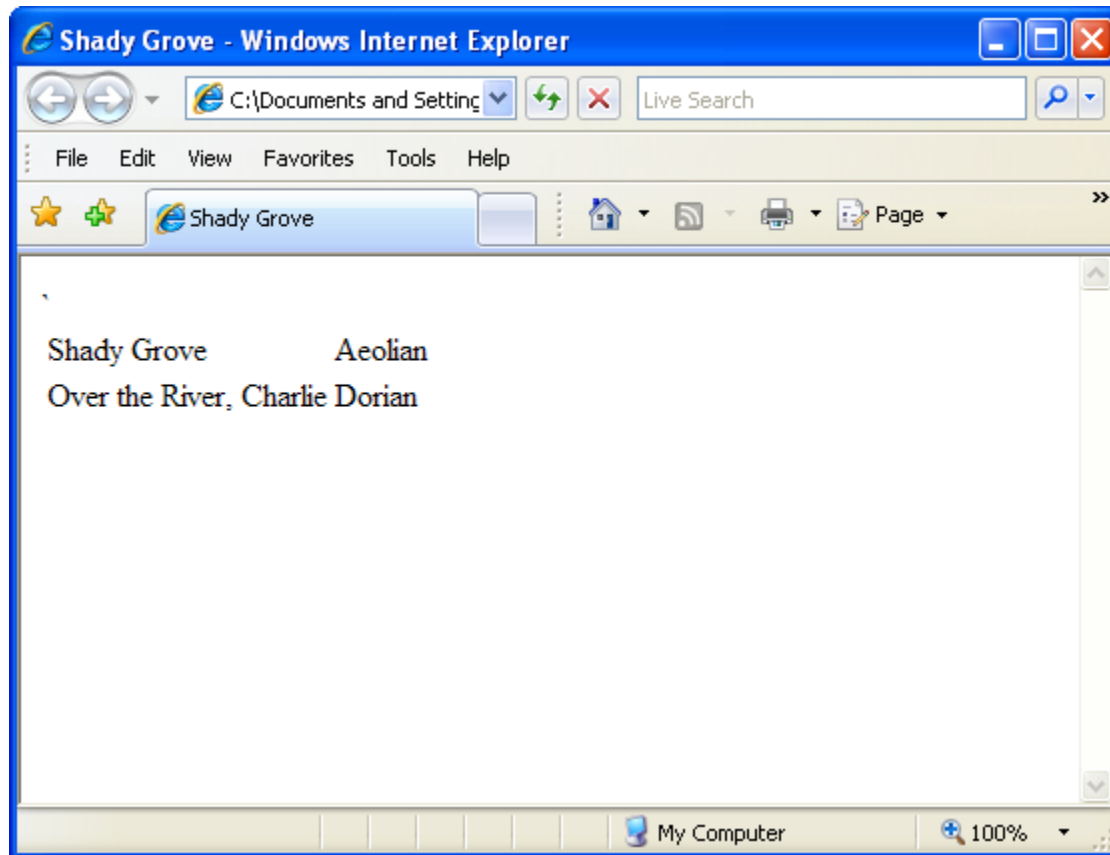


The DOM



A Simple Table

```
<table>
  <tbody>
    <tr>
      <td>Shady Grove</td>
      <td>Aeolian</td>
    </tr>
    <tr>
      <td>Over the River, Charlie</td>
      <td>Dorian</td>
    </tr>
  </tbody>
</table>
```



Windows and Documents

- A browser opens a window that is in turn filled with a document.
- Only one document can be in a window at a time.
- Other windows can be created by script in the active document (pop ups) or by user actions that open new documents in a different window.
- The current document is replaced in the current window when you click on a link or by script replacing the document in the current window.

The Hierarchy

- There is a strict hierarchy of windows, documents, and all the tags that are used in a document.
- There is a concept of *containership*.
- For example - a form contains check boxes, text entry boxes, and buttons.
- A properly formed HTML document has a hierarchy of *objects*. Windows, documents, and tags are all objects in this hierarchy.

Properties, Methods, and Events

- Each element of the DOM hierarchy may have properties that describe some aspect of the object and may be changeable by script.
- Methods can be associated with objects to perform operations. For example, the *window* object has an `Open()` method.
- Events are associated with actions performed by the user and may be passed along to script. For example clicking on some objects trigger the *onClick event*.

JavaScript

- JavaScript is not directly related to Java.
- It is very much like C or C++.
- It supports a simple object programming model.
- JavaScript is *typeless*. Variables are not declared to be of a particular type.
- JavaScript is *interpreted* by the browser and code is only parsed when it executes.
- This means that some errors may not show up when the page is first loaded. Generally browsers ignore errors and do the best they can to render the page.

MSDN References

- Unless you want to purchase a good book on JavaScript you should read the MSDN Library documentation.
- It is found at [msdn:Web Development /Scripting/JScript and VBscript/JScript/Jscript Users Guide](#)
- There are multiple sections to the user guide and you should read at least the fundamentals section.
- A very comprehensive book is:
“JavaScript, The Definitive Guide,” David Flanagan, O’Reilly, 2011.

How to put JavaScript into a document.

- The `<script>` tag is used to place script into the document.
- To protect some very old browsers you may see the script inside a comment which is inside the `<script>` tag. Don't bother with this anymore.
- Note – the use of the “language” attribute is **deprecated**. Use “type” instead.

```
<script type="text/javascript">  
<!--  
-----script goes here-----  
//-->  
</script>
```

XHTML Compatibility

- To comply with XML standards the following is recommended and works with HTML as well:

```
<script type="text/javascript"><![CDATA[  
-----script goes here-----  
]]></script>
```

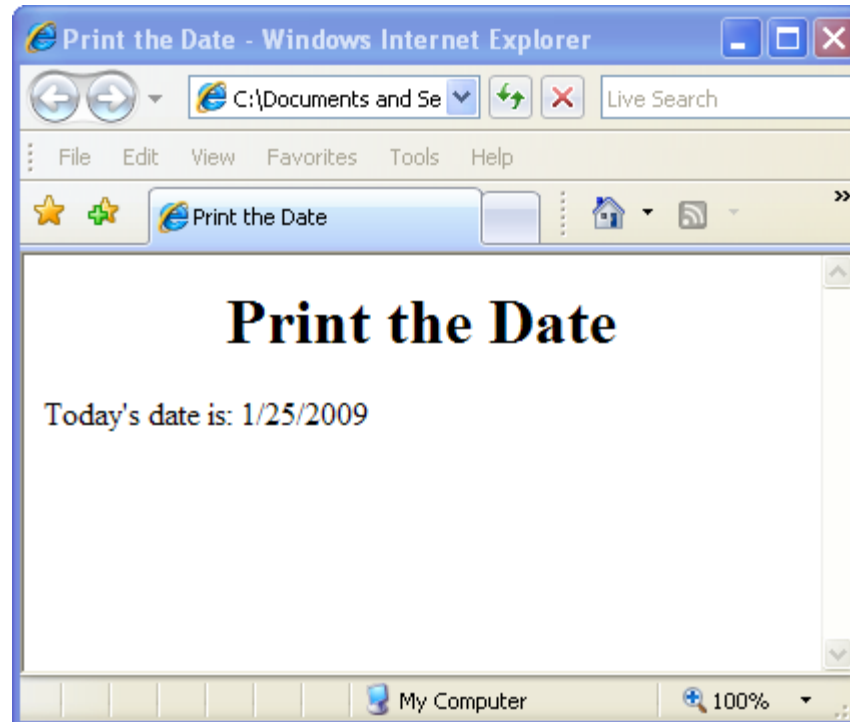
Where to place JavaScript

- JavaScript that is not executed "in line" should be placed between the `<head>` and `</head>` tags.
- JavaScript can be used in the body of your document to generate dynamic effects as the page is being "flowed." The `document.write` method can be used for this purpose.
- If the script is in a function it will not execute. However, you can call a function from the in line code.

Print the date

(date.htm)

```
<body>
<h1 align="center">Print the Date</h1>
<script type="text/javascript">
var d, s = "Today's date is: ";    //Declare variables.
    d = new Date();                //Create Date object.
    s += (d.getMonth() + 1) + "/"; //Get month (zero based)
    s += d.getDate() + "/";       //Get day
    s += d.getFullYear();         //Get year.
    document.write(s);
</script>
</body>
```



Debugging

- Let's use the previous example to show how to debug using VS 2013
- Create a web site on your disk
- Add your HTML file
- Place breakpoints etc.
- Click Debug/Start.
- If asked for a web.config file to allow debugging say yes.
- **Demo.**

The Window Object

- The window object is the object created by the browser that contains the current document.
- A new window can be created with the open method.
- This will be a second window and will contain the URL passed to open.
- There are many other options. Check them out.
- This is the nasty pop-up window and will usually be blocked by the browser until you allow it. (DEMO)

```
<script type="text/javascript">  
    window.open("http://www.bu.edu");  
</script>
```

Javascript Events

- Events are triggered by various actions associated with the document objects (tags).
- There are many events including:
 1. onclick
 2. onmouseover
 3. onmouseout
 4. onchange
- Event handlers are added to the inside of the opening tag.
- A Javascript function or simple Javascript can be executed when the event occurs.
- The following example uses a button object and an onclick event handler.

The Document Object

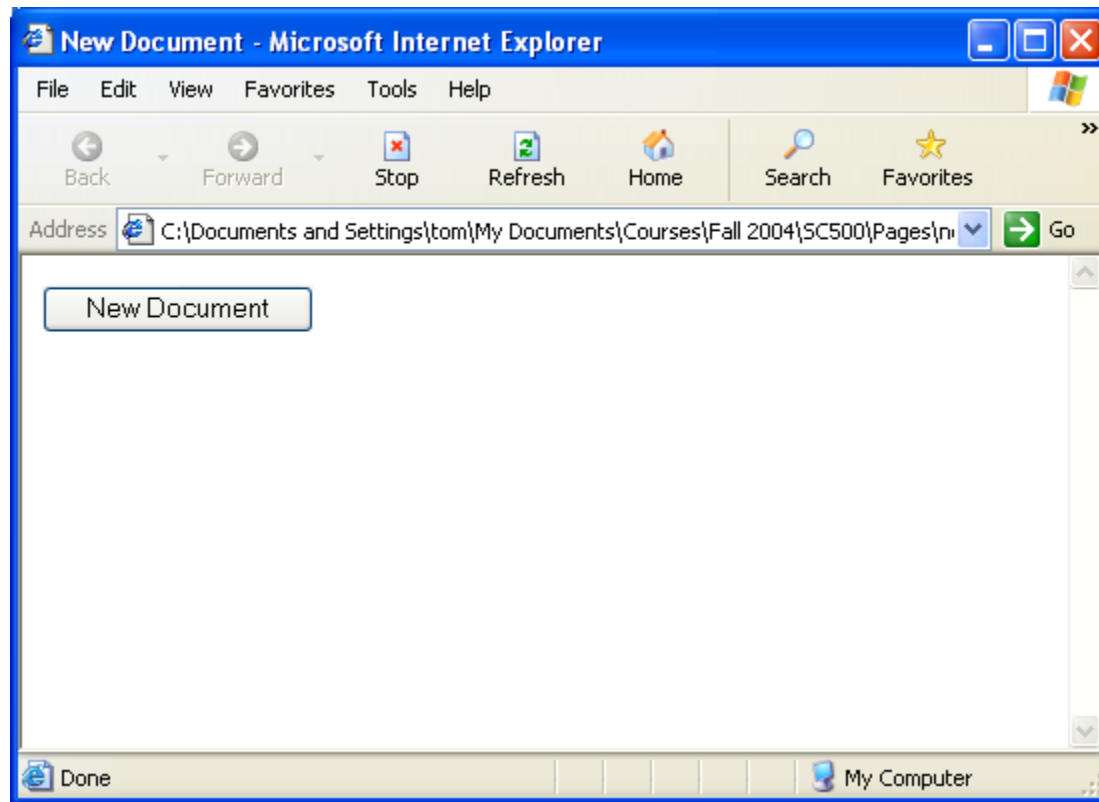
- The document object is contained in the window object and represents the current document being displayed.
- The tags in your document are manipulated through the document object.
- The current document can be replaced dynamically or a new document can be opened and then script can write on it.

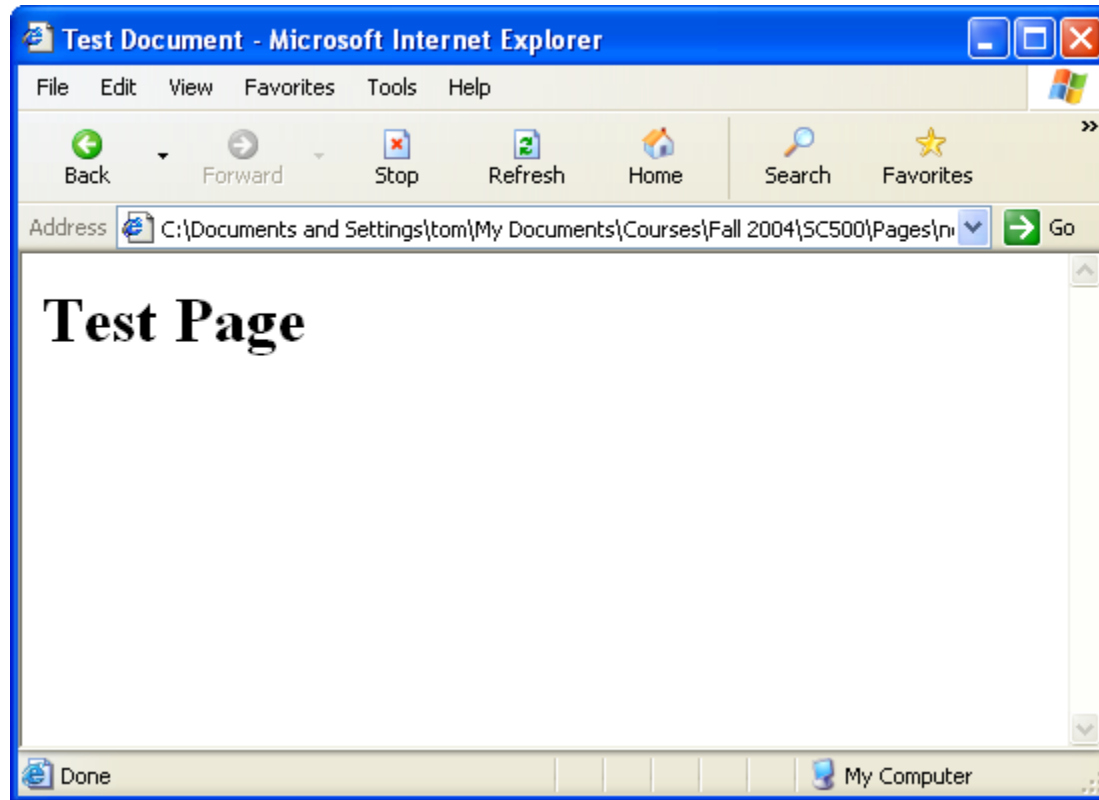
Replacing the Document

(winwrite.htm)

- This example opens a new document and then uses script to write the document with HTML.
- You should call the close method to cause the page to be rendered.

```
<html>
<head>
<title>New Document</title>
<script language="javascript">
function newdoc()
{
    document.open();
    document.write("<HTML><HEAD><TITLE>Test
Document</TITLE></HEAD");
    document.write("<BODY><H1>Test Page</H1></BODY></HTML>");
    document.close();
}
</script>
</head>
<body>
<button onclick="newdoc()">New Document</button>
</body>
</html>
```



Using JavaScript to Change the Styles of an Element

- We must assign an ID to the HTML element or ASP.NET control.
- Use the `document.getElementById("id")` method to get the element object.
- Change the style.

```
<head runat="server">
  <title>Javascript Demo</title>
  <script type="text/javascript">
    function ChangeColor() {
      var el = document.getElementById("Panel1");
      el.style.backgroundColor = 0xff0000;
    }
  </script>
</head>
<body>
  <form id="form1" runat="server">
    <div style="width: 321px">
      <asp:Panel ID="Panel1" runat="server" BorderColor="Black" BorderStyle="Solid"
        BorderWidth="1px" onmouseover="ChangeColor();">
        Place mouse over here to change background color.</asp:Panel>
    </div>
  </form>
</body>
</html>
```

